Single channel EEG sleep stage scoring using a Recurrent Neural Network

Stephen Harvell[1], **Giulio Borghesi**[1], **Olutosin Sonuyi**[1]
[1]**Georgia Institute of Technology, Atlanta, GA, USA**

Abstract

Sleep plays a most important role in human health, yet we struggle to understand its underlying mechanics. Rapid advances in our comprehension of sleep patterns are hindered by the lack of tools that can identify the sleep stages experienced by an individual with no supervision. In this study, we propose a sleep annotation tool that is able to predict the sleep stages experienced by an individual during sleep from a single electroencephalogram (EEG) signal. Our model uses a recurrent neural network and was trained with EEG data from two datasets, a reduced one consisting of 18 patients, and a large one consisting of 5196 patients. The model was validated using data from 2 and 595 randomly selected patients, and the overall prediction accuracy was $\sim$70 % and $\sim$65 % respectively. Examination of the confusion matrix revealed a tendency of the proposed algorithm to misclassify the least common sleep stages. A discussion is provided on possible remedies that can be adopted to address this limitation. A video presentation of this work can be found at the following link: `https://youtu.be/0r_755jqAI8`.

Introduction

Sleep plays a central role in human health. Abnormal sleep patterns can have serious health consequences and have also been identified as valuable markers of vulnerability and risk in the early stages of Alzheimers disease and Parkinsons disease. Being able to monitor the quality of sleep of an individual is thus important not only to detect underlying medical conditions, but also from a medical research perspective.

Sleep quality is traditionally assessed by sleep experts through manual scoring of a polysomnogram (PSG), a set of electrical signals that include an electroencephalogram (EEG), an electrooculogram (EOG), an electromyogram (EMG) and an electrocardiogram (ECG). To score a PSG, the signals are partitioned into 30-s epochs which are manually classified into five sleep stages (REM, N1, N2, N3 and wake) following the staging criteria found in sleep manuals such as the one from the American Academy of Sleep Medicine[1]. This procedure is expensive as it requires at least one hour of a sleep experts time to score an eight-hours long PSG. Additionally, the sleep monitoring system needed to record the PSG is intrusive and not readily available outside of medical research facilities. There is thus significant interest in developing machine learning models for automatic sleep stage scoring based on a limited set of signals, such as a single-channel EEG, which can be collected easily using inexpensive wearable devices.

Previous studies on automatic sleep stage scoring have explored the use of deep learning methods to correctly classify each 30-s epoch of a PSG alongside more traditional machine learning models such as decision trees. Tsinalis et al.[2] developed a convolutional neural network (CNN) architecture in which the input data was a single-channel raw EEG signal. The model was trained and evaluated on a dataset of 20 healthy patients using 20-fold cross-validation and then compared against previous state-of-the-art results with hand-engineered features. The new model achieved performance comparable to that of the hand-engineered approach, with the advantage that no prior expert knowledge was needed to develop the learning algorithm. Supratak et al.[3] proposed a deep learning model called DeepSleepNet that combined two CNNs with different filter sizes to extract time-invariant features and bidirectional-Long Short-Term Memories (LSTM) to learn stage transition rules among sleep stages from sequences of EEG epochs. The model was trained and evaluated on two separate datasets of 31 and 20 patients, using 31-fold and 20-fold cross-validation respectively, and achieved performance similar to state-of-the-art models available in the open literature. Similarly to the work of Tsinalis et al.[2], DeepSleepNet used a single-channel EEG signal as input data. Biswal et al.[4] presented a sleep annotation tool called SLEEPNET in which sleep stage scoring was performed using a deep Recurrent Neural Network (RNN) with expert-defined features extracted from six-channel raw EEG signals. The model was trained and evaluated on a dataset of 10,000 PSGs (9,000 PSGs for training, 1,000 for testing) and achieved the best performance among a pool of traditional machine learning approaches (logistic regression, tree boosting, multi-layer perceptron) and deep learning ones (CNN, Recurrent-Convolutional Neural Network (RCNN)) that were trained on the same dataset. Finally, the model proposed by Lajnef et al.[5] provides an example of automatic sleep stage classification using a traditional machine learning model, e.g. a decision tree with multi-class Support Vector Machines (SVM). The proposed model was trained and tested on PSG data from 15 patients and performed better than standard multi-class procedures such as Linear Discriminant Analysis (LDA) and One-Against-All SVM. In this study, features extraction

Sleep stage	Training data	Percent of training	Test data	Percent of test
Wake	5765	35.5 %	966	48.3 %
N1	484	3.0 %	47	2.35 %
N2	5891	36.25 %	647	32.35 %
N3	2316	14.25 %	191	9.6 %
REM	1794	11.0 %	148	7.4 %

Table 1: Total number of 30-s epochs spent in each sleep stage by the patients in the training and test sets for the reduced dataset.

