IMPROVING AUTOMATIC SLEEP STAGING VIA TEMPORAL SMOOTHNESS REGULARIZATION

Huy Phan[1,2], *Elisabeth Heremans*[3], *Oliver Y. Chén*[4], *Philipp Koch*[5,6], *Alfred Mertins*[5,6], *Maarten De Vos*[3]

[1]School of Electronic Engineering and Computer Science, Queen Mary University of London, UK
[2]The Alan Turing Institute, UK
[3]Department of Electrical Engineering, KU Leuven, Belgium
[4]School of Economics, Finance and Management, University of Bristol, UK
[5]Institute for Signal Processing, University of Lübeck, Germany
[6]German Research Center for Artificial Intelligence (DFKI), Germany
*Correspondence email: h.phan@qmul.ac.uk

ABSTRACT

We propose a regularization method, so-called temporal smoothness regularization, for training deep neural networks for automatic sleep staging in small data settings. In intuition, we constrain the cross-entropy losses of any two adjacent epochs in the sequential input to be as close to each other as possible. The regularization closely reflects the slow transition nature of sleep process which implies small information changes between two consecutive sleep epochs. Via the regularization, we essentially discourage the network from overfitting to these small changes. Our experiments show that training the SeqSleepNet base network with the proposed regularization leads to performance improvement over the baseline without the regularization applied. Furthermore, our developed method achieves the performance on par with the state-of-the-art performance while outperforming other existing methods.

Index Terms— Automatic sleep staging, transfer learning, regularization, temporal smoothness, SeqSleepNet

1. INTRODUCTION

Deep neural networks (DNNs) have recently achieved expert-level performance on sleep staging [1] which was once largely done by clinicians in sleep clinics following well-established guidelines, such as the American Academy of Sleep Medicine (AASM) manual [2]. They thus hold promise to transform the current sleep practice and provide a foundation for clinical studies of human sleep. However, reaching this expert-level performance requires a deep learning model to be trained on large databases, for example, with thousands of over-night recordings [3, 4, 5]. In practice, the reliance on large training data limits the applicability of these deep learning models in many scenarios when such a large database is unavailable. For example, when studying sleep disorders [6, 7] or exploring the feasibility of a new monitoring device, like mobile EEG device [8, 9], the collected databases are typically as small as a few tens of subjects. Training a deep learning model on such a small database oftentimes results in an overfitting model, which degrades the performance .

In small data regimes, from the network architecture perspective, it is important that the whole model (like SeqSleepNet [10]), or a part of the multi-branch model (like XSleepNet [3]) is designed to have a small footprint. This can be observed from Fig. 3 where the small models in terms of the number of parameters enjoy better accuracy than the larger one. From the data manipulation perspective, overfitting can be remedied via data augmentation in order to increase the amount and the diversity of the data [11, 12]. From the model training perspective, the most common approach is transfer learning/domain adaptation [8, 7, 13]. That is, a neural network is firstly trained on a large database and then fine-tuned on a small target database. Still, even fine-tuning cannot escape from overfitting especially when the source domain and the target domain are significantly different [13]. Model regularization [14] is another potential direction to explore. However, apart from the common regularization methods like weight decay, leveraging the characteristics of the target signals (sleep EEG signal in this case) for the regularization purpose remains mostly unexplored.

A well-known characteristic of sleep is the slow transition nature of the physiological processes behind sleep stages. That is why a large amount of sleep epochs have the same sleep stages as its neighbors [15]. This also implies the signal changes between two consecutive sleep epochs should be slight. In this work, we leverage this property to propose a regularization method, so-called temporal smoothness regularization, to enforce the cross-entropy losses between any two adjacent sleep epochs to be small. This is expected to help the network to avoid overfitting to minor changes between consecutive sleep epochs. Using SeqSleepNet [10], a sleep staging model with a small footprint, as the base network, our experiments on the SleepEDF Expanded database show that the network with the temporal smoothness regularization outperforms the baseline network without the regularization. In addition, the obtained performance is better than most reported results in the literature while being on par to the state-of-the-art results.

