SleepEEGNet: Automated sleep stage scoring with sequence to sequence deep learning approach

Sajad Mousavi , Fatemeh Afghah, U. Rajendra Acharya

Published: May 7, 2019 • https://doi.org/10.1371/journal.pone.0216456

Abstract

Electroencephalogram (EEG) is a common base signal used to monitor brain activities and diagnose sleep disorders. Manual sleep stage scoring is a time-consuming task for sleep experts and is limited by inter-rater reliability. In this paper, we propose an automatic sleep stage annotation method called *SleepEEGNet* using a single-channel EEG signal. The *SleepEEGNet* is composed of deep convolutional neural networks (CNNs) to extract time-invariant features, frequency information, and a sequence to sequence model to capture the complex and long short-term context dependencies between sleep epochs and scores. In addition, to reduce the effect of the class imbalance problem presented in the available sleep datasets, we applied novel loss functions to have an equal misclassified error for each sleep stage while training the network. We evaluated the performance of the proposed method on different single-EEG channels (i.e., Fpz-Cz and Pz-Oz EEG channels) from the Physionet Sleep-EDF datasets published in 2013 and 2018. The evaluation results demonstrate that the proposed method achieved the best annotation performance compared to current literature, with an overall accuracy of 84.26%, a macro F1-score of 79.66% and $\kappa = 0.79$. Our developed model can be applied to other sleep EEG signals and aid the sleep specialists to arrive at an accurate diagnosis. The source code is available at https://github.com/SajadMo/SleepEEGNet.

Citation: Mousavi S, Afghah F, Acharya UR (2019) SleepEEGNet: Automated sleep stage scoring with sequence to sequence deep learning approach. PLoS ONE 14(5): e0216456. https://doi.org/10.1371/journal.pone.0216456

Editor: Paweł Pławiak, Politechnika Krakowska im Tadeusza Kosciuszki, POLAND

Received: March 12, 2019; **Accepted:** April 23, 2019; **Published:** May 7, 2019

Data Availability: All EEG files are available from the physiobank database (doi:10.13026/C2C30J). https://physionet.org/physiobank/database/sleep-edf/.

Funding: This material is based upon work supported by the National Science Foundation under Grant Number 1657260. Research reported in this publication was also supported by the National Institute On Minority Health and Health Disparities of the National Institutes of Health under Award Number U54MD012388. The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Competing interests: The authors have declared that no competing interests exist.

Introduction

The electroencephalogram (EEG), electrooculogram (EOG), and electromyogram (EMG) signals are widely used to diagnose the sleep disorders (e.g., sleep apnea, parasomnias, and hypersomnia). These signals are typically recorded by placing sensors on different parts of the patient's body. In an overnight polysomnography (PSG) (also called as sleep study), the EEG signal is usually the main collected signal being used to monitor the brain activities to diagnose different sleep disorders [1] and other common disorders such as epilepsy [2].

The EEG signals are split into a number of predefined fixed length segments which are termed as epochs. Then, a sleep expert manually labels each epoch according to sleep scoring standards provided by the American Academy of Sleep Medicine (AASM) [3] or the Rechtschaffen and Kales standard [4]. Each EEG recording is around 8-hour long on average. Therefore, the manual scoring of such a long signal for a sleep expert is a tedious and time-consuming task. The human-based annotation methods also highly rely on an inter-rater agreement in place. Therefore, such restrictions call for automated sleep stage classification system that is able to score each epoch automatically with a high accuracy.

Several studies have focused on developing automated sleep stage scoring algorithms. Generally, they can be divided into two different categories in terms of the feature extraction approaches. First, the hand-engineered feature-based methods that require a prior knowledge of EEG analysis to extract the most relevant features. These approaches first extract the most common features such as time, frequency and time-frequency domain features of single channel-EEG waveforms [5–7]. Then, they apply conventional machine learning algorithms such as support vector machines (SVM) [8], random forests [9] and neural networks [10] to train the model for sleep stage classification based on the extracted features. Although these methods have achieved a reasonable performance, they carry several limitations including the need for a prior knowledge of sleep analysis and are not able to generalize to larger datasets from various patients with different sleep patterns. The second category includes the automated feature extraction-based methods such as deep learning algorithms, in which the machine extracts the pertaining features automatically (e.g., the CNNs to extract time-invariant features from raw EEG signals).

In recent years, deep neural networks have shown impressive results in various domains ranging from computer vision and reinforcement learning to natural language processing [11–14]. One key reason for the success of deep learning based methods in these domains is the availability of large amounts of data to learn the underlying complex pattern in the data sets. Due to availability of a large number of sleep EEG recordings [15], deep learning algorithms have also been applied for sleep stage classification [1, 16–19]. However, in spite of the remarkable achievements in using deep learning models compared to the shallow machine learning methods for sleep stage scoring task, they still suffer from the class imbalance problem present in the sleep datasets. Thus, this problem can limit the use of deep learning techniques and in general machine learning techniques toward reaching an expert-level performance for sleep stage classification.

