TinySleepNet: An Efficient Deep Learning Model for Sleep Stage Scoring based on Raw Single-Channel EEG

Akara Supratak[1,*] and Yike Guo[2]

***Abstract*— Deep learning has become popular for automatic sleep stage scoring due to its capability to extract useful features from raw signals. Most of the existing models, however, have been overengineered to consist of many layers or have introduced additional steps in the processing pipeline, such as converting signals to spectrogram-based images. They require to be trained on a large dataset to prevent the overfitting problem (but most of the sleep datasets contain a limited amount of class-imbalanced data) and are difficult to be applied (as there are many hyperparameters to be configured in the pipeline). In this paper, we propose an efficient deep learning model, named *TinySleepNet*, and a novel technique to effectively train the model end-to-end for automatic sleep stage scoring based on raw single-channel EEG. Our model consists of a less number of model parameters to be trained compared to the existing ones, requiring a less amount of training data and computational resources. Our training technique incorporates data augmentation that can make our model be more robust the shift along the time axis, and can prevent the model from remembering the sequence of sleep stages. We evaluated our model on seven public sleep datasets that have different characteristics in terms of scoring criteria and recording channels and environments. The results show that, with the same model architecture and the training parameters, our method achieves a similar (or better) performance compared to the state-of-the-art methods on all datasets. This demonstrates that our method can generalize well to the largest number of different datasets.**

I. INTRODUCTION

Polysomnography (PSG) is a sleep study to objectively assess the quality of sleep, which plays a vital role in promoting mental and physical health [1]. People with sleep disorders, such as sleep apnea, are diagnosed with PSG to evaluate the severity of their symptoms [2]. In the sleep study, a number of sensors are attached to different parts of the body to record what is called polysomnogram (also abbreviated PSG). The PSG mainly consists of an electroencephalogram (EEG), an electrooculogram (EOG), an electromyogram (EMG), and an electrocardiogram (ECG). The PSG is typically segmented into epochs of 20 or 30 seconds, which are then classified into different sleep stages by sleep experts according to sleep manuals, such as the Rechtschaffen and Kales (R&K) [3] and the American Academy of Sleep Medicine (AASM) [4]. This manual process is known as sleep stage scoring or sleep stage classification. However, it is labor-intensive, time-consuming, and prone to human errors, as the experts have to go through a large number of PSG records. With a limited number of experts, it cannot be applied on a large scale.

[1]Faculty of Information and Communication Technology (ICT), Mahidol University, Thailand (e-mail: akara.sup@mahidol.edu).

[2]Data Science Institute, Imperial College London, London, SW7 2AZ, UK (e-mail: y.guo@imperial.ac.uk)

[*]Corresponding author.

Recently, deep learning has become popular for automatic sleep stage scoring due to its capability to extract useful features from raw signals automatically, making the process of hand-engineering features no longer necessary. Many studies have employed convolutional neural networks (CNNs) in the top layers to transform from raw PSG epochs into useful features and bidirectional recurrent neural networks (RNNs) in the bottom layers to learn temporal information such as transition rules [5]–[7]. However, these models consist of too many layers and parameters that require to be trained on large sleep data to prevent the overfitting problem [8], in which most of the sleep datasets have only a limited amount of data. Also, the bidirectional RNNs require extra resources to buffer PSG epochs to operate in both forward and backward directions, which is problematic to be adopted in wearable devices. Several studies have proposed to transform raw PSG epochs into spectral-based images, typically small ones, before training deep learning models [9]. Even though this image-based approach can help reduce the size of the model, it introduces an additional step for signal conversion in the pipeline that is not part of the model. It then becomes difficult to be applied as there are many hyperparameters in the pipeline to be configured for the models to perform well on target datasets.

In this paper, we propose an efficient deep learning model, named *TinySleepNet*[1], and a novel training technique to address the problems mentioned above. Our model is an improved version of the DeepSleepNet model [5] for automatic sleep stage scoring based on raw single-channel EEG. The main contributions of this work are as follows:

- We develop an *efficient* model architecture that significantly reduces the number of model parameters and computational resources required to score EEG epochs, compared to the previous version.
- We develop a novel training technique that utilizes *data augmentation* to generate different training sets for different training epochs to alleviate the overfitting problem and help the trained model to be more robust to new patterns.
- We demonstrate that, with the same model architecture and the training parameters, our model can achieve a similar (if not better) performance compared to the state-of-the-art methods on seven public sleep datasets, which have different characteristics in terms of scoring criteria and recording channels and environments.