2. THE NETWORK ARCHITECTURE

We employ SeqSleepNet presented in [10], as the backbone network architecture in this work. Due to its small footprint, SeqSleepNet has been shown to be suitable for sleep staging in small data regimes [13, 16, 7, 8]. The architecture of SeqSleepNet is illustrated in Fig. 1. At the high level, the network receives a sequence of L consecutive epochs $(\boldsymbol{S}_1, \boldsymbol{S}_2, \ldots, \boldsymbol{S}_L)$ as input and produces a sequence of corresponding sleep stages $(\hat{\boldsymbol{y}}_1, \hat{\boldsymbol{y}}_2, \ldots, \hat{\boldsymbol{y}}_L)$, following the sequence-to-sequence sleep staging framework [1].

Input: The EEG signal of a 30-second epoch is converted into a log-magnitude time-frequency image $\boldsymbol{S}$ with $T = 29$ time steps and $F = 129$ frequency bins. To that end, short-time Fourier transform (STFT) is applied to the signal with a window length of 2 seconds and 50% overlap. In addition, Hamming window and 256-point fast Fourier transform (FFT) are used. The obtained amplitude spectrum is then log-transformed to result in the image $\boldsymbol{S}$.

Epoch encoding: An epoch of the input sequence in the form of a time-frequency image $\boldsymbol{S}$ is encoded into an embedding vector $\bar{\boldsymbol{x}}$ via the epoch encoding sub-network. This sub-network is composed of (i) a learnable filterbank layer, (ii) a bidirectional recurrent neural network (biRNN), and (iii) a gated attention layer. The learnable filterbank layer consists of M filters ($M < F$), being tasked to smooth and reduce the frequency dimension from F to M bins. The resulting image $\tilde{\boldsymbol{S}}$ of size $T \times M$ is treated as a sequence of T vectors (i.e., T image columns), $(\tilde{\boldsymbol{s}}_1, \tilde{\boldsymbol{s}}_2, \ldots, \tilde{\boldsymbol{s}}_T)$, where $\tilde{\boldsymbol{s}}_t \in \mathbb{R}^M$, $1 \leq t \leq T$. In order to capture the sequential information at the epoch level, the biRNN encodes the sequence $(\tilde{\boldsymbol{s}}_1, \tilde{\boldsymbol{s}}_2, \ldots, \tilde{\boldsymbol{s}}_T)$ and outputs a sequence of feature vectors $(\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_T)$. Here, the biRNN is realized by long short-term memory (LSTM) cells [17] with recurrent batch normalization [18]. Afterwards, the gated attention layer [19] is learned to produce attention weights $(w_1, w_2, \ldots, w_T)$ which are used to combine the feature vectors $(\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_T)$ to derive the embedding vector $\bar{\boldsymbol{x}} = \sum_{t=1}^{T} w_t \boldsymbol{x}_t$.

Sequence modelling: Via the above epoch encoding sub-network, the original input sequence $(\boldsymbol{S}_1, \boldsymbol{S}_2, \ldots, \boldsymbol{S}_L)$ is transformed into the sequence of embedding vectors $(\bar{\boldsymbol{x}}_1, \bar{\boldsymbol{x}}_2, \ldots, \bar{\boldsymbol{x}}_L)$. Sequential modelling on the sequence $(\bar{\boldsymbol{x}}_1, \bar{\boldsymbol{x}}_2, \ldots, \bar{\boldsymbol{x}}_L)$ is accomplished by a biRNN and produces the sequence of output vectors $(\boldsymbol{o}_1, \boldsymbol{o}_2, \ldots, \boldsymbol{o}_L)$. An output layer with softmax activation is finally employed for classification and results in the sequence of probability output vectors $(\hat{\boldsymbol{y}}_1, \hat{\boldsymbol{y}}_2, \ldots, \hat{\boldsymbol{y}}_L)$, where $\hat{\boldsymbol{y}}_\ell \in [0, 1]^C$, $1 \leq \ell \leq L$, and $C = 5$ is the number of sleep stages.