The sleep is a cyclical process. Typically, a sleeper experiences five main sleep stages during his sleep time, including *wake*, *N1*, *N2*, *N3*, and rapid eye movement (*REM*) stages. Usually, each sleep cycle goes through the Non-REM (Stages 1, 2 and 3) sleep to REM sleep. In most cases, the cycle takes 90-120 minutes resulting in four to five cycles per night [20]. Hence, we believe the sleep stage classification problem is sequential in nature and taking into account this sequential characteristic by considering the correlation between different stages can enhance the accuracy of sleep stage scoring process. Therefore, it is essential to propose a sleep stage scoring system with the capability of extracting non-linear dependencies present in the consecutive stages during scoring different stages. In this paper, we introduced a novel deep learning approach, called *SleepEEGNet*, for automated sleep stage scoring using a single-channel EEG. In this model, we applied a sequence to sequence deep learning model with the following building blocks: (1) CNNs to perform the feature extraction, (2) a bidirectional recurrent neural network (BiRNN) to capture temporal information from sequences and consider the previous and future input information simultaneously, and (3) an attention network to let the model learn the most relevant parts of the input sequence while training. Also, we utilized new loss functions to reduce the effect of imbalance class problem on the model by treating the error of each misclassified sample equally regardless of being a member of the majority class or minority class.

The main contributions of our study are as follows:

- We propose a sequence to sequence deep learning approach along with the BiRNN and attention mechanism that suits best for the sleep stage scoring problem.
- We apply novel loss functions to address the imbalance class problem.
- The proposed model is an end-to-end deep learning approach that uses raw single-channel EEGs as its input without using any handcrafted features and significant signal pre-processing such as filtering or noise removal methods.

The rest of the paper is structured as follows: Methodology section introduces the proposed method. Dataset and Data Preparation section describes the utilized datasets and the data preparation techniques. Experimental Results section presents the experimental design and shows the achieved results by the proposed method along with a performance comparison to the state-of-the-art algorithms. Finally, Conclusion section concludes the paper.

Methodology

In the following sections, we present a detailed description of our proposed model developed to automatically score each sleep stage from a given EEG signal.

Pre-processing

The input to this method is a sequence of 30-s EEG epochs. In order to extract the EEG epochs from a given EEG signal, we follow two simple steps:

1. Segmenting the continuous raw single-channel EEG to a sequence of 30-s epochs and assigning a label to each epoch (i.e., sleep stage) based on the annotation file.
2. Normalizing 30-s EEG epochs such that each one has a zero mean and unit variance.

It is worth mentioning that, these pre-processing steps for the sleep stage extraction are very simple and do not involve any form of filtering or noise removal methods.

The architecture

The sequence to sequence models have shown very impressive results in neural machine translation applications, nearly similar to human-level performance [21]. The architecture of sequence to sequence networks is usually composed of two main parts: the encoder and decoder which are types of recurrent neural network (RNN). In this study, we used an RNN sequence to sequence model along with a convolutional neural network (CNN) to perform automatic sleep stage scoring task.

Fig 1 illustrates the proposed network architecture for automatic sleep stage classification. We applied almost the same CNN architecture provided by [17]. The CNN consists of two sections, one with small filters to extract temporal information and another one with large filters to extract frequency information. The idea behind these variable sizes of filters comes from the signal processing community to have a trade-off between extracting time domain (i.e., time-invariant) and frequency domain features [22]. This helps get benefit from both time and frequency domain features in the classification task.

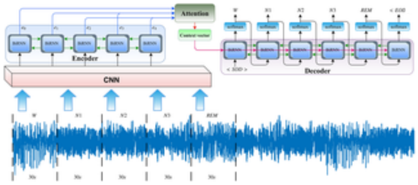


Fig 1. Illustration of the proposed sequence to sequence deep learning network architecture for automated sleep stage scoring.
The input signal is a sequence of 30-s EEG epochs and the outputs are their corresponding stages (or classes) generated by our proposed method.
https://doi.org/10.1371/journal.pone.0216456.g001

Each CNN part consists of four consecutive one-dimensional convolutional layers. Each convolutional layer is passed to a rectified linear unit (ReLU) nonlinearity. The first layer is followed by a max pooling layer and a dropout block, and just a dropout block comes after the last convolutional layer. At each time-step of training/testing the model, a sequence (size of *maxtime*) of 30-s EEG epochs is fed into the CNN for feature extraction. In the end, the outputs of CNN parts are concatenated serially and followed by a dropout block in order for the encoder network to encode the sequence input. Fig 2 depicts the detailed CNN structure.

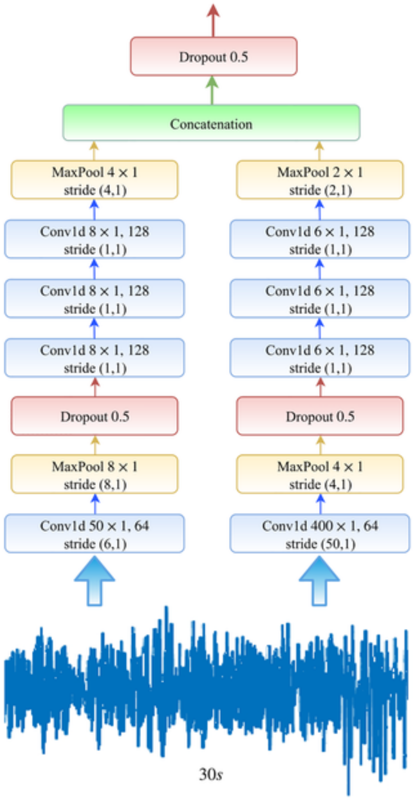


Fig 2. Detailed sketch of the utilized CNN model in the proposed work.
https://doi.org/10.1371/journal.pone.0216456.g002