Sleep stage	Training data	Percent of training	Test data	Percent of test
Wake	1384470	28.79 %	155073	28.28 %
N1	179549	3.73 %	19532	3.56 %
N2	1965621	40.88 %	226399	41.30 %
N3	608505	12.65 %	70173	12.80 %
REM	670605	13.95 %	77073	14.06 %

Table 2: Total number of 30-s epochs spent in each sleep stage by the patients in the training and test sets for the full dataset.

from the available EEG signals was performed manually using expert knowledge.

In this study, we present an automatic sleep annotation tool that combine the use of expert-defined features with a recurrent neural network to predict sleep stages from an EEG reading. Our approach is similar to the one followed by Biswal et al.[4] in developing SLEEPNET: differently from that work, however, we decided to use only a single EEG channel as input data. Two separate datasets were compiled from the Sleep Heart Health Study (SHHS) database[6] to validate and train our model: a reduced one, which was used to verify the model correctness, and a full one, which was used to train our final model. The scope of this work is to develop a sleep annotation tool capable of achieving high predictive accuracy while using as less EEG signals as possible. This choice is motivated by the fact that multiple EEG signals are difficult to record in a unique sleeping session without the intervention of highly skilled technicians. The remainder of this paper consists of three parts: first, we describe our model and provide some background on the dataset used for training / testing purposes; then, we evaluate the model's performance on selected metrics and discuss the results; finally, we summarize our findings and provide directions for future work.

Methodology

This section consists of five parts. We first describe how the expert-defined features used in our model were constructed from the EEG signals. We then describe the architecture of the recurrent neural network, and provide some background on the dataset used to train and validate our model. Finally, we discuss our choice of performance metrics and provide some background on how the model was implemented.

Feature selection

Each 30-s epoch of EEG data was used to extract features in both the time and frequency domain. Two time domain features were chosen in this study: the line length and the kurtosis of the EEG signal. The line length is defined as the sum of the absolute difference between consecutive data in a 30-s epoch and is used to quantify the amplitude and frequency of the oscillations in the EEG signal. The kurtosis is chosen here because of its ability to reveal the presence of extreme values in the data.

To extract frequency domain features, we partitioned each 30-s epoch of the EEG signal into 29 sub-epochs. The duration of each sub-epoch was 2-s with a 1-s overlap with the previous sub-epoch. For each of these sub-epochs, we computed the power spectral density, which was used to find the spectral powers of the delta (0.5-4 Hz), theta (4-8 Hz), alpha (8-12 Hz) and sigma (12-20 Hz) frequency bands. To compute the spectral powers, the trapezoidal rule was used to integrate the power spectral density in the frequency bands of interest. The spectral powers were normalized by the total spectral power in the 0-20 Hz frequency band. For each 30-s epoch, we choose the 95% percentile, minimum, mean and standard deviation of the normalized delta, theta and alpha spectral powers of its 29 sub-sequences as the frequency-domain features. Additional frequency-domain features included the 95% percentile, minimum, mean and

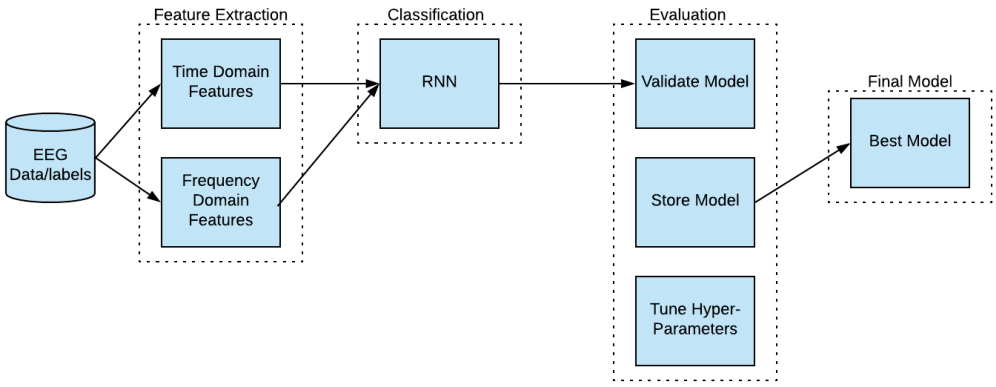

Figure 1: Sleep annotation tool pipeline.

standard deviation of the delta / theta, delta / alpha, theta / alpha spectral powers ratios, and the kurtosis of the power spectral density in the delta, theta, alpha and sigma frequency bands.