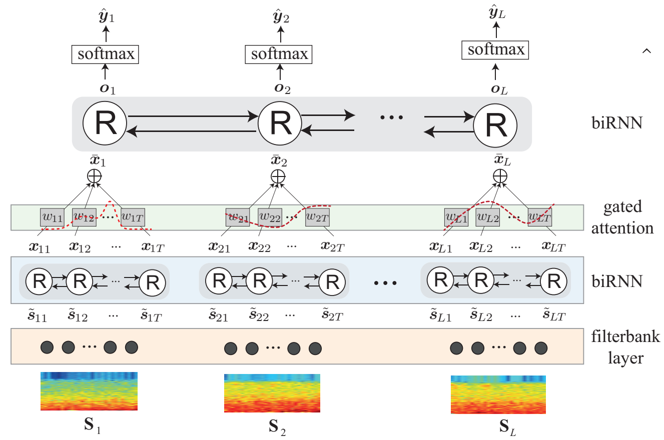

Fig. 1. The architecture of SeqSleepNet adapted from [10].

3. REGULARIZATION VIA ENFORCING THE TEMPORAL SMOOTHNESS

The seminal SeqSleepNet [10] was trained using the cross-entropy loss averaged over the input sequence. Given an input sequence with the ground-truth one-hot encoding vectors $(\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_L)$ and the corresponding sequence of classification outputs $(\hat{\boldsymbol{y}}_1, \hat{\boldsymbol{y}}_2, \ldots, \hat{\boldsymbol{y}}_L)$, the loss function is given as

$$E(\boldsymbol{\theta}) = -\frac{1}{L} \sum\nolimits_{l=1}^{L} \boldsymbol{y}_l \log \hat{\boldsymbol{y}}_l + \lambda ||\boldsymbol{\theta}||_2^2. \quad (1)$$

Here, $\boldsymbol{\theta}$ denotes the network parameters and λ is the hyper-parameter of the ℓ_2-norm regularization. To introduce the temporal smoothness regularization to the loss function, we enforce the difference between the cross-entropy losses of $L - 1$ pairs of adjacent epochs in the sequence to be as small as possible. The loss function with the temporal smoothness regularization becomes

$$\begin{aligned} E(\boldsymbol{\theta}) = &- \frac{1}{L} \sum_{l=1}^{L} \boldsymbol{y}_l \log \hat{\boldsymbol{y}}_l + \lambda ||\boldsymbol{\theta}||_2^2 \\ &+ \gamma \frac{1}{L-1} \sum_{l=1}^{L-1} (\boldsymbol{y}_l \log \hat{\boldsymbol{y}}_l - \boldsymbol{y}_{l+1} \log \hat{\boldsymbol{y}}_{l+1})^2, \quad (2) \end{aligned}$$

where γ is the hyper-parameter of the temporal smoothness regularization. Enforcing small differences between the losses of adjacent epochs is sensible given the fact that the signal changes between two adjacent epochs is small due to the slow physiological process during sleep. In other words, the classification probability the network outputs for two adjacent epochs with similar features should not diverge too far away from each other. This helps to prevent the network from overfitting to minor changes between consecutive epochs.

The temporal smoothness regularization in (2) can be considered as the first-order one. In general, regularization of higher orders (up to $L-1$) can be further incorporated. However, we empirically found that the first-order regularization is sufficient as adding the second-order one (i.e. the difference of first-order differences) did not result in a considerable performance improvement.

Concretely, the loss function in (2) is used to train the network described in Section 2. It is also worth noting that the seminal SeqSleepNet is the special case of the network presented here when $\gamma = 0$.

4. EXPERIMENTS

4.1. Sleep-EDF Expanded Database

We used the Sleep Cassette (SC) subset of the popular Sleep-EDF Expanded dataset [20, 21] in the experiments. It consists of 20 subjects (10 males and 10 females) aged 25-34. Two consecutive day-night PSG recordings were collected for each subject (except for subject 13 who had one night's data lost due to device failure), making a total of 39 recordings. Each 30-second PSG epoch was manually labelled into one of eight categories {W, N1, N2, N3, N4, REM, MOVEMENT, UNKNOWN} by sleep experts according to the R&K standard [22]. N3 and N4 stages were merged into N3 collectively and MOVEMENT and UNKNOWN categories were excluded. We adopted the Fpz-Cz EEG channel in this study.