The sequence to sequence model is designed based on the encoder-decoder abstract ideas. The encoder encodes the input sequence, while the decoder computes the category of each single channel 30-s EEG of the input sequence. The encoder is composed of long short-term memory (LSTM) units which capture the complex and long short-term context dependencies between the inputs and the targets [23]. They capture non-linear dependencies present in the entire time series while predicting a target. The (time) sequence of input feature vectors herein are fed to the LSTMs and then the hidden states, (e_0, e_1, e_2, …), calculated by the LSTM are considered as the encoder representation, and are fed to the attention network (or to initialize the first hidden state of the decoder, if the basic decoder is used), as depicted in Fig 1.

Bidirectional recurrent neural network

We have utilized the bidirectional recurrent neural network (BiRNN) units in the network architecture instead of the standard LSTM (i.e., standard RNN). Standard RNNs are unidirectional, hence they are restricted to use the previous input state. To address this limitation, the BiRNN have been proposed [24], which can process data in both forward and backward directions. Thus, the current state has access to previous and future input information simultaneously. The BiRNN consists of forward and backward networks. The input sequence is fed in normal time order, t = 1, …, T for the forward network, and in reverse time order, t = T, …, 1 for the backward network. Finally, the weighted sum of the outputs of the two networks is computed as the output of the BiRNN. This mechanism can be formulated as follows:

$$\overrightarrow{h_t} = \tanh(\overrightarrow{W}x_t + \overrightarrow{V}\overrightarrow{h}_{t-1} + \overrightarrow{b}) \tag{1}$$

$$\overleftarrow{h_t} = \tanh(\overleftarrow{W}x_t + \overleftarrow{V}\overleftarrow{h}_{t+1} + \overleftarrow{b}) \tag{2}$$

$$y_t = (U[\overrightarrow{h_t}; \overleftarrow{h_t}] + b_y), \tag{3}$$

where ($\overrightarrow{h_t}$, $\overrightarrow{b}$) are the hidden state and the bias of the froward network, and ($\overleftarrow{h_t}$, $\overleftarrow{b}$) are the hidden state and the bias of the backward network. Also, x_t and y_t are the input and the output of the BiRNN, respectively. Fig 3 illustrates a BiRNN architecture with T time steps.

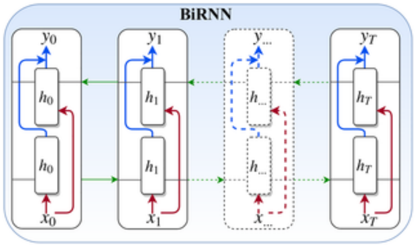


Fig 3. A schematic diagram of the bidirectional recurrent neural network.
https://doi.org/10.1371/journal.pone.0216456.g003

Attention decoder

The decoder is used to generate the target sequence epoch by epoch. Similar to the encoder, the building block of the decoder is an LSTM. In a basic decoder, at every step of decoding, the decoder gets a new representation of an input element of the sequence generated by the encoder and an element of the target input. The last element of the input sequence is usually the last influence to update the encoder's hidden state. Therefore, the model can be biased to the last element. To address such a problem, we have applied an attention mechanism to the model to consider not only the whole encoder representation of the input but also to put more emphasis on different parts of the encoder outputs in each step of decoding. In other words, the attention mechanism makes the model to learn the most relevant parts of the input sequence in the decoding phase. In a sequence to sequence model without an attention approach, the decoder part relies on the hidden vector of the decoder's RNN (or BiRNN), while the sequence to sequence model with the attention is more goal-oriented by putting attention on the most related input regions to produce the targets. It considers the combination of encoder's representation and decoder hidden vector calling the context vector or the attention vector, (c_t).

To calculate the c_t vector, we first computed a set of attention weights with a function f(.) followed by a softmax function. These attention weights are probabilities, (α_i), corresponding to the importance of each hidden state. Then, these scores are multiplied by the hidden states (i.e, the encoder output vectors) to calculate the weighted combination, (c_t).

$$f(h_{t-1}, e_i) = \tanh(W_h h_{t-1} + W_e e_i) \tag{4}$$

$$\alpha_i = \quad \mathrm{softmax}(f(h_{t-1}, e_i)) \approx \frac{\exp(f(h_{t-1}, e_i))}{\sum_{j=1}^{n} \exp(f(h_{t-1}, e_j))}, i \quad \in 1, 2, \ldots, n, \tag{5}$$

$$c_t \quad = \sum_{i=0}^{n} \alpha_i e_i, \tag{6}$$

where α_i is a parameter reflecting the importance of part i of hidden state e_i. In other words, at every time step t, the attention layer computes f(.), a combination of the values of e_j (the encoder's hidden state) and h_{t-1} (the decoder's hidden state) followed by a tanh layer. Then, the f(.) output is fed into a softmax module to calculate α_i over n parts. Finally, the attention module computes c_t, a weighted sum of all vectors e_i, $i \in 1, 2, \ldots, n$ based on computed α_i's. Thus, the model can learn to focus on the important regions of the input sequence when decoding.