[1]Our code will be publicly available after publication.

978-1-7281-1990-8/20/$31.00 ©2020 IEEE

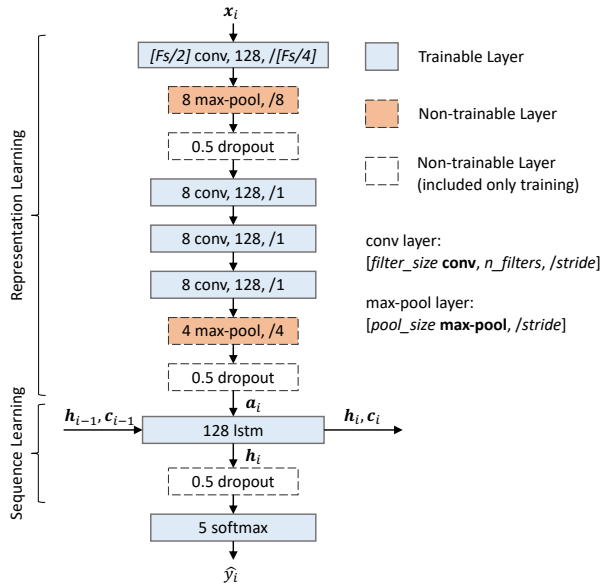

Fig. 1: An overview architecture of TinySleepNet. Each rectangular box represents one layer in the model, and the arrows indicate the flow of data from raw single-channel EEG epochs ($\mathbf{x}_i$) to sleep stages ($\hat{y}_i$).

II. METHODS

Our TinySleepNet model processes a sequence of single-channel EEG epochs and produces a sequence of sleep stages of the same length in the *many-to-many* scheme (see Fig. 1). Formally, suppose there are N EEG epochs $\{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$ from single-channel EEG, where $\mathbf{x}_i \in \mathbb{R}^{E_s \times F_s}$, E_s is the duration of an EEG epoch in seconds and F_s is the sampling rate. Our model f_θ determines sleep stages for all epochs, resulting in N predicted sleep stages $\{\hat{y}_1, \ldots, \hat{y}_N\}$, where $\hat{y}_i$ is the predicted sleep stage of $\mathbf{x}_i$, and $\hat{y}_i \in \{0, 1, 2, 3, 4\}$ corresponding to the five sleep stages W, N1, N2, N3 and REM, respectively, following the AASM manual [4].

A. Representation Learning

The first part of the network close to the input signals is a CNNs. The CNNs consists of four convolutional layers, interleaved with two max-pooling and two dropout layers. This part is used to extract time-invariant features from raw EEG signals. Formally, it extracts the i-th feature $\mathbf{a}_i$ from the i-the EEG epoch $\mathbf{x}_i$ as follows:

$$\mathbf{a}_i = \text{CNN}_{\theta_r}(\mathbf{x}_i)\,, \tag{1}$$

where CNN_{θ_r} represents the CNNs that transforms from a single-channel EEG epoch into a feature vector, and θ_r is the learnable parameters of the CNNs. Note that the size of $\mathbf{a}_i$ varies depending on the sampling rate of the input EEG.

Unlike the previous work [5], we use only one branch of the CNNs, instead of two branches with small and large filters. This is motivated by the design patterns of VGGNet [10], in which a stack of *conv* layers has the same effective receptive fields as one *conv* layer with a lager filter. This means that the model should be able to construct a larger filter by combining several *conv* layers at the top layers (close to the input), but using a fewer number of parameters. Thus, we combine the two CNNs branches into one CNNs with a double number of small filters (i.e., from 64 to 128).