4.2. Experimental setup

We conducted leave-one-subject-out cross validation. At each iteration, 4 subjects were randomly left out from the training subjects for validation purpose. The performance metrics, including accuracy, Cohen's kappa (κ), and macro F1-score (MF1), were then calculated over all the cross-validation iterations. Due to the small size of the data, we repeated each experiment 5 times and reported the mean and standard deviation of the performance metrics.

For clarity, let us denote the proposed SeqSleepNet with the temporal smoothness regularization as *Regularized SeqSleepNet*. To assess the effectiveness of the temporal smoothness regularization, the seminal SeqSleepNet [10] was used as the baseline for comparison. Both the networks (i.e., Regularized SeqSleepNet and SeqSleepNet) were implemented using the Tensorflow framework [23] and shared a similar configuration. More specifically, the sequence length $L = 20$, the number of learnable filters $M = 32$, the hidden state vector of all the LSTM cells used for epoch encoding and sequence modelling was configured to have 64 units. The ℓ_2-norm regularization parameter λ in (1) and (2) was fixed to 10^{-3}. Particularly for the Regularized SeqSleepNet, to investigate the influence of the temporal smoothness regularization on the network's performance, we experimented with different values in the set $\{10^0, 10^{-1}, 10^{-2}, 10^{-3}, 10^{-4}, 10^{-5}\}$ for the hyper-parameter γ in (2).

The network training was done using Adam optimizer [29] with a learning rate of 10^{-4} for 10 epochs with a mini-batch size of 32 sequences. During training, evaluation on the validation data was carried out every 100 training steps and the network snapshot with the best validation accuracy was retained for evaluating on the test data.

4.3. Experimental results

Table 1 shows the overall performance obtained by the Regularized SeqSleepNet with different values of γ and the SeqSleepNet baseline. Note that the proposed Regularized SeqSleepNet reduces to the SeqSleepNet baseline when $\gamma = 0$ (i.e., $10^{-\infty}$). Overall, there is a range of values for γ (from 10^{-1} to 10^{-5}) where the Regularized SeqSleepNet outperforms the baseline. $\gamma = 10^{-4}$ appears to be the most suitable one, leading to 0.6%, 0.007, and 0.7% gain on the accuracy, κ, and MF1, respectively. On the one hand, when over-regularized, for example when $\gamma = 10^0$, the Regularized SeqSleepNet's performance decreases and becomes worse than that of the baseline. It can be expected that its performance will be further degraded with larger values of γ. This is understandable since stronger regularization will likely cause the model to ignore changes of the input signal over time, leading to an underfitting model. On the other hand, it can also be expected that less regularization with small γ will make the Regularized SeqSleepNet converge to the baseline.

The influence of the temporal smoothness regularization is further manifested in Fig. 2 in which we show the development of the Regularized SeqSleepNet's cross-entropy loss on the test data during the training course of one cross-validation iteration. In the figure, for clarify, the loss curves were averaged over 5 experiment times and smoothed via moving average with a window size of 5. As can be seen, among all the curves, the one with the best $\gamma = 10^{-4}$ is lowest while the one with over-regularization with $\gamma = 10^0$ is clearly highest, reflecting the overall performance in Table 1.

In Table 2, we compare the performance obtained by the

Table 1. Overall performance obtained by Regularized SeqSleepNet with different values of γ, compared to that obtained by the SeqSleepNet baseline.

	γ	Acc.	κ	MF1
Regularized SeqSleepNet	10^0	85.5 ± 0.1	0.802 ± 0.004	78.4 ± 0.4
	10^{-1}	86.0 ± 0.3	0.808 ± 0.004	79.2 ± 0.5
	10^{-2}	85.9 ± 0.3	0.807 ± 0.004	79.0 ± 0.3
	10^{-3}	86.0 ± 0.3	0.809 ± 0.004	79.1 ± 0.2
	10^{-4}	86.2 ± 0.2	0.811 ± 0.003	79.3 ± 0.3
	10^{-5}	85.8 ± 0.4	0.807 ± 0.006	79.0 ± 0.3
SeqSleepNet	$10^{-\infty}$	85.6 ± 0.3	0.803 ± 0.004	78.6 ± 0.2

Table 2. Performance comparison between the Regularized SeqSleepNet, the SeqSleepNet baseline, and existing methods reporting results on the Sleep-EDF Expanded dataset.