The total number of features for each 30-s epoch of the EEG signal was 30 (2 time domain features, plus 4 frequency domain features for each of the 6 spectral power quantities, plus 4 features for the kurtosis of the spectrogram in the four frequency bands considered). The features selected in this study are the same ones used by Biswal et al.[4] in the SLEEPNET sleep annotation tool, with the difference that only EEG data from a single channel is used here.

Recurrent neural network architecture

The expert-defined features introduced in the previous section could be used with no further modification to predict the sleep stage associated with a 30-s epoch of the EEG. Unfortunately, since these features do not contain any information about the previous epochs, they cannot identify temporal patterns; in particular, they are not able to learn the stage transition rules used by sleep experts to score a sleep stage based on what happened in the past. To address this shortcoming, we converted each patients EEG into a sequence of expert-defined features and then fed these sequences to a recurrent neural network (RNN) to learn temporal dependencies among features in different 30-s epochs. RNNs are a class of artificial neural networks that use an internal memory to identify temporal relationships between data.

The network utilized in this study consists of 5 layers of long short-term memory (LSTM) cells, each with 1000 hidden nodes. For the reduced datasets, a smaller network was used, consisting of 3 layers of LSTM cells, each with 250 hidden nodes. In both models, the output of the last LSTM cell is fed to a softmax layer to convert the network's output into sleep stages probabilities. The most probable sleep stage is then selected as the sleep stage associated with the input data. To make the learning process computationally tractable, we limited the length of the input sequences to 25: in other words, the input sequences were constructed by collecting expert features from 25 consecutive 30-s epochs of each patients EEG data. Dropout[8] was used for the cells in the LSTM layers to prevent overfitting of the neural network. This regularization technique consists in dropping the LSTM units alongside their connections with a fixed probability during training. The dropout probability controls the balance between the model's tendency to overfit the training data and the learning efficiency: in general, as its value is increased, overfitting is less likely, but learning becomes slow.

Dataset description

The EEG data used in this study was obtained from the Sleep Heart Health Study (SHHS)[6,7]. The SHHS database consists of ∼9,000 PSG readings obtained from more than 5,000 patients in two separate clinical visits, hereby referred to as baseline and follow-up. 5,804 patients participated in the baseline visit, during which a PSG was recorded. Of these patients, 4,080 returned for the follow-up visit five years later, during which a second PSG was taken. The PSGs were manually annotated by sleep experts using 6 sleep stages: Wake, N1, N2, N3, N4 and REM.

20 patients records were chosen at random from the SHHS database to perform an initial assessment of our automatic sleep annotation tool: in particular, 18 patient records were used to train the RNN, and 2 to validate it. In the following, we will refer to this dataset as the reduced patients records dataset. Once the correctness of our model was verified, we

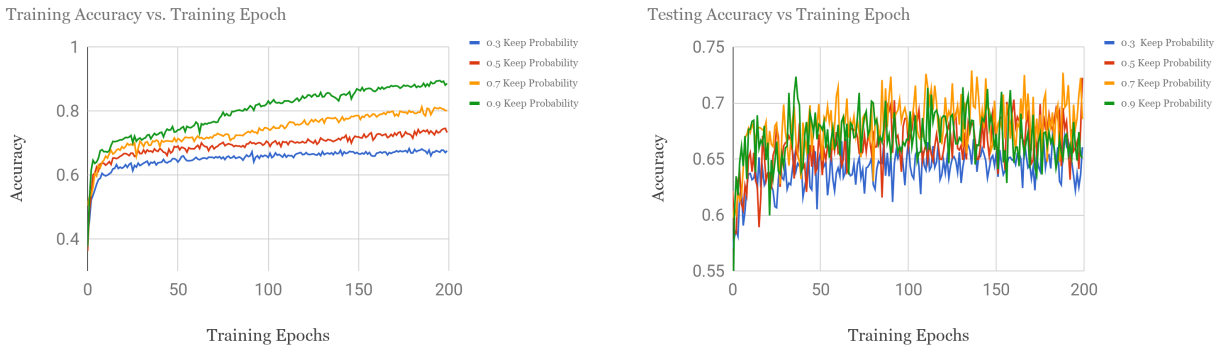

Figure 2: Evolution of the prediction accuracy during training on the training (left) and validation (right) datasets.

considered a much larger dataset of 5791 patients records to train and validate our model: specifically, 5196 patients records were used for training the RNN and the remaining 595 for validating it. Following the guidelines set forth by the American Academy of Sleep Medicine[1], we combined stage N4 with stage N3, and thus only considered 5 sleep stages. A breakdown of the number of 30-s epochs spent in each of these 5 sleep stages is provided in Table 1 for the patients in the initial assessment study, and in Table 2 for the patients in the study using the full dataset. The most common sleep stage for both datasets is N2, followed by Wake. These two stages alone represents approximately 70 % of all 30-s epochs in our datasets. The least common sleep stage is N1.

Performance metrics

Two metrics were used to assess the performance of our sleep annotation tool: the classification accuracy and the confusion matrix. The classification accuracy is defined as the number of correct predictions made divided by the total number of predictions made. This quantity provides the percentage of sleep stages correctly identified by the algorithm; unfortunately, it is not always indicative of its performance because of the imbalance between the numbers of the 30-s epochs for the various sleep stages in the SHHS database. The confusion matrix addresses this shortcoming by revealing the number of correct and incorrect predictions made by the algorithm for each sleep stage.

Algorithm implementation

We implemented our deep learning model using the deep learning library TensorFlow[9]. The EEG data for each patient was pre-processed using the Spark Python API (PySpark) to extract the time and frequency domain features for each 30-s epoch of the EEGs. The network was trained using an Adam optimizer with learning rate $\alpha = 5 \times 10^{-4}$ and mini-batch size of 10. To construct the mini-batches, we partitioned each patients EEG into 10 subsequences of approximately equal length and then selected 25 consecutive 30-s epochs from each of these subsequence to form the mini-batch. For each patient, we constructed as many mini-batches as possible before starting to use the next patients data. During each training epoch, the network was trained on all available data. The cross-entropy between the networks predictions and the actual sleep stages was used as the loss function.

While the RNN cannot be trained in parallel, it is still possible to simultaneously train multiple RNNs. Training different RNNs at the same time is of interest to identify the combination of model's hyper-parameters that achieves the best performance on the validation set. Although time and computing resources availability were not sufficient to perform a systematic investigation of the hyper-parameters space, we still develop the underlying software infrastructure to conduct such a study using PySpark.

A schematic representation of our algorithm pipeline is shown in Figure 1. As our model was trained, we also assessed its performance during each training epoch using patients data from the validation set. Once training was completed, the models weights were stored to disk, so to be readily available for future use. Training was considered complete when the accuracy on the validation set reached its maximum. To identify the best model, we investigated the impact of the dropout probability on both the learning efficiency and the prediction accuracy on the validation set. Because of time constraints, we were unable to assess the influence of the networks architecture on the prediction accuracy: we plan to conduct this study as part of future work.

		Predicted				
		Wake	**N1**	**N2**	**N3**	**REM**
Actual	**Wake**	**5242**	0	422	44	57
	N1	225	**0**	228	2	29
	N2	274	0	**5112**	357	148
	N3	53	0	409	**1848**	6
	REM	181	0	556	69	**988**

		Predicted				
		Wake	**N1**	**N2**	**N3**	**REM**
Actual	**Wake**	**944**	0	18	0	4
	N1	42	**0**	5	0	0
	N2	207	0	**411**	28	1
	N3	64	0	69	**58**	0
	REM	26	0	76	16	**30**

Table 3: Confusion matrices for training (left) and validation (right) reduced sets after 150 training epochs.

		Predicted				
		Wake	**N1**	**N2**	**N3**	**REM**
Actual	**Wake**	**5382**	19	286	38	40
	N1	167	**149**	142	0	26
	N2	159	19	**5379**	261	73
	N3	27	0	354	**1932**	3
	REM	73	2	283	37	**1399**

		Predicted				
		Wake	**N1**	**N2**	**N3**	**REM**
Actual	**Wake**	**909**	1	55	1	0
	N1	38	**1**	7	0	0
	N2	235	1	**383**	17	12
	N3	50	0	124	**17**	0
	REM	40	2	77	11	**18**

Table 4: Confusion matrices for training (left) and validation (right) reduced sets after 300 training epochs.

Results on reduced dataset

Figure 2 shows how the prediction accuracy of the model proposed in this study changes during training on the reduced dataset as a function of the dropout probability. It is clear that high values of the keep probability[1] leads to model overfitting, especially for long training periods; low values of keep probability, on the other hand, results in slow learning. In general, the prediction accuracy of the model on the training set always increases, whereas the one on the validation set reaches a maximum around 150 training epochs before starting to decrease as the network is trained further. The highest prediction accuracy on the validation set was obtained for a value of keep probability of 0.7 after approximately 150 training epochs and was in excess of 70 %. This value of the keep probability was subsequently adopted to analyze the per-class prediction accuracy of our model.

Confusion matrices are shown in Table 3 for the validation and training sets after approximately 150 training epochs. The same matrices after 300 training epochs are shown in Table 4. The overall prediction accuracy after 150 training epochs is 81.2 % and 72.15 % for the training and validation sets respectively. After 300 training epochs, the prediction accuracies change to 87.6 % for the training set and 66.4 % for the validation set. We observe that, after 150 training epochs, the model has become effective in correctly labeling the most common sleep stages, e.g. Wake and N2. N3 is predicted with good accuracy (79.8 %) on the training set, but not on the validation set, for which the accuracy is just 30.4 %. Neither the REM nor the N1 sleep stages can be correctly identified by our model at this point. These results may suggest that our network is not fully trained yet. This is not the case though: indeed, as the number of training epochs is increased, the model's accuracy on the validation set decreases, as shown by the corresponding confusion matrix after 300 training epochs. Although the model becomes able to predict with high precision the sleep stages for the training set, its performance on the validation set decreases slightly for the Wake and N2 sleep stages, and significantly for the less common N3 and REM stages.

Results on full dataset

The model proposed in this study was also trained for 20 epochs on the full dataset; unfortunately, we were unable to train the model further due to unexpected memory errors that we experienced while running our code. The confusion matrices obtained for the validation and training sets after 20 epochs are shown in Table 5. Similarly to what observed for the reduced dataset, the model appears to perform reasonably well for the Wake and N2 stages. The performance on the less represented stages (N1, N3 and REM) is not as good, and ranges from 12.27 % for N1 to 63.43 % for N3 on the training set (13.90 % for N1 to 71.90 % for REM on the validation set). In general, we observed an improvement in the model's overall accuracy on the training set as the number of training epochs was increased. The performance on the validation set, unfortunately, was relatively flat around 65 %. It is possible that, due to the larger size of the RNN used with the full dataset, a higher number of training epochs is needed before the model is able to learn patterns in

[1] the keep probability is one minus the dropout probability. High values of keep probability corresponds to low probability of dropping a LSTM unit alongside its connection.

		Predicted				
		Wake	**N1**	**N2**	**N3**	**REM**
Actual	**Wake**	**1155921**	10831	160864	15084	41770
	N1	57348	**22037**	71821	405	27938
	N2	136785	11520	**1550279**	147307	119730
	N3	15066	36	201620	**385973**	5810
	REM	61594	5455	222636	5627	**375293**

		Predicted				
		Wake	**N1**	**N2**	**N3**	**REM**
Actual	**Wake**	**133879**	575	12343	447	7829
	N1	9248	**2716**	3623	1	3944
	N2	21749	3004	**149878**	1891	49877
	N3	1493	19	46081	**14994**	7586
	REM	10369	980	10265	44	**55415**

Table 5: Confusion matrices for training (top) and validation (bottom) full sets after 20 training epochs.

the underlying data. A preliminary investigation conducted on the reduced dataset confirmed this intuition. As future work, we hope to be able to address the aforementioned memory errors and show that, once trained on a sufficiently large number of epochs, our full model is able to identify the various sleep stages in an EEG with good accuracy.

Discussion

The results presented in the previous section indicate that our model is able to identify with reasonable accuracy the most common sleep stages (Wake and N2), but fails at correctly labeling the least frequent ones (N1, N3 and REM). One reason behind this behavior may be related to the characteristics of the sleep data used in this study. As already mentioned in the previous sections, more than $70\,\%$ of the 30-s epochs used to train our model were classified as either Wake or N2. Given the smaller numbers of training examples associated with the N1, N3 and REM stages, it is possible that the data presented to our model was insufficient to learn how to correctly recognize these sleep stages. In previous studies on sleep classification using neural networks, oversampling was used to address this problem. Although oversampling is not easily applicable to train a RNN, for which temporal sequences of data must be provided as input, future studies should explore methods to apply such a strategy.

It is also possible that the set of expert-defined features selected for this study is not sufficient to unambiguously characterize the various sleep stages. The combination of expert-defined features with a recurrent neural network to identify the sleep stages from an EEG signal was already explored by Dong et al.[10]. The feature set used in that study appears significantly richer than the one used here and may have a strong impact on the model's accuracy. An alternative approach could be to follow Biswal et al.[4] and use multiple-channels EEG signals: indeed, it is possible that some of the expert-defined features used in this work can be readily identified from certain EEG channels but not from others.

Another possible approach to improve our results could be to automatically identify features from the EEG data using a convolutional neural network (CNN). Automatic features detection through a CNN has been already used with satisfying results by Supratak et al.[3] and Biswal et al.[4]. Although we extended our model to incorporate a CNN that processes the EEG raw data and returns a features vector for each 30-s epochs of the PSG, we could not test this model's extension due to time constraint and unavailability of computing resources.

Conclusion

We presented a tool for automatic sleep stage scoring of a EEG reading based on the combination of expert-defined features with a recurrent neural network. Our model is able to correctly identify the most common sleep stages (Wake and N2) in an EEG, but is not very accurate in predicting the less common stages, such as N3. We argue that these results are possibly due to the use of a set of expert-defined features that is not fully representative of the various sleep stages, and perhaps to limiting the number of EEG channels used to one. Future work should address the impact of the number of EEG signals used on the model's accuracy and also explore whether the use of different sets of

expert-defined features can lead to better results or not. Using a convolutional neural network to automatically extract features from the raw EEG data is also a viable strategy to avoid manual selection of the expert defined features used to train the RNN.

References

1. Iber C, Ancoli-Israel A, Chesson A, Quan SF. The AASM manual for the scoring of sleep and associated events: rules, terminology and technical specifications, Westchester, IL, American Association of Sleep Medicine, 2007
2. Tsinalis O, Matthews P, Guo Y, Zafeiriou S. Automatic sleep stage scoring with single-channel EEG using convolutional neural networks, arXiv preprint arXiv:1610.01683, 2017
3. Supratak A, Dong H, Wu C, Guo Y. DeepSleepNet: a model for automatic sleep stage scoring based on raw single-channel EEG. eprint arXiv:1703.04046. 2017
4. Biswal S, Kulas J, Sun H, Goparaju B, Westover MB, Bianchi M, Sun J. SLEEPNET: automated sleep staging system via deep learning. eprint arXiv:1707.08262 2017
5. Lajnef T, Chaibi S, Ruby P, Aguera PE, Eichenlaub JB, Samet M, Kachouri A, Jerbi K. Learning machines and sleeping brains: automatic sleep stage classification using decision-tree multi-class support vector machines. Journal of Neuroscience Methods, 250, pp. 94-105, 2015
6. Quan SF, Howard BV, Iber C, Kiley JP, Nieto FJ, O'Connor GT, Rapoport DM, Redline S, Robbins J, Samet JM, Wahl PW. The Sleep Heart Health Study: design, rationale, and methods. Sleep, 12, pp. 10771085. 1997
7. Dean DA., Goldberger AL, Mueller R, Kim M, Rueschman M, Mobley D, Sahoo SS, Jayapandian CP, Cui L, Morrical MG, Surovec S, Zhang GQ, Redline S. Scaling up scientific discovery in sleep medicine: The National Sleep Research Resource. Sleep, 5, pp. 11511164, 2016
8. Srivastava N, Hinton G, Krizhevsky A, Sutskever I, Salakhutdinov R. Dropout: a simple way to prevent neural networks from overfitting. Journal of Machine Learning Research, 15, pp. 1929-1958, 2014
9. Google Team. TensorFlow: large-scale machine learning on heterogeneous systems, 2015
10. Dong H, Supratak A, Pan W, Wu C, Matthews PM, Guo Y. Mixed neural network approach for temporal sleep stage classification. IEEE Transactions on Neural Systems and Rehabilitation Engineering, 26, pp. 324-333, 2018