B. Sequence Learning

The second part of the network close to the output sleep stages is a unidirectional RNNs, consisting of a single LSTM layer followed by a dropout layer. This part is used to learn the temporal information of the input signals, such as sleep transition rules [4] that sleep experts use to determine the next possible sleep stages based on the previous stages. Formally, suppose there are N feature vectors from the CNNs $\{\mathbf{a}_1, \ldots, \mathbf{a}_N\}$ arranged sequentially. This part processes the i-th feature $\mathbf{a}_i$ as follows:

$$\mathbf{h}_i, \mathbf{c}_i = \text{RNN}_{\theta_s}(\mathbf{h}_{i-1}, \mathbf{c}_{i-1}, \mathbf{a}_i)\,, \tag{2}$$

where RNN_{θ_s} represents the RNNs that processes the sequences of features $\mathbf{a}_i$, θ_s is the learnable parameters of the RNNs, $\mathbf{h}_i$ and $\mathbf{c}_i$ are vectors of hidden and cell states of the *lstm* layer after processing the features $\mathbf{a}_i$, $\mathbf{h}_{i-1}$ and $\mathbf{c}_{i-1}$ are the hidden and cell states from the previous input, and $\mathbf{h}_0$ and $\mathbf{c}_0$ are the initial hidden and cell states that are set to $\vec{0}$.

Unlike the previous work [5], we use the unidirectional LSTMs, instead of bidirectional, to eliminate the needs for buffering a chunk of EEG epochs for processing in the backward direction. This reduces the computational resources required for the sequence learning approximately by half, as the model only needs to process the epochs in the forward direction. Also, as our model consists of not too many layers, the residual connection, employed in the previous version to solve the vanishing gradient problem, is removed.

C. Model Training with Data Augmentation

Our technique trains the model end-to-end via minibatch gradient descent, equipped with what is called, *signal* and *sequence* augmentation. Such data augmentation helps us to synthesize new training data from the original data for *every* training epoch. The weighted cross-entropy loss is also used to alleviate the class imbalance problem by making the model prioritize on N1 stage, which is significantly low compared to the other stages in datasets (see Table I), by setting the weight for N1 stage to 1.5 and others to 1.

In signal augmentation, the EEG signal from each sleep is shifted along the time axis. The shifting duration is uniformly sampled from a range of $\pm B_{sig}\%$ of the EEG epoch duration. Suppose $B_{sig} = 10$ and the duration of the EEG epoch is 30 seconds, the shifting duration will be randomly selected from the range of +3 and -3 s. This technique helps us synthesize new signal patterns for each training epoch.

In sequence augmentation, the starting point of the sequence of EEG epochs from each sleep is randomly chosen. In other words, a few EEG epochs at the beginning of the sequence are skipped by a random amount. The skipping amount is uniformly sampled from a range of 0 to B_{seq}, where 0 means no skipping (i.e., original sequence) and B_{seq} is the maximum skipping amount. This technique helps us generate new batches of multiple sequences of EEG epochs in the minibatch gradient descent.

With these two techniques, we can train the model to be more robust to the shift along the time axis introduced in the process of segmenting PSG recordings into 20s or 30s epochs, and can also prevent the model from remembering the sequences of sleep stages in the training data.

Unlike the previous works [5], this technique does not pretrain the network with the oversampled, class-balanced data. It trains the model with the original data that contains the actual sequences of sleep stages.

III. RESULTS

A. Datasets

We evaluated our model using seven public sleep datasets obtained from Montreal Archive of Sleep Studies (MASS) [11] and Sleep-EDF [12], [13]. These datasets have been collected in different environments and annotated with different sleep manuals, which can be used to demonstrate the generalizability of the model.

In MASS (cohort 1), there were PSG recordings from 200 subjects aged between 18-76 years (97 males and 103 females), gathered from different sleep research laboratories. These recordings were organized into five subsets, SS1-SS5, according to their research and acquisition protocols. The PSG recordings were segmented into 30s epochs (SS1 and SS3) or 20s epochs (SS2, SS4, and SS5). Such PSG epochs were manually labeled by experts according to the AASM (SS1 and SS3) or the R&K (SS2, SS4, and SS5) manuals. For each subset, we evaluated our model using the F4-EOG (Left) channel, except the SS4, where we instead used the C4-EOG (Left) as the F4 channel was not available. All EEG signals have a sampling rate of 256 Hz.