System	Overall metrics					Class-wise MF1				
	Acc.	κ	MF1	Sens.	Spec.	Wake	N1	N2	N3	REM
Regularized SeqSleepNet	86.2 ± 0.2	0.811 ± 0.003	79.3 ± 0.3	78.8 ± 0.4	96.3 ± 0.1	91.8 ± 0.2	45.7 ± 1.4	88.3 ± 0.3	86.9 ± 0.4	84.0 ± 0.8
SeqSleepNet [10]	85.6 ± 0.3	0.803 ± 0.004	78.6 ± 0.2	78.2 ± 0.1	96.2 ± 0.1	91.2 ± 0.6	44.7 ± 0.8	88.0 ± 0.1	86.2 ± 0.2	83.0 ± 0.8
XSleepNet2 [3]	86.3	0.813	80.6	80.2	96.4	92.2	51.8	88.0	86.8	83.9
XSleepNet1 [3]	86.0	0.810	80.0	79.6	96.3	91.3	49.5	88.0	86.9	84.2
RL+TCNN+CRF [12]	85.4	0.800	79.3	–	–	90.0	46.6	88.4	86.1	84.6
TinySleepNet [24]	85.4	0.800	80.5	–	–	90.1	51.4	88.5	88.3	84.3
U-Sleep [4]	–	–	79.0	–	–	93.0	57.0	86.0	71.0	88.0
AttnSleep [25]	84.4	0.790	78.1	–	–	89.7	42.6	88.8	90.2	79.0
IITNet [26]	83.9	0.780	77.6	–	–	–	–	–	–	–
DeepSleepNet* [27]	82.0	0.760	76.9	–	–	86.7	45.5	85.1	83.3	82.6
FCNN+RNN [3]	81.8	0.754	75.6	75.7	95.3	89.4	44.1	84.0	84.0	76.3
SleepEEGNet [28]	81.5	0.750	76.6	–	–	89.4	44.4	84.7	84.6	79.6

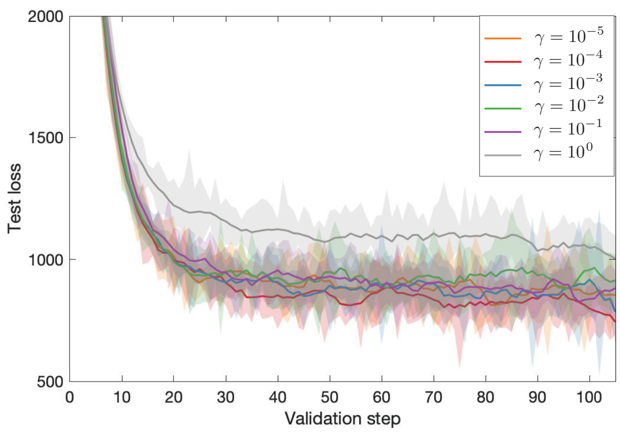

Fig. 2. The development of the Regularized SeqSleepNet's cross-entropy loss on the test data during the training course of one cross-validation iteration.

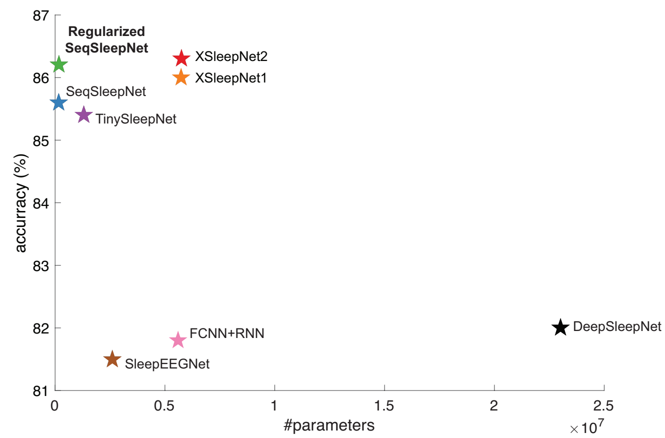

Fig. 3. The overall accuracy vs. the model size of the Regularized SeqSleepNet and other systems (when their model size is available) in Table 2.

