



HAL
open science

Feature Transformation Based on Stacked Sparse Autoencoders for Sleep Stage Classification

Shirin Najdi, Ali Abdollahi Gharbali, José Manuel Fonseca

► **To cite this version:**

Shirin Najdi, Ali Abdollahi Gharbali, José Manuel Fonseca. Feature Transformation Based on Stacked Sparse Autoencoders for Sleep Stage Classification. 8th Doctoral Conference on Computing, Electrical and Industrial Systems (DoCEIS), May 2017, Costa de Caparica, Portugal. pp.191-200, 10.1007/978-3-319-56077-9_18 . hal-01629592

HAL Id: hal-01629592

<https://inria.hal.science/hal-01629592v1>

Submitted on 6 Nov 2017

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License

Feature Transformation Based on Stacked Sparse Autoencoders for Sleep Stage Classification

Shirin Najdi, Ali Abdollahi Gharbali, José Manuel Fonseca

CTS, Uninova, 2829-516 Caparica, Portugal
Faculdade de Ciências e Tecnologia, Universidade Nova de Lisboa Campus da Caparica,
Quinta da Torre, 2829-516 Monte de Caparica, Portugal
{s.najdi, a.gharbali}@campus.fct.unl.pt, jmrf@fct.unl.pt

Abstract. In this paper a deep learning based dimension reduction, feature transformation and classification method is proposed for automatic sleep stage classification. In order to enhance the feature vector, before feeding it to the deep network, a discriminative feature selection method is applied for removing the features with minimum information. Two-layer Stacked Sparse Autoencoder together with Softmax classifier is selected as the deep network model. The performance of the proposed method is compared with Softmax and k-nearest neighbour classifiers. Simulation results show that proposed deep learning structure outperformed others in terms of classification accuracy.

Keywords: Sleep Stage Classification, Deep Learning, Stacked Sparse Autoencoders, Feature Transformation.

1 Introduction

Sleep occupies significant part of human life. Diagnosis and treatment of sleep related disorders and sleep deficiency is of great importance in the sleep research community. Normal human sleep generally consists of two distinct stages with independent functions known as Non-Rapid Eye Movement (NREM) and Rapid Eye Movement (REM) sleep. In their ideal situation, NREM and REM states alternate regularly, each cycle lasting 90 minutes on average. NREM sleep accounts for 75-80% of sleep duration. According to the American Academy of Sleep Medicine (AASM) [1], NREM is subdivided into three stages: stage 1 or light sleep, stage 2 and stage 3 or Slow Wave Sleep (SWS). On the other hand, REM sleep accounts for 20-25% of sleep duration. The first REM state usually occurs 60-90 minutes after the onset of the NREM and lasts a few minutes.

A multiple-parameter test, called polysomnography (PSG), is normally used for analysis and interpretation of multiple, simultaneous physiologic events during sleep. PSG includes several body functions such as electroencephalogram (EEG), electro-oculogram (EOG), chin and leg electromyogram (EMG), airflow signals, respiratory effort signals, oxygen saturation and electrocardiogram (ECG). The data acquired in this way is analyzed by an expert in a clinic or hospital environment. The proportion

and distribution of sleep stages can then be used for diagnosis of sleep related problems. However, visual sleep stage classification is a subjective task and requires much time and effort. On the other hand, there is an increasing interest for at home sleep monitoring technologies that try to modernize sleep analysis and reduce the workload of healthcare centers. Therefore, there is a need for reliable automatic sleep stage classification systems that can efficiently perform sleep scoring.

One of the main challenges of automatic sleep stage classification is to compactly represent the subject's data in the form of a feature vector. This feature vector should be informative and non-redundant enough in order to decrease the computational complexity and facilitate the subsequent classification step. Proper selection of the classifier in order to achieve the highest possible classification accuracy is another challenge in these systems. Looking at the existing literature, It can be noted that a wide range of features including temporal [2–4], spectral [3,5,6], linear/non-linear [4,7,8] and statistical features [3,5] were extracted from different subsets of PSG recording and used for sleep scoring. Some of conventional feature transformation methods such as principal component analysis (PCA) [9] and kernel dimensionality reduction (KDR) [10] are used for reducing the dimensionality and enhancing the descriptive power of feature vector. Considering the fact that deep learning methods have found their way into many artificial intelligence applications with successful results reported from academia and industry, the main motivation for the current work was to explore the potential of deep learning for feature transformation and classification in the automatic sleep stage classification area. In view of this, the following research question emerges:

How should a deep learning technique be used for dimensionality reduction and classification of sleep stages, so that the computational complexity of classification step is reduced and the scoring accuracy is improved?

In this work, we propose an algorithm that answers this question. This paper is organized as follows: Section 2 explains how the proposed algorithm contributes to the smart systems. In Section 3, a review of the application of deep learning techniques in sleep stage classification is presented. Section 4, provides detailed description of the proposed algorithm. Simulation results and discussion are presented in Section 5. Finally, conclusion and future work are given in Section 6.

2 Relationship to Smart Systems

Smart systems refer to diverse range of technological systems that can perform autonomously or in collaboration with other systems. These systems have an ability to combine functionalities including sensing, actuating and controlling a particular situation. Based on the information they acquire, they have the ability to perform smart actions such as prediction or deciding and communicate with the user through highly sophisticated user interfaces. Generally, