DNN Filter Bank Improves 1-Max Pooling CNN for Single-Channel EEG Automatic Sleep Stage Classification

Huy Phan, Fernando Andreotti, Navin Cooray, Oliver Y. Chén, and Maarten De Vos
{huy.phan,fernando.andreotti,navin.cooray,yibing.chen,maarten.devos}@eng.ox.ac.uk

***Abstract*— We present in this paper an efficient convolutional neural network (CNN) running on time-frequency image features for automatic sleep stage classification. Opposing to deep architectures which have been used for the task, the proposed CNN is much simpler. However, the CNN's convolutional layer is able to support convolutional kernels with different sizes, and therefore, capable of learning features at multiple temporal resolutions. In addition, the 1-max pooling strategy is employed at the pooling layer to better capture the shift-invariance property of EEG signals. We further propose a method to discriminatively learn a frequency-domain filter bank with a deep neural network (DNN) to preprocess the time-frequency image features. Our experiments show that the proposed 1-max pooling CNN performs comparably with the very deep CNNs in the literature on the Sleep-EDF dataset. Preprocessing the time-frequency image features with the learned filter bank before presenting them to the CNN leads to significant improvements on the classification accuracy, setting the state-of-the-art performance on the dataset.**

I. INTRODUCTION

Inspired by the success of deep learning paradigms in numerous domains, there is an ongoing methodology trend in dealing with the automatic sleep stage classification task, shifting from conventional techniques to modern deep learning methods. Deep networks have recently been reported to set state-of-the-art performances on different benchmark datasets [1], [2], [3], [4], [5], [6]. CNNs with their great capability of automatic feature learning have been most commonly employed for the task [1], [2], [3]. Combinations of recurrent neural networks (RNNs) with DNNs [6] and CNNs [5] have also been explored to leverage their sequence modelling capability on the learned features.

We propose in this work an approach using a simple yet efficient CNN architecture with time-frequency image features for sleep stage classification. The proposed architecture is similar to those used in [7], [8] for template learning and matching. Opposing to deep architectures which have been used for the task, the proposed CNN is much simpler, consisting of one over-time convolutional layer, one pooling layer, and one softmax layer for classification. However, in contrast to typical CNN architectures which have a single fixed kernel size at a certain convolutional filter, the convolutional layer of the proposed CNN supports different kernel sizes simultaneously. Furthermore, instead of the common subsampling pooling strategy, we exploit 1-max pooling strategy at the pooling layer to retain a single maximum value of each feature map. While being simple, the 1-max pooling strategy is arguably more suitable for capturing the *shift-invariance* property of temporal signals than the common subsampling pooling ones since a particular feature could be replicated at any time in the signal rather than its local region. This pooling scheme has been successfully applied in other domains, such as natural language processing [9] and audio analysis [7], [8].

HP, FA, NC, OYC, and MDV are with the Institute of Biomedical Engineering, University of Oxford, Oxford OX3 7DQ, United Kingdom.

The research was supported by the NIHR Oxford Biomedical Research Centre and Wellcome Trust under Grant 098461/Z/12/Z.

Before presenting the time-frequency image features to the CNN for classification, preprocessing is carried out for frequency smoothing and dimension reduction. We propose to learn a frequency-domain filter bank using a DNN for this purpose. Discriminatively learning filter banks using DNNs has been found useful for audio analysis [10], [11]. We will show that this idea also works well for EEG signals. The proposed DNN has its first hidden layer tailored to enforce various constraints to be able to learn a filter bank which has the characteristics of a normal filter bank, i.e. being non-negative, band-limited and ordered by frequency. We demonstrate significant performance improvements on the Sleep-EDF dataset when using the DNN-learned filter bank for preprocessing rather than a standard filter bank.

II. THE PROPOSED APPROACH

An overview of the proposed approach is shown in Fig. 1. A raw 30-second EEG epoch is firstly transformed into log-power spectrum. A frequency-domain filter bank, e.g. the triangular filter bank or the one learned by a DNN, is then applied on the spectrum for frequency smoothing and dimension reduction. The resulted time-frequency image is finally classified by the proposed 1-max pooling CNN.

A. Time-Frequency Image Representation

The 30-second EEG epoch, sampled at 100 Hz, is transformed into a power spectrum using short-time Fourier transform (STFT) with a window size of two seconds and 50% overlap. Hamming window and 256-point Fast Fourier Transform (FFT) are used. The spectrum is then converted to logarithm scale to produce the log-power spectrum. As a result, we obtain a log-power spectrum image of size $F \times T$, where $F = 129$ and $T = 19$.

For frequency smoothing and dimension reduction, the spectrum is filtered by a filter bank in frequency direction. We study both a linear-frequency triangular filter bank and a DNN-learned filter bank, as illustrated in Fig. 4 (a) and (c), respectively, for this purpose. The former consists of $M = 20$ filters linearly spaced with 50% overlap in the frequency range of $[0, 50]$ Hz. The later also consists of $M = 20$ filters, however, their coefficients are learned automatically by a DNN (cf. Section II-C). Finally, we obtain a time-frequency image $\mathbf{X} \in \mathbb{R}^{M \times T}$ which serves as input to the 1-max pooling CNN.

B. 1-Max Pooling CNN

The architecture of the proposed 1-max pooling CNN is illustrated in Fig. 2. Actually, this CNN acts as a template learning and matching algorithm. A convolutional filter can be interpreted as a time-frequency template that is learned by the CNN. During testing, template matching is carried out by convolving the learned filter through time, resulting in a feature map which indicates how well the template is matched to different parts of the input EEG signal. In turn, the 1-max pooling operator retains a single maximum

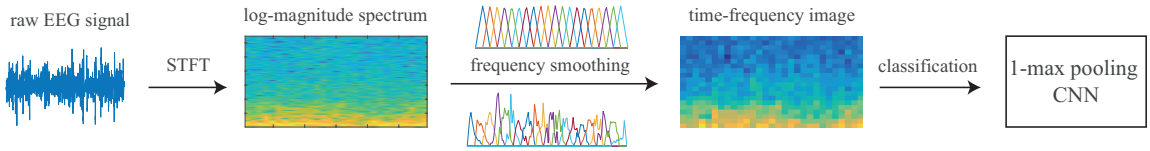

Fig. 1: Overview of the proposed approach.

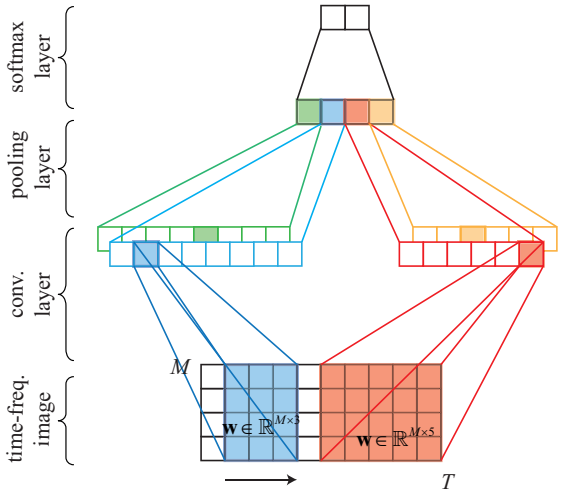

Fig. 2: Illustration of the proposed 1-max pooling CNN architecture. The convolution layer of the CNN consists of two filter sets with temporal widths $w = 3$ and $w = 5$. Each filter set has two individual filters.

matching score as the final feature. Such features derived from all feature maps are concatenated and presented into the softmax layer for classification, as illustrated in Fig. 2. Since the CNN is trained to maximize the classification accuracy on the training set, it is supposed to learn useful templates from the EEG signals for the classification task.

1) Convolutional Layer: Let $\mathbf{w} \in \mathbb{R}^{M \times w}$ be the impulse response of a 2-dimensional linear filter where w denotes the filter's temporal width. Let $\mathbf{X}[i:j]$ further denote the adjacent image slices from i to j. Convolving a filter $\mathbf{w}$ with the image $\mathbf{X}$ in the time direction results in an output vector $\mathbf{O} = (o_1, \ldots, o_{T-w+1})$ where:

$$o_i = (\mathbf{X} * \mathbf{w})_i = \sum_{k,l} (\mathbf{X}[i : i+w-1] \odot \mathbf{w})_{k,l}. \quad (1)$$

Here, $*$ indicates the convolution operation and $\odot$ denotes elementwise multiplication.

Afterwards, *Rectified Linear Units* (ReLU) [12], given in (3), is applied to the output vector to yield the feature map $\mathbf{A} = (a_1, \ldots, a_{T-w+1})$:

$$a_i = h(o_i + b), \quad (2)$$
$$h = \max(0, x), \quad (3)$$

where $b \in \mathbb{R}$ denotes a bias term.

The CNN is designed to have Q different filters of the same temporal width concurrently so that it is able to learn multiple complementary features. Furthermore, in order to capture features at multiple temporal resolutions, R such filter sets with different temporal widths are included into the CNN. Hence, the network consists of $Q \times R$ different filters in total.

2) 1-Max Pooling Layer: The feature map obtained by convolving a filter over an time-frequency image indicates how well the template is matched to different parts of the EEG signal. We then apply 1-max pooling function [9], [7] on a feature map to reduce it into a single most dominant feature which corresponds to the maximum matching score:

$$a_* = \max_{i \in \{1,\ldots,T-w+1\}} a_i. \quad (4)$$

By reducing its feature map to a single most dominant feature, each filter in the convolutional layer is expected to be optimized to a useful pattern that could occur at any time in the signal. Pooling all feature maps pf $Q \times R$ filters results in a feature vector of size $Q \times R$.

3) Softmax Layer: Classification is accomplished by a standard softmax layer. The network is trained to minimize the cross-entropy error over N training samples:

$$E(\boldsymbol{\theta}) = -\frac{1}{N} \sum_{i=1}^{N} \boldsymbol{y}_i \log(\hat{\boldsymbol{y}}_i(\boldsymbol{\theta})) + \frac{\lambda}{2} \|\boldsymbol{\theta}\|_2^2. \quad (5)$$

In (5), $\boldsymbol{\theta}$, $\hat{\boldsymbol{y}}$, and $\boldsymbol{y}$ denote the network parameters, the predicted posterior distribution, and the one-hot encoded groundtruth distribution, respectively. λ denotes the hyper-parameter that trades off the error terms and the ℓ_2-norm regularization term. For further regularization, *dropout* [13] is also employed. The network training is performed using the *Adam* optimizer [14].

C. DNN for Filter-Bank Learning

In Section II-A, using the fixed triangular filterbank to preprocess the time-frequency images, we have considered different frequency subbands equally. However, it is reasonable to somehow emphasize the subbands that are more important for the task and attenuate those less important. Towards this goal, the filter bank can be learned in a discriminative fashion with a DNN. The proposed DNN architecture for filter-bank learning is illustrated in Fig. 3, consisting of one *filter-bank layer*, three fully-connected (FC) layers, and one softmax layer. The FC layers are typical nonlinear ones with ReLU activation. The filter bank layer is tailored similarly to that in [11] for filter-bank learning.

Formally, let $\mathbf{x} \in \mathbb{R}^F$ be the input vector, the output of the filter-bank layer is computed as:

$$\mathbf{h}_1 = \mathbf{x}\mathbf{W}_{\text{fb}}, \quad (6)$$

where $\mathbf{W}_{\text{fb}} \in \mathbb{R}^{F \times M}$ in (6) plays the role of the filter-bank weight matrix and M denote the number of filters. Note that M is also the number of hidden units of the filter-bank layer. For the learned filter bank to have the characteristics of a normal filter bank, i.e. non-negative, band limited and ordered by frequency, it is necessary to enforce constraints and re-write $\mathbf{W}_{\text{fb}}$ as

$$\mathbf{W}_{\text{fb}} = f_+(\mathbf{W}) \odot \mathbf{S}, \quad (7)$$

where $\mathbf{W} \in \mathbb{R}^{F \times M}$, $\mathbf{S} \in \mathbb{R}_+^{F \times M}$. $\mathbf{W}$ is now the weight matrix that will practically be learned by the DNN. The non-negative function

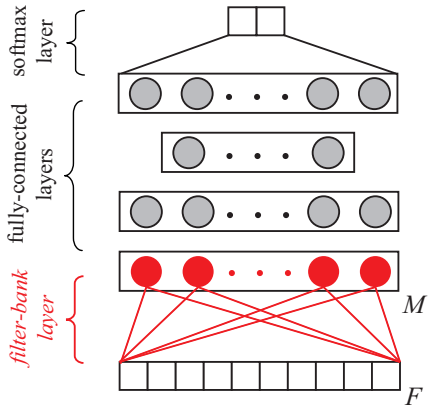

Fig. 3: Illustration of the DNN architecture for filter bank learning.

f_+ is applied on $\mathbf{W}$ to make the elements of the filter bank non-negative. $\mathbf{S}$ is the constant non-negative matrix to enforce the filters to have limited band, regulated shape and ordered by frequency. We employ *sigmoid* for the function $f_+(x) = \frac{1}{1+\exp(-x)}$ and a linear-frequency triangular filter bank matrix for $\mathbf{S}$, as illustrated in Fig. 4 (b).

Different from the 1-max pooling CNN which works on entire time-frequency images of 30-second EEG epochs, the DNN operates on feature vectors of two-second EEG frames, i.e. the spectral columns of the log-power spectrum described in Section II-A. The DNN is trained to minimize the cross-entropy given in (5) (without ℓ-2 norm regularization) over the training set. For training purpose, a two-second EEG frame is labelled by the label of the 30-second epoch from which it is stemmed. Dropout is also applied to weight matrices for regularization purpose, except for those of the filter bank layer. Fig. 4 (c) shows one of the filter banks learned in the experiments (cf. Section III for further details).

III. EXPERIMENTS

A. Sleep-EDF Expanded Dataset

We evaluated the proposed approach on the Sleep-EDF Expanded dataset [15] available from PhysioNet [16]. There are 20 subjects in total. PSG recordings, sampled at 100 Hz, of two subsequent day-night periods are available for each subject, except for subject 13. Each 30-second epoch of the recordings was manually labelled by sleep experts according to the R&K standard [17] into one of eight categories {W, N1, N2, N3, N4, REM, MOVEMENT, UNKNOWN}. Similar to previous works [3], [4], [5], N3 and N4 stages were merged into a single stage N3. MOVEMENT and UNKNOWN were also excluded. We used the single-channel Fpz-Cz in the experiments. It is worth mentioning that there exist two different experimental settings on this dataset: (1) only the in-bed parts of the recordings are included [3], [4] and (2) 30-minute periods before and after in-bed periods are additionally included. For convenience, we will identify them as Setting 1 and 2. We experimented the proposed approach with both settings to make a proper comparison with the previous works.

B. Experimental Setup

We conducted leave-one-subject-out cross validation. At each time, one subject was left out for testing while the remaining 19 subjects were used to train the CNN and DNN networks. Particularly for the CNN training, 4 out of 19 training subjects were further left out for validation. The classification performance over 20 folds will be reported in terms of overall accuracy, macro F1-score (MF1), and kappa index (κ). Per-class F1-score will also be presented.

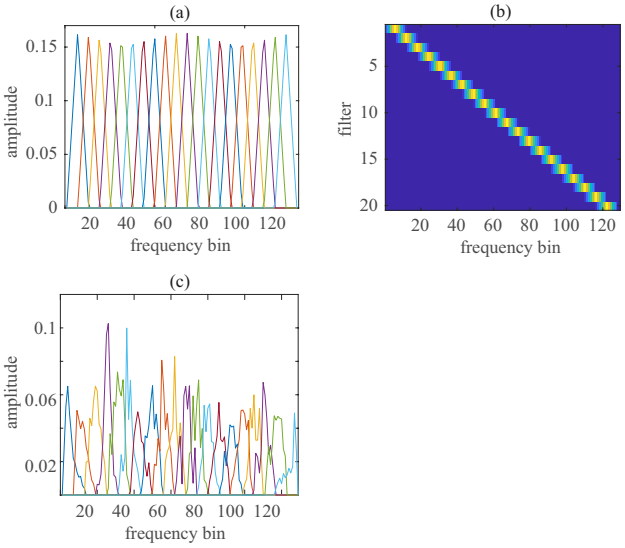

Fig. 4: (a) A linear-frequency triangular filterbank with $M = 20$ filters, (b) the corresponding shape matrix $\mathbf{S}$, (c) a DNN-learned filter bank with $M = 20$ filters.

C. Parameters

The parameters associated with the 1-max pooling CNN are given in Table I (a) while those of the filter-bank-learning DNN are shown in Table I (b). The implementation was based on *Tensorflow* framework. Both the networks were trained for 200 epochs with a batch size of 200. The learning rate was commonly set to 10^{-4} for the *Adam* optimizer. During training, we always randomly generated a data batch to have an equal number of samples for all sleep stages to mitigate the imbalance issue of the dataset. Particularly for the CNN, which network yielding the best overall accuracy on the validation subjects was retained for evaluation.

D. Experimental Results

Table II shows the classification performances obtained by the proposed CNN in comparison with previous deep-learning works on the Sleep-EDF dataset [3], [4], [5]. In the table, *1-Max-CNN 1* and *1-Max-CNN 2* represent the proposed CNN with the regular triangular filtel bank and the DNN-learned filter bank, respectively. The best performance on Setting 1 was reported in [3] which employed auto-encoders combined with a handful of hand-crafted features. Note that the authors in [1] reported better accuracy with a deep CNN similar to that in [4]. However, they experimented on multi-channel (EEG and EOG) setup and fine tuning for personalization, and therefore, the reported results are not comparable with those in Table II. On Setting 2, the best results were reported by

TABLE I: Parameters of the proposed networks: (a) 1-max pooling CNN, (b) filter-bank-learning DNN.

(a)

Parameter	Value
Filter width w	$\{3, 5, 7\}$
Number of filters Q	1000
Dropout	0.2
λ for regularization	10^{-4}

(b)

Layer	Size	Dropout
FC 1	512	0.2
FC 2	256	0.2
FC 3	512	0.2

TABLE II: Performances obtained by different approaches on the Sleep-EDF dataset.

		Overall metrics			Per-class F1-score				
		Acc	MF1	κ	W	N1	N2	N3	REM
Setting 1	Deep CNN [3]	74.8	69.8	–	65.4	43.7	80.6	84.9	74.5
	Auto-encoder [4]	78.9	73.7	–	71.6	47.0	84.6	84.0	81.4
	1-Max-CNN 1	78.3	70.3	0.69	75.3	32.4	85.4	81.9	76.7
	1-Max-CNN 2	**79.8**	72.0	0.72	**77.0**	33.3	**86.8**	**86.3**	76.4
Setting 2	DeepSleepNet 1 [5]	79.8	73.1	0.72	88.1	37.0	82.7	77.3	80.3
	DeepSleepNet 2 [5]	82.0	76.9	0.76	84.7	46.6	85.9	84.8	82.4
	1-Max-CNN 1	76.7	68.1	0.68	87.0	32.9	81.0	69.4	70.2
	1-Max-CNN 2	**82.6**	74.2	**0.76**	**89.8**	33.2	**86.7**	**86.0**	75.4

TABLE III: Confusion matrices of *1-Max-CNN 2* on two experimental settings of the EDF-Sleep dataset.

		Prediction				
		W	N1	N2	N3	REM
Setting 1 Groundtruth	W	3482	458	132	72	365
	N1	463	860	563	13	863
	N2	227	515	14967	795	1073
	N3	77	6	571	4922	15
	REM	283	562	665	8	6193
Setting 2 Groundtruth	W	11338	444	136	97	503
	N1	505	832	571	16	855
	N2	407	424	14995	782	991
	N3	87	3	605	4920	14
	REM	408	529	677	5	6092

DeepSleepNet [5], in which recurrent layers were stacked on top of convolutional layers to leverage both their sequential modelling capability (i.e. the recurrent layers) and feature learning power (i.e. the convolutional layers). We included the results obtained by this network on both channels Pz-Oz and Fpz-Cz for comparison, indicated as DeepSleepNet 1 and 2 in Table II.

Overall, *1-Max-CNN 1* produces relatively good performances, especially on Setting 1. On this setting, even though its performance is marginally lower than the best one reported by Auto-encoder [4], it works better than the deep CNN proposed in [3]. However, using the DNN-learned filter banks to preprocess the time-frequency image features appears to facilitate the template learning process of the CNN, leading to significant improvement on the classification performance. Absolute gains of 1.5%, 1.7%, and 0.03 on overall accuracy, MF1, and κ, respectively, achieved by *1-Max-CNN 2* over *1-Max-CNN 1* can be seen on Setting 1. The respective improvements on Setting 2 are even better, reaching 5.9%, 6.1%, and 0.08. We show the confusion matrices of *1-Max-CNN 2* on the two experimental settings in Table III.

Moreover, *1-Max-CNN 2* sets state-of-the-art performance on overall accuracy on both Setting 1 and 2, improving 1.1% and 0.6% absolute over Auto-encoder [4] and DeepSleepNet 2 [5], respectively. Significant improvements on per-class F1-scores of W, N2, and N3 can also be seen. However, the proposed approach is inefficient to recognize N1 which has been proven challenging to be correctly classified [4], [5], due to similarities with other stages and generally infrequent. The inferior F1-score on this stage average down the overall F1-score of the proposed approach.

IV. CONCLUSIONS

In this paper, we present an efficient approach for automatic sleep stage classification using 1-max pooling CNN and time-frequency image features. The CNN has a simple architecture compared to those have been proposed for the task. However, it is capable of learning features at multiple temporal resolutions and arguably better at capturing time shift-invariance property of EEG signals, thanks to its 1-max pooling layer. For further improvement, a DNN architecture is proposed for learning frequency-domain filter bank to preprocess the time-frequency image inputs. Several constraints are enforced to the first hidden layer so that a filter bank can be learned properly, i.e. it has the characteristics of a normal filter bank. The 1-max pooling CNN combined with a preprocessing step using the DNN-learned filter bank demonstrates state-of-the-art performance in terms of overall accuracy and outperforms other deep-network approaches on the Sleep-EDF dataset. However, there is still room for improvement on recognizing the N1 stage.

REFERENCES

[1] K. Mikkelsen and M. de Vos, "Personalizing deep learning models for automatic sleep staging," *arXiv:1801.02645*, 2018.

[2] J. Zhang and Y. Wu, "A new method for automatic sleep stage classification," *IEEE Trans. on Biomedical Circuits and Systems*, vol. 11, no. 5, pp. 1097–1110, 2017.

[3] O. Tsinalis, P. M. Matthews, Y. Guo, and S. Zafeiriou, "Automatic sleep stage scoring with single-channel EEG using convolutional neural networks," *arXiv:1610.01683*, 2016.

[4] O. Tsinalis, P. M. Matthews, and Y. Guo, "Automatic sleep stage scoring using time-frequency analysis and stacked sparse autoencoders," *Annals of Biomedical Engineering*, vol. 44, no. 5, pp. 1587–1597, 2016.

[5] A. Supratak, H. Dong, C. Wu, and Y. Guo, "Deepsleepnet: A model for automatic sleep stage scoring based on raw single-channel eeg," *IEEE Trans. on Neural Systems and Rehabilitation Engineering*, vol. 25, no. 11, pp. 1998–2008, 2017.

[6] H. Dong, A. Supratak, W. Pan, C. Wu, P. M. Matthews, and Y. Guo, "Mixed neural network approach for temporal sleep stage classification," *IEEE Trans. on Neural Systems and Rehabilitation Engineering*, 2017.

[7] H. Phan, L. Hertel, M. Maass, and A. Mertins, "Robust audio event recognition with 1-max pooling convolutional neural networks," in *Proc. Interspeech*, 2016, pp. 3653–3657.

[8] H. Phan, L. Hertel, M. Maass, P. Koch, R. Mazur, and A. Mertins, "Improved audio scene classification based on label-tree embeddings and convolutional neural networks," *IEEE/ACM Trans. on Acoustics Speech and Signal Processing*, vol. 25, no. 6, pp. 1278–1290, 2017.

[9] Y. Kim, "Convolutional neural networks for setence classification," in *Proc. EMNLP*, 2014, pp. 1746–1751.

[10] T. N. Sainath, B. Kingsbury, A.-R. Mohamed, and B. Ramabhadra, "Learning filter banks within a deep neural network framework," in *Proc. ASRU*, 2013, pp. 297–302.

[11] Hong Yu, Zheng-Hua Tan, Yiming Zhang, Zhanyu Ma, and Jun Guo, "DNN filter bank cepstral coefficients for spoofing detection," *IEEE Access*, vol. 5, pp. 4779–4787, 2017.

[12] X. Glorot, A. Bordes, and Y. Bengio, "Deep sparse rectifier neural networks," in *Proc. AISTATS*, 2011, pp. 315–323.

[13] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *Journal of Machine Learning Research (JMLR)*, vol. 15, pp. 1929–1958, 2014.

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

[15] B. Kemp, A. H. Zwinderman, B. Tuk, H. A. C. Kamphuisen, and J. J. L. Oberye, "Analysis of a sleep-dependent neuronal feedback loop: the slow-wave microcontinuity of the EEG," *IEEE Trans. on Biomedical Engineering*, vol. 47, no. 9, pp. 1185–1194.

[16] A. L. Goldberger, L. A. N. Amaral, L. Glass, J. M. Hausdorff, P. Ch. Ivanov, R. G. Mark, J. E. Mietus, G. B. Moody, C.-K. Peng, and H. E. Stanley, "Physiobank, physiotoolkit, and physionet: Components of a new research resource for complex physiologic signals," *Circulation*, vol. 101, pp. e215–e220, 2000.

[17] 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.