Regularized SeqSleepNet with existing methods reporting performance on the SleepEDF Expanded database in literature. Overall, with the achieved accuracy of 86.2%, the Regularized SeqSleepNet outperforms the majority of prior works while being on par with the state-of-the-art XSleepNets [3]. These results is noteworthy given that the number of parameters of the Regularized SeqSleepNet is just fractional compared to that of other models, as shown in Fig. 3 for a comparison. With 1.64×10^5 parameters, the Regularized SeqSleepNet is, for example, roughly 8 times smaller than TinySleepNet [24] (1.3×10^6 parameters), 16 times smaller than SleepEEGNet [28] (2.6×10^6 parameters), 35 times smaller than XSleepNet [3] (5.74×10^6 parameters), and 140 times smaller than DeepSleepNet [27] (2.3×10^7 parameters).

5. CONCLUSIONS AND DISCUSSION

Inspired the slow temporal transition process of sleep, we presented a method for regularizing deep neural networks for sequence-to-sequence sleep staging in small data scenarios via enforcing the temporal smoothness over the input sequence. It was achieved by constraining the cross-entropy losses of two adjacent epochs in the input sequence to be as close as possible. Using the SeqSleepNet as the base network, our evaluation on the SleepEDF Expanded database showed better sleep staging performance when the temporal smoothness regularization was applied, compared to the baseline network without the regularization. We further showed that the Regularized SeqSleepNet also outperformed most of existing works while being comparable to the state-of-the-art XSleepNet even though its model footprint is immensely smaller than other models.

Our finding in this work suggests the characteristics of the input signal are not only able to guide the design of network architectures for a problem at hand as demonstrated in many works in different domains but also able to inform domain-specific regularization methods for network training. We speculate that the presented regularization method can be used in combination with other regularization methods, data augmentation techniques, and transfer learning approaches to gain a collective effect on training sleep staging models in small data settings.

6. REFERENCES

[1] H. Phan and K. Mikkelsen, "Automatic sleep staging of eeg signals: Recent development, challenges, and future directions," *Physiological Measurement*, vol. 43, no. 4, pp. 04TR01, 2022.

[2] Richard B. Berry, Rita Brooks, Charlene Gamaldo, Susan M. Harding, Robin M. Lloyd, C. L. Marcus, and Bradley V. Vaughn, *The AASM Manual for the Scoring of Sleep and Associated Events: Rules, Terminology and Technical Specifications*, American Academy of Sleep Medicine, 2.3 edition, 2016.

[3] H. Phan *et al.*, "XSleepNet: Multi-view sequential model for automatic sleep staging," *IEEE Trans. on Pattern Analysis and Machine Intelligence (TPAMI)*, vol. 44, no. 9, pp. 5903–5915, 2022.

[4] M. Perslev *et al.*, "U-Sleep: resilient high-frequency sleep staging," *npj Digital Medicine*, vol. 4, no. 72, 2021.

[5] S. Biswal *et al.*, "Expert-level sleep scoring with deep neural networks," *J Am Med Inform Assoc.*, vol. 25, no. 12, pp. 1643–1650, 2018.

[6] N. Cooray, F. Andreotti, C. Lo, M. Symmonds, M. T. M. Hu, and M. De Vos, "Detection of rem sleep behaviour disorder by automated polysomnography analysis," *Clinical Neurophysiology*, vol. 130, no. 4, pp. 505–514, 2019.

[7] E. R. M. Heremans, H. Phan, P. Borzée, B. Buyse, D. Testelmans, and M. De Vos, "From unsupervised to semi-supervised adversarial domain adaptation in eeg-based sleep staging," *Journal of Neural Engineering (JNE)*, vol. 19, no. 2022, pp. 036044, 2022.

[8] E. R. M. Heremans, H. Phan, A. H. Ansari, P. Borzée, B. Buyse, D. Testelmans, and M. De Vos, "Feature matching as improved transfer learning technique for wearable eeg," *Biomedical Signal Processing and Control*, vol. 78, pp. 104009, 2022.