During the training phase, the decoder, in addition to the augmented version of the encoder's hidden states, captures the given target sequence shifted by one starting with a special feature vector < *SOD* > (i.e., the start of decoding) as its input. Then, the decoder starts to generate outputs until it confronts the special label called < *EOD* > (i.e., the end of decoding). We should note that the target sequence is just used during the training phase and is not applied for the testing phase. During the testing phase, the decoder uses whatever label it generates at each step as the input for the next step. Finally, a softmax is applied to the output of the decoder to convert it to a vector of probabilities $p \in [0, 1]^C$, where C represents the number of classes and each element of p indicates the probability of each class of the sleep stage.

Loss calculation

Similar to other biomedical applications, the sleep stage classification also deals with the problem of class imbalance. To alleviate the effect of this problem on the model, we calculated new loss functions based on [25] to treat the error of each misclassified sample equally regardless of being a member of the majority or minority class.

We extended the proposed loss functions, mean false error (MFE) and mean squared false error (MSFE), in [25] for the multi-class classification task. MFE and MSFE can be defined as follows:

$$l(c_i) = \frac{1}{C_i} \sum_{i=j}^{C_i} (y_j - \hat{y}_j)^2 , \tag{7}$$

$$l_{MFE} = \sum_{i=1}^{N} l(c_i), \tag{8}$$

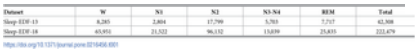
$$l_{MSFE} = \sum_{i=1}^{N} l(c_i)^2, \tag{9}$$

where c_i is the class label (e.g., W or N1), C_i is the number of the samples in class c_i, N is the number available classes (here sleep stage classes), and $l(c_i)$ is the calculated error for the class c_i. In the most common used loss function, mean squared error (MSE), the loss is calculated by averaging the squared difference between predictions and targets. This way of computing the loss function makes the contribution of the majority classes be much more in comparison with the minorities classes in the imbalanced dataset. However, the MFE and MSFE try to consider the errors of all classes equally.

Dataset and data preparation

In this study, we used the Physionet Sleep-EDF dataset [15, 26]: version 1 contributed in 2013 with 61 polysomnograms (PSGs) and version 2 contributed in 2018 with 197 PSGs to evaluate the performance of the proposed method for the sleep stage scoring task. For simplicity's sake, the Sleep-EDF-13 and the Sleep-EDF-18 are used for versions 1 and 2, respectively. Sleep-EDF 2013 to The Sleep-EDF dataset contains two different studies including (1) study of age effects on sleep in healthy individuals (SC = Sleep Cassette) and (2) study of temazepam effects on sleep (ST = Sleep Telemetry). The dataset includes whole-night polysomnograms (PSGs) sleep recordings at the sampling rate of 100 Hz. Each record contains EEG (from Fpz-Cz and Pz-Oz electrode locations), EOG, chin electromyography (EMG), and event markers. Few records often also contain oro-nasal respiration and rectal body temperature. The hypnograms (sleep stages; 30-s epochs) were manually labeled by well-trained technicians according to the Rechtschaffen and Kales standard [4].

Each stage was considered to belong to a different class (stage). The classes include wake (W), rapid eye movement (REM), N1, N2, N3, N4, M (movement time) and '?' (not scored). According to American Academy of Sleep Medicine (AASM) standard, we integrated the stages of N3 and N4 in one class named N3 and excluded M (movement time) and ? (not scored) stages to have five sleep stages [3]. Stages 1 and 2-3 are the light sleep time in which the stage N1 is the lightest stage and has a short period time. The stage N2-N3 takes longer than the stage N1, including approximately 40-60% of total sleep time. The stage N3 is called as deep sleep and the REM is known as the dreaming stage taking 90-120 minutes per night [20]. Considering different stage time periods results in having a imbalanced stage numbers in the sleep datasets. In addition, we considered Fpz-Cz/Pz-Oz EEG channels from SCs of both versions in our evaluations. Table 1 presents the number of sleep stages in two different versions.



Dataset	W	N1	N2	N3-N4	REM	Total
Sleep-EDF-13	8,285	2,804	17,799	5,703	7,717	42,308
Sleep-EDF-18	65,951	21,522	96,132	13,039	25,835	222,479

https://doi.org/10.1371/journal.pone.0216456.t001

Table 1. Details of number of sleep stages in each version of Sleep-EDF dataset.
https://doi.org/10.1371/journal.pone.0216456.t001

Experimental results

Experimental design

The distribution of sleep stages in the Sleep-EDF database is not uniform. Hence, the number of W and N2 stages are much greater than other stages. The machine learning approaches do not perform well with the class imbalance problem. To address this problem, in addition to using the novel loss functions described in Loss calculation section, the dataset is oversampled to nearly reaching a balanced number of sleep stages in each class. We have used the synthetic minority over-sampling technique (SMOTE) to generate the synthetic data points by considering the similarities between existing minority samples [27].