In Sleep-EDF, we used 153 PSG recordings from the study of age effects in healthy subjects (SC), collected from 78 subjects aged between 25-101 years (37 males and 41 females). The PSG recordings were segmented into 30s epochs and manually annotated by experts according to the R&K manual. We evaluated our model using the Fpz-Cz EEG channel provided in the PSG recordings, whose sampling rate is 100 Hz. There were long periods of awake (W) at the start and the end of each recording. We only included 30 minutes of such periods just before and after the sleep periods, as we were interested in sleep periods. To facilitate the comparison with the existing methods, we also used version 1 of the Sleep-EDF dataset published in 2013 before the expansion, in which there were 39 PSG recordings from the SC study, collected from 20 subjects.

For all datasets, we excluded movement artifacts at the beginning and the end of each sleep data that was labeled as MOVEMENT or UNKNOWN, as they did not belong to the five sleep stages. If the datasets were scored according to the R&K manual, we converted them to be the same as the AASM manual by merging the N3 and N4 stages into a single stage N3 to facilitate comparison across datasets [14]. It should also be emphasized that we did not apply any preprocessing techniques (e.g., filtering) to the original signals. Table I summarizes the number of subjects and the distribution of sleep stages of each dataset.

TABLE I: The number of subjects (N_s) and the distribution of sleep stages of each dataset.

Dataset	N_s	W	N1	N2	N3	REM	Total
Sleep-EDF-v1	20	10197	2804	17799	5703	7717	44220
Sleep-EDF	78	69824	21522	69132	13039	25835	199352
MASS-SS1	53	12242	7112	22167	3407	6365	51293
MASS-SS2	19	2827	1483	13090	4401	4910	26711
MASS-SS3	62	6442	4839	29802	7653	10581	59317
MASS-SS4	40	6701	4021	25807	8188	10593	55310
MASS-SS5	26	2972	1904	17064	6734	7735	36409

B. Experimental Setting

The k-fold cross-validation (CV) scheme was used to evaluate our model performance on the seven datasets. Our choices of k can be found in Table II. The k for Sleep-EDF-v1, Sleep-EDF, and MASS-SS3 were chosen to facilitate the comparison with the existing methods. The k for MASS-SS1 and MASS-SS4 were chosen, such that there were two subjects left out for testing. The k for MASS-SS2 and MASS-SS5 were chosen to leave only one subject out for testing as there are only a few subjects in the subsets.

In each fold, we further split 10% of the training set into a validation set for evaluating the training model. The model that achieved the best overall accuracy was kept for evaluation with the test set. The model was trained using the Adam optimizer for 200 epochs, where the learning rate, Adam's beta1, and beta2 were 10^{-4}, 0.9, and 0.999, respectively. We used the mini-batch size of 20 and the sequence length of 15. L2 weight decay was also applied in the loss function as a regularization term whose weight was 10^{-3}. The gradient clipping with the threshold of 5.0 was applied to prevent the exploding gradient problem when training RNNs. For data augmentation, we used $B_{sig} = 10$ and $B_{seq} = 5$ to specify the range of the time shift and the skipping amount to be [-10%,10%] and [0,5], respectively.

We evaluated the performance of our model using per-class precision (PR), per-class recall (RE), per-class F1-score (F1), macro-averaging F1-score (MF1), overall accuracy (ACC), and Cohen's Kappa coefficient (κ).

C. Results

Table II shows a performance comparison between our method and other sleep stage scoring methods across ACC, MF1, κ, and F1. We only considered the recent methods that utilize the deep learning model to extract features from raw single-channel EEGs (i.e., no hand-engineering features) and evaluate using independent training and test sets (i.e., exclude all epochs of the test subjects from the training set).

Overall, our method scores sleep stages with the ACC and the MF1 of more than 82% and 75, respectively, on all datasets, and achieves a similar (if not better) performance compared to the state-of-the-art methods evaluated on the same EEG channels and datasets. Our model also achieves such performance without sacrificing the performance on *any* sleep stages, especially N1 (the most difficult sleep stage to classify). This demonstrates that our model can score well on all sleep stages and does not favor the majority sleep stages than the minority ones. Also, Cohen's Kappa shows

TABLE II: Comparison between TinySleepNet (our method) and other methods that utilize the deep learning model to extract features from raw-single-channel EEGs across overall accuracy (ACC), macro F1-score (MF1), Cohen's Kappa (κ) and per-class F1-score (F1). The numbers in bold indicate the best performance metrics of all methods for each dataset.

Methods	Datasets	Manual	EEG Channels	F_s (Hz)	Epoch (sec)	k-fold CV	Test Epochs	Overall Metrics			Per-class F1-Score (F1)				
								ACC	MF1	κ	W	N1	N2	N3	REM
IITNet [6]	Sleep-EDF-v1	R&K	Fpz-Cz	100	30	20	42308	84.0	77.7	0.78	87.9	44.7	88.0	85.7	82.1
SeqSleepNet+ (FT) [9]	Sleep-EDF-v1	R&K	Fpz-Cz	100	30	20	-	85.2	79.6	0.79	-	-	-	-	-
SleepEEGNet [7]	Sleep-EDF-v1	R&K	Fpz-Cz	100	30	20	42308	84.3	79.7	0.79	89.2	**52.2**	86.8	85.1	**85.0**
DeepSleepNet [5]	Sleep-EDF-v1	R&K	Fpz-Cz	100	30	20	41950	82.0	76.9	0.76	84.7	46.6	85.9	84.8	82.4
Our method	Sleep-EDF-v1	R&K	Fpz-Cz	100	30	20	44220	**85.4**	**80.5**	**0.80**	**90.1**	51.4	**88.5**	**88.3**	84.3
SleepEEGNet [7]	Sleep-EDF	R&K	Fpz-Cz	100	30	10	195479	80.0	73.6	0.73	91.7	44.1	82.5	73.5	76.1
Our method	Sleep-EDF	R&K	Fpz-Cz	100	30	10	199352	**83.1**	**78.1**	**0.77**	**92.8**	**51.0**	**85.3**	**81.1**	**80.3**
Our method	MASS-SS1	AASM	F4-EOG (L)	256	30	27	51293	83.1	79.3	0.76	90.0	60.6	87.4	73.2	85.1
Our method	MASS-SS2	R&K	F4-EOG (L)	256	20	19	26711	82.6	75.5	0.75	76.6	48.2	87.8	80.5	84.3
IITNet [6]	MASS-SS3	AASM	F4-EOG (L)	256	30	31	57395	86.6	80.8	0.80	86.1	54.4	91.3	**86.0**	86.2
DeepSleepNet [5]	MASS-SS3	AASM	F4-EOG (L)	256	30	31	58600	86.2	81.7	0.80	87.3	59.8	90.3	81.5	**89.3**
Our method	MASS-SS3	AASM	F4-EOG (L)	256	30	31	59317	**87.5**	**83.2**	**0.82**	**87.3**	**62.7**	**91.8**	85.5	88.6
Our method	MASS-SS4	R&K	C4-EOG (L)	256	20	20	55310	84.0	78.0	0.77	79.8	50.2	88.9	82.4	88.5
Our method	MASS-SS5	R&K	F4-EOG (L)	256	20	26	36409	86.6	80.9	0.81	85.5	55.0	89.9	86.6	87.7

that the agreement between our model and the sleep experts is almost perfect (0.81-1) or substantial (0.61-0.8) [15].

IV. DISCUSSION

Among the existing methods that we compared, the SeqSleepNet+ (FT) [9] is the one that processes spectral-based images and employs the transfer learning technique to finetune the model pretrained on the whole MASS dataset to the Sleep-EDF-v1 dataset. The results show that our method, which processes raw signals, performs as well as the SeqSleepNet+ (FT), but utilizing only the available recordings on the Sleep-EDF-v1 (i.e., no pretraining on a large dataset). This may well be due to our data augmentation that helps synthesize new patterns of sleep data that are useful for training the model to score sleep stages. Our method eliminates the need for converting EEG signals into spectral-based images and enables us to train a deep learning model even when the number of PSG recordings is limited.

In terms of model parameters, we have significantly reduced the number of parameters and the computational resources by utilizing one CNNs branch with small filters and the unidirectional LSTM. The numbers of parameters for DeepSleepNet [5] and SleepEEGNet [7] are $\sim$21m and $\sim$2.6m, which are approximately 15 and 2 times larger than our model ($\sim$1.3m), respectively. We did not compare with the IITNet [6] as its number of parameters cannot be estimated from the literature, and SeqSleepNet+ [9] as it processes spectral-based images instead of raw EEG signals. In terms of computation resources, the unidirectional RNNs also reduces the resources required to handle temporal information approximately by half, as the model only processes the EEG epochs in the forward direction. These modifications yield a much smaller model that is less likely to overfit the training data and is a good choice to explore the possibility of remote sleep monitoring in the home environment. The results also show that our model can perform as well (if not better) as the others.

ACKNOWLEDGMENT

This project is supported by Mahidol University. The authors would like to thank Thanapon Noraset and Peter Haddawy from Mahidol University who reviewed this paper.

REFERENCES

[1] M. R. Irwin, "Why Sleep Is Important for Health: A Psychoneuroimmunology Perspective," *Annu. Rev. Psychol*, vol. 66, pp. 143–172, 2015.

[2] D. Shrivastava, S. Jung, M. Saadat, R. Sirohi, and K. Crewson, "How to interpret the results of a sleep study," *Journal of Community Hospital Internal Medicine Perspectives*, vol. 4, no. 5, p. 24983, 2014.

[3] A. Rechtschaffen and A. Kales, "A manual for standardized terminology, techniques and scoring system for sleep stages in human subjects," *Brain information service*, 1968.

[4] C. Iber, S. Ancoli-Israel, A. L. Chesson Jr., and S. F. Quan, *The AASM manual for the scoring of sleep and associated events*. American Academy of Sleep Medicine, Westchester, Illinois, 2007.

[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 Transactions on Neural Systems and Rehabilitation Engineering*, pp. 1–10, 2017.

[6] S. Back, S. Lee, H. Seo, D. Park, T. Kim, and K. Lee, "Intra- and Inter-epoch Temporal Context Network (IITNet) for Automatic Sleep Stage Scoring," in *arXiv preprint*, 2019.

[7] S. Mousavi, F. Afghah, and U. R. Acharya, "SleepEEGNet: Automated sleep stage scoring with sequence to sequence deep learning approach," *PLOS ONE*, vol. 14, no. 5, pp. 1–15, 2019.

[8] C. Zhang, S. Bengio, M. Hardt, B. Recht, and O. Vinyals, "Understanding deep learning requires rethinking generalization," in *Iclr*, 2016.

[9] H. Phan, O. Y. Chén, P. Koch, Z. Lu, I. McLoughlin, A. Mertins, and M. De Vos, "Towards More Accurate Automatic Sleep Staging via Deep Transfer Learning," in *arXiv preprint*, 2019.

[10] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," in *arXiv preprint*, 2014.

[11] C. O'Reilly, N. Gosselin, J. Carrier, and T. Nielsen, "Montreal Archive of Sleep Studies: an open-access resource for instrument benchmarking and exploratory research," *Journal of Sleep Research*, vol. 23, no. 6, pp. 628–635, 2014.

[12] B. Kemp, A. H. Zwinderman, B. Tuk, H. A. C. Kamphuisen, and J. J. L. Oberyé, "Analysis of a sleep-dependent neuronal feedback loop: The slow-wave microcontinuity of the EEG," *IEEE Transactions on Biomedical Engineering*, vol. 47, no. 9, pp. 1185–1194, 2000.

[13] A. L. Goldberger, L. Amaral, L. Glass, J. Hausdorff, P. Ivanov, R. Mark, J. Mietus, G. Moody, C.-K. Peng, and H. Stanley, "PhysioBank, PhysioToolkit, and PhysioNet: Components of a New Research Resource for Complex Physiologic Signals," *Circulation*, vol. 101, no. 23, pp. 215–220, 2000.

[14] S. A. Imtiaz and E. Rodriguez-Villegas, "Recommendations for performance assessment of automatic sleep staging algorithms," in *2014 36th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, 2014, pp. 5044–5047.

[15] A. R. Hassan and A. Subasi, "A decision support system for automated identification of sleep stages from single-channel EEG signals," *Knowledge-Based Systems*, vol. 0, pp. 1–10, 2017.