[9] K. B. Mikkelsen *et al.*, "Accurate whole-night sleep monitoring with dry-contact ear-EEG," *Sci. Rep.*, vol. 9, no. 1, pp. 1–12, 2019.

[10] H. Phan *et al.*, "SeqSleepNet: end-to-end hierarchical recurrent neural network for sequence-to-sequence automatic sleep staging," *IEEE Trans. Neural Syst. Rehabilitation Eng.*, vol. 27, no. 3, pp. 400–410, 2019.

[11] J. Fan, C. Sun, C. Chen, X. Jiang, X. Liu, X. Zhao, L. Meng, C. Dai, and W. Chen, "Eeg data augmentation: towards class imbalance problem in sleep staging tasks," *Journal of Neural Engineering*, vol. 17, no. 5, pp. 056017, 2022.

[12] E. Khalili and B. M. Asl, "Automatic sleep stage classification using temporal convolutional neural network and new data augmentation technique from raw single-channel eeg," *Computer Methods and Programs in Biomedicine*, vol. 204, no. 2021, pp. 106063, 2021.

[13] H. Phan *et al.*, "Towards more accurate automatic sleep staging via deep transfer learning," *IEEE Trans. Biomed. Eng.*, vol. 68, no. 6, pp. 1787–1798, 2021.

[14] J. Kukacka, V. Golkov, and D. Cremers, "Regularization for deep learning: A taxonomy," *arXiv preprint arXiv:1710.10686*, 2017.

[15] H. Phan *et al.*, "Joint classification and prediction CNN framework for automatic sleep stage classification," *IEEE. Trans. Biomed. Eng.*, vol. 66, no. 5, pp. 1285–1296, 2019.

[16] H. Phan *et al.*, "Personalized automatic sleep staging with single-night data: a pilot study with KL-divergence regularization," *Physiological Measurement*, vol. 41, no. 6, pp. 064004, 2020.

[17] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computing*, vol. 9, no. 8, pp. 1735–1780, 1997.

[18] T. Cooijmans *et al.*, "Recurrent batch normalization," *arXiv Preprint arXiv:1603.09025*, 2016.

[19] T. Luong, H. Pham, and C. D. Manning, "Effective approaches to attention-based neural machine translation," in *Proc. EMNLP*, 2015, pp. 1412–1421.

[20] B. Kemp *et al.*, "Analysis of a sleep-dependent neuronal feedback loop: the slow-wave microcontinuity of the EEG," *IEEE. Trans. Biomed. Eng.*, vol. 47, no. 9, pp. 1185–1194, 2000.

[21] A. L. Goldberger *et al.*, "Physiobank, physiotoolkit, and physionet: Components of a new research resource for complex physiologic signals," *Circulation*, vol. 101, pp. e215–e220, 2000.

[22] J. A. Hobson, "A manual of standardized terminology, techniques and scoring system for sleep stages of human subjects," *Electroencephalography and Clinical Neurophysiology*, vol. 26, no. 6, pp. 644, 1969.

[23] M. Abadi *et al.*, "Tensorflow: Large-scale machine learning on heterogeneous distributed systems," *arXiv:1603.04467*, 2016.

[24] A. Supratak and Y. Guo, "TinySleepNet: An efficient deep learning model for sleep stage scoring based on raw single-channel EEG," in *Proc. 42nd Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, 2020.

[25] E. Eldele *et al.*, "An attention-based deep learning approach for sleep stage classification with single-channel eeg," *IEEE Trans. on Neural Systems and Rehabilitation Engineering*, vol. 29, pp. 809–818, 2021.

[26] H. Seo *et al.*, "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*, 2020.

[27] A. Supratak *et al.*, "DeepSleepNet: A model for automatic sleep stage scoring based on raw single-channel EEG," *IEEE Trans. Neural Syst. Rehabilitation Eng.*, vol. 25, no. 11, pp. 1998–2008, 2017.

[28] S. Mousavi *et al.*, "SleepEEGNet: Automated sleep stage scoring with sequence to sequence deep learning approach," *PLoS One*, vol. 14, no. 5, pp. e0216456, 2019.

[29] D. P. Kingma and J. L. Ba, "Adam: a method for stochastic optimization," in *Proc. ICLR*, 2015, number 1-13.