Our proposed model was evaluated using a k-fold cross-validation. We set k to 20 and 10 for version 1 and version 2 of the Sleep-EDF dataset, respectively. In other words, we split the dataset into k folds. Then, for each unique fold, (1) the fold is taken as test set and the remaining folds as a training set and (2) trained the model using the training set and evaluated the model using the test set. Finally, all evaluation results were combined.

The network was trained (for each dataset) with a maximum of 120 epochs. RMSProp optimizer was used to minimize the l_{MFE} loss with mini batches of size 20 and a learning rate of $\alpha = 0.001$. We also applied an additional L_2 regularization element with $\beta = 0.001$ to the loss function to mitigate the overfitting. Python programming language and Google Tensorflow deep learning library were utilized to implement our proposed approach. We ran the k-fold cross validation on a machine with 8 CPUs (Intel(R) Xeon(R) CPU @ 3.60 GHz), 32 GB memory and Ubuntu 16.04. The training time for each epoch was 98 seconds on average and the testing time for each batch of 20 EEG epochs was approximately 0.102 seconds.

Evaluation metrics

We have used different metrics to evaluate the performance of the proposed approach including, overall accuracy, precision, recall (sensitivity), specificity, and F1-score. We also computed macro-averaging of F1-score (MF1) which is the sum of per-class F1-scores over the number of classes (i.e., sleep stages). These metrics are defined as follows:

$$Accuracy = \frac{TP + TN}{TN + FP + FP + FN} \tag{10}$$

$$Precision = TP/(TP + FP) \tag{11}$$

$$Recall = TP/(TP + FN) \tag{12}$$

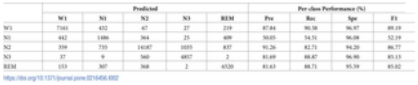
$$Specificity = TN/(TN + FP) \tag{13}$$

$$F_1\,score = 2\frac{Precision \times Recall}{Precision + Recall} \tag{14}$$

where TP (True Positive), TN (True Negative), FP (False Positive) and FN (False Negative) indicate the number of sleep stages correctly labeled, the number of sleep stages correctly identified as not correspond to the sleep stages, the number of sleep stages that incorrectly labeled, and the number of sleep stages which were not identified as the sleep stages that they should have been, respectively. The other main metric that we have used for performance evaluation of our proposed method is Cohen's Kappa coefficient (κ). When two persons (algorithms or raters) try to measure the same data, the Cohen's Kappa coefficient, κ, is used as a measure of agreement between their decisions. For example, in this study, we aim to measure the amount of agreement between our algorithm as one rater and the provided labels for sleep stages by the dataset as another rater.

Results and discussion

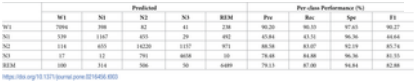
Tables 2 and 3 present the confusion matrices and the performances of each class achieved by the proposed method using Fpz-Cz and Pz-Oz channels of the EDF-Sleep-2013 data set, respectively. The main diagonals in each confusion matrix denote the true positive (TP) values which indicate the number of stages scored correctly. It can be seen from the tables (the confusion matrices' parts) that TP values are higher than other values in the same rows and columns. These tables also show the prediction performance (i.e., the precision, recall, specificity and F1 score) of our model for each class (i.e., the stage). Among all stages, the model performance is better for W1, N2, N3, and REM stages than the N1 stage. This may be because the number of N1 stages in the dataset is smaller compared to the other stages. However, our results for stage N1 is better than other state-of-the-art algorithms listed in Table 4.



	Predicted					Per-class Performance (%)			
	W1	N1	N2	N3	REM	Pre	Rec	Spe	F1
W1	7161	432	67	27	219	87.84	90.58	96.97	89.19
N1	442	1486	364	25	409	50.05	54.51	96.08	52.19
N2	359	735	14187	1035	837	91.26	82.71	94.20	86.77
N3	37	9	560	4857	2	81.69	88.87	96.90	85.13
REM	153	307	368	2	6520	81.63	88.71	95.59	85.02

https://doi.org/10.1371/journal.pone.0216456.t002

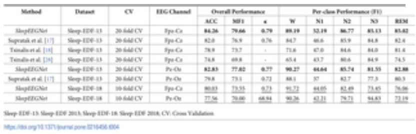
Table 2. Confusion matrix and per-class performance achieved by the proposed method using Fpz-Cz EEG channel of the EDF-Sleep-2013 database.
https://doi.org/10.1371/journal.pone.0216456.t002



	Predicted					Per-class Performance (%)			
	W1	N1	N2	N3	REM	Pre	Rec	Spe	F1
W1	7094	398	82	41	238	90.20	90.33	97.63	90.27
N1	539	1167	455	29	492	45.84	43.51	96.36	44.64
N2	114	655	14220	1157	971	88.58	83.07	92.19	85.74
N3	17	12	791	4658	10	78.48	84.88	96.36	81.55
REM	100	314	506	50	6489	79.13	87.00	94.84	82.88

https://doi.org/10.1371/journal.pone.0216456.t003

Table 3. Confusion matrix and per-class performance achieved by the proposed method using Pz-Oz EEG channel of the EDF-Sleep-2013 database.
https://doi.org/10.1371/journal.pone.0216456.t003



Method	Dataset	CV	EEG Channel	Overall Performance			Per-class Performance (F1)				
				ACC	MF1	κ	W	N1	N2	N3	REM
SleepEEGNet	Sleep-EDF-13	20-fold CV	Fpz-Cz	**84.26**	**79.66**	**0.79**	**89.19**	**52.19**	**86.77**	**85.13**	**85.02**
Supratak et al. [17]	Sleep-EDF-13	20-fold CV	Fpz-Cz	82.0	76.9	0.76	84.7	46.6	85.9	84.8	82.4
Tsinalis et al. [16]	Sleep-EDF-13	20-fold CV	Fpz-Cz	78.9	73.7	-	71.6	47.0	84.6	84.0	81.4
Tsinalis et al. [28]	Sleep-EDF-13	20-fold CV	Fpz-Cz	74.8	69.8	-	65.4	43.7	80.6	84.9	74.5
SleepEEGNet	Sleep-EDF-13	20-fold CV	Pz-Oz	**82.83**	**77.02**	**0.77**	**90.27**	**44.64**	**85.74**	**81.55**	**82.88**
Supratak et al. [17]	Sleep-EDF-13	20-fold CV	Pz-Oz	79.8	73.1	0.72	88.1	37	82.7	77.3	80.3
SleepEEGNet	Sleep-EDF-18	10-fold CV	Fpz-Cz	80.03	73.55	0.73	91.72	44.05	82.49	73.45	76.06
SleepEEGNet	Sleep-EDF-18	10-fold CV	Pz-Oz	77.56	70.00	68.94	90.26	42.21	79.71	94.83	72.19

Sleep-EDF-13: Sleep-EDF 2013; Sleep-EDF-18: Sleep-EDF 2018; CV: Cross Validation

https://doi.org/10.1371/journal.pone.0216456.t004

Table 4. Comparison of performance obtained by our approach with other state-of-the-art algorithms.
https://doi.org/10.1371/journal.pone.0216456.t004

Typically, there are two approaches to evaluate the proposed methods in the literature: (i) intra-patient paradigm in which the training and evaluation sets can include epochs from the same subjects, and (ii) inter-patient paradigm in which the epochs sets for test and training come from different individuals. As the inter-patient scheme seems to be a more realistic evaluation mechanism, the results and comparisons presented in this study are based on the inter-patient paradigm. Table 4 presents the comparison of stage sleep scoring results for the proposed method with the existing algorithms. It can be noted from Table 4 that the proposed model outperformed the state-of-the-art algorithms presented in the table. Our model has performed better in all listed channels (i.e., the Fpz-Cz and the Pz-Oz EEG channels) in terms of all evaluation metrics compared to others. According to Table 4, the results for Fpz-Cz channel are better than Pz-Oz channel. The reason is that Fpz-Cz channel position captures most of the frequencies including delta activity, K-complexes, lower frequency sleep spindles (predominantly frontal phenomena) that are important for sleep staging. However, Pz-Oz channel position extracts Theta activity and higher frequency sleep spindles (predominantly parietal phenomena) [18].

Furthermore, it may be noted that in spite of the imbalance-class problem, our model yielded desirable performance, especially for stage N1. In addition to the Sleep-EDF 2013 dataset, we also evaluated our model with the Sleep-EDF 2018 dataset. Since the dataset has been published recently, we could not find any work to compare the performance of our model. Therefore, we just reported our findings without any comparison.

Fig 4a (left) shows the performance graph of the accuracy. It is shown that the model can offer a comparable performance on both training and test sets. Also, we can see that the test accuracy is greater than the training accuracy meaning the network has generalized very well. Fig 4b (right) illustrates the performance graphs of the loss function. From Fig 4b (right), we can see that the loss curves grow constantly at the final epochs. This means that we should stop training.

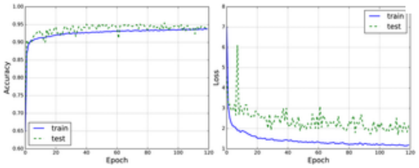


Fig 4. Graphs of the performance of the accuracy (a) and the loss function (b) of the proposed model in each epoch for a randomly selected fold (i.e., the fold 4).
https://doi.org/10.1371/journal.pone.0216456.g004

Fig 5 illustrates the hypnogram produced manually by a sleep expert and its corresponding hypnogram generated by our method for a subject for approximately 8 hours of sleep at night. It can be noted from the figure that around 85% the manually scored hypnogram and automatically scored correctly.

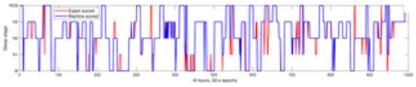


Fig 5. A example of hypnograms generated by the machine (i.e., the proposed method) and a sleep expert of a subject from the Sleep-EDF-13 dataset; approximately 85% coverage.
https://doi.org/10.1371/journal.pone.0216456.g005

Furthermore, by employing the attention mechanism into the network, we are able to illustrate (in the form of attention maps) which input epochs are important to score the sleep stages. As shown in Fig 6, we can see the network used almost the exact input epoch to predict its corresponding sleep stage (on the diagonal of each figure). The higher brightness indicates more attention (i.e., the amount of each square brightness shows the importance of its corresponding input (on the x-axis in the figure)) to generate each sleep stage. For example, Fig 6a (left) shows that to score the input epoch 8 (on the x-axis) the important epochs are its previous (epoch 7, N1) and next (epoch 9, N2) epochs, and specially its epoch as the corresponding square in the attention map is brighter.

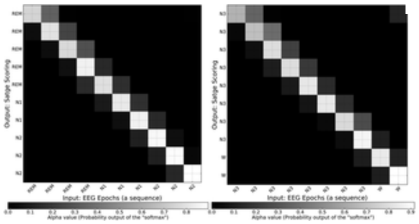


Fig 6. Attention maps of two sequence inputs (EEG epochs) and their corresponding sleep stage scores provided by our proposed method.
https://doi.org/10.1371/journal.pone.0216456.g006

Our model has performed better than the rest of the works due to the following two reasons: First, the nature of the sleep stage scoring task is sequential in which each sleep stage has a relationship with the previous and next stage. Hence, applying a sequence to sequence deep learning model for such a problem would be a desirable choice. Also, using the attention model and BiRNNs as the building blocks of the sequence to sequence model enhanced the performance. Second, the sleep stage classification suffers from the imbalance-class problem. To reduce the effect of this problem, we applied new loss functions (i.e., the MFE and MSFE) to have an equal misclassified error effect for each sleep stage while training the network.

One of the remarkable aspects of our proposed method is that, the model is generic in nature hence it generalizes for other problems in the biomedical signal processing applications that are inherently sequential and have the imbalance-class problem such as the heartbeat classification for arrhythmia detection [29, 30].

Even though our proposed model achieved significant results compared to the existing methods for the sleep stage classification, the model still carries several limitations. First, similar to other deep learning methods, our method needs a sufficient amount of sleep stage samples in training phase to learn discriminative features of each stage. Second, as our model is a sequence to sequence approach, at each time step, it requires to have a certain amount of 30-s EEG epochs (as input sequence) to be able to score the input epochs. Finally, our proposed method is evaluated with two available EEG channels (i.e., Fpz-Cz and Pz-Oz EEG channels) extracted from the Physionet Sleep-EDF datasets. Therefore, to evaluate its performance on other EEG channels, the network has to be trained with new EEG epochs.

Furthermore, in future, we intend to extend this work using multimodal polysomnography (PSG) signals including EEG, EOG (electrooculography) and EMG (electromyogram) to boost the performance of the sleep stage classification.

Conclusion

We have presented a novel algorithm for automated sleep stage annotation problem. The proposed method leverages the ability of deep convolutional neural network and encoder-decoder network in which we have used bidirectional recurrent neural networks and attention mechanisms as its building blocks. The proposed new loss calculation approaches helped to reduce the effect of the class-imbalance problem and boost the performance, especially the performance of our method on the stage N1, that is more difficult than other sleep stages to score. Table 4 presents that, our proposed model significantly outperformed the existing algorithms by yielding the highest performance for the sleep stage scoring task. While developing the automated systems, generally there will be imbalance data problem (normal class more data than diseased class). Our developed model can be applied to such biomedical applications such as arrhythmia detection using ECG signals, epilepsy detection using EEG signals and EMG signals to study the postures.

References

1. Biswal S, Kulas J, Sun H, Goparaju B, Westover MB, Bianchi MT, et al. SLEEPNET: automated sleep staging system via deep learning. arXiv preprint arXiv:170708262. 2017;.

2. Acharya UR, Oh SL, Hagiwara Y, Tan JH, Adeli H. Deep convolutional neural network for the automated detection and diagnosis of seizure using EEG signals. Computers in biology and medicine. 2018;100:270–278. pmid:28974302
View Article • PubMed/NCBI • Google Scholar

3. Berry RB, Brooks R, Gamaldo CE, Harding SM, Marcus C, Vaughn B, et al. The AASM manual for the scoring of sleep and associated events. Rules, Terminology and Technical Specifications, Darien, Illinois, American Academy of Sleep Medicine. 2012;.

4. Rechtschaffen A. A manual for standardized terminology, techniques and scoring system for sleep stages in human subjects. Brain information service. 1968;.

5. Zafar R, Dass SC, Malik AS. Electroencephalogram-based decoding cognitive states using convolutional neural network and likelihood ratio based score fusion. PloS one. 2017;12(5):e0178410. pmid:28558002

View Article • PubMed/NCBI • Google Scholar

6. Zaeri-Amirani M, Afghah F, Mousavi S. A Feature Selection Method Based on Shapley Value to False Alarm Reduction in ICUs A Genetic-Algorithm Approach. In: 2018 40th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC); 2018. p. 319–323.
7. Afghah F., Razi A., Soroushmehr R., Ghanbari H., Najarian K. Game Theoretic Approach for Systematic Feature Selection; Application in False Alarm Detection in Intensive Care Units. Entropy. 2018; 3(190).
View Article • Google Scholar
8. Koley B, Dey D. An ensemble system for automatic sleep stage classification using single channel EEG signal. Computers in biology and medicine. 2012;42(12):1186–1195. pmid:23102750
View Article • PubMed/NCBI • Google Scholar
9. Fraiwan L, Lweesy K, Khasawneh N, Wenz H, Dickhaus H. Automated sleep stage identification system based on time–frequency analysis of a single EEG channel and random forest classifier. Computer methods and programs in biomedicine. 2012;108(1):10–19. pmid:22178068
View Article • PubMed/NCBI • Google Scholar
10. Hsu YL, Yang YT, Wang JS, Hsu CY. Automatic sleep stage recurrent neural classifier using energy features of EEG signals. Neurocomputing. 2013;104:105–114.
View Article • Google Scholar
11. Sutskever I, Vinyals O, Le QV. Sequence to sequence learning with neural networks. In: Advances in neural information processing systems; 2014. p. 3104–3112.
12. Mousavi SS, Schukat M, Howley E. Deep reinforcement learning: an overview. In: Proceedings of SAI Intelligent Systems Conference. Springer; 2016. p. 426–440.
13. Mousavi S, Schukat M, Howley E, Borji A, Mozayani N. Learning to predict where to look in interactive environments using deep recurrent q-learning. arXiv preprint arXiv:161205753. 2016;.
14. Mousavi SS, Schukat M, Howley E. Traffic light control using deep policy-gradient and value-function-based reinforcement learning. IET Intelligent Transport Systems. 2017;11(7):417–423.
View Article • Google Scholar
15. Goldberger AL, Amaral LA, Glass L, Hausdorff JM, Ivanov PC, Mark RG, et al. PhysioBank, PhysioToolkit, and PhysioNet: components of a new research resource for complex physiologic signals. Circulation. 2000;101(23):e215–e220. pmid:10851218
View Article • PubMed/NCBI • Google Scholar
16. Yildirim O, Baloglu UB, Acharya UR. A Deep Learning Model for Automated Sleep Stages Classification Using PSG Signals. International Journal of Environmental Research and Public Health. 2019;16(4):599.
View Article • Google Scholar
17. Supratak A, Dong H, Wu C, Guo Y. DeepSleepNet: a model for automatic sleep stage scoring based on raw single-channel EEG. IEEE Transactions on Neural Systems and Rehabilitation Engineering. 2017;25(11):1998–2008. pmid:28678710
View Article • PubMed/NCBI • Google Scholar
18. Tsinalis O, Matthews PM, Guo Y. Automatic sleep stage scoring using time-frequency analysis and stacked sparse autoencoders. Annals of biomedical engineering. 2016;44(5):1587–1597. pmid:26464268
View Article • PubMed/NCBI • Google Scholar
19. Michielli N, Acharya UR, Molinari F. Cascaded LSTM recurrent neural network for automated sleep stage classification using single-channel EEG signals. Computers in biology and medicine. 2019;. pmid:30685634
View Article • PubMed/NCBI • Google Scholar
20. Tuck. Stages of Sleep and Sleep Cycles; 2018. Available from: https://www.tuck.com/stages/.
21. Johnson M, Schuster M, Le QV, Krikun M, Wu Y, Chen Z, et al. Google’s multilingual neural machine translation system: enabling zero-shot translation. arXiv preprint arXiv:161104558. 2016;.
22. Cohen MX. Analyzing neural time series data: theory and practice. MIT press; 2014.
23. Fernandez R, Rendel A, Ramabhadran B, Hoory R. Using deep bidirectional recurrent neural networks for prosodic-target prediction in a unit-selection text-to-speech system. In: Sixteenth Annual Conference of the International Speech Communication Association; 2015.
24. Schuster M, Paliwal KK. Bidirectional recurrent neural networks. IEEE Transactions on Signal Processing. 1997;45(11):2673–2681.
View Article • Google Scholar

25. Wang S, Liu W, Wu J, Cao L, Meng Q, Kennedy PJ. Training deep neural networks on imbalanced data sets. In: Neural Networks (IJCNN), 2016 International Joint Conference on. IEEE; 2016. p. 4368–4374.

26. Kemp B, Zwinderman AH, Tuk B, Kamphuisen HA, Oberye JJ. Analysis of a sleep-dependent neuronal feedback loop: the slow-wave microcontinuity of the EEG. IEEE Transactions on Biomedical Engineering. 2000;47(9):1185–1194. pmid:11008419
View Article • PubMed/NCBI • Google Scholar

27. Chawla NV, Bowyer KW, Hall LO, Kegelmeyer WP. SMOTE: synthetic minority over-sampling technique. Journal of artificial intelligence research. 2002;16:321–357.
View Article • Google Scholar

28. Tsinalis O, Matthews PM, Guo Y, Zafeiriou S. Automatic sleep stage scoring with single-channel EEG using convolutional neural networks. arXiv preprint arXiv:161001683. 2016;.

29. Mousavi S, Afghah F. Inter-and intra-patient ECG heartbeat classification for arrhythmia detection: a sequence to sequence deep learning approach. ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE; 2019;pp. 1308–1312.

30. Mousavi S, Afghah F, Razi A, Acharya UR. ECGNET: Learning where to attend for detection of atrial fibrillation with deep visual attention. arXiv preprint arXiv:181207422. 2018, to appear in IEEE BHI 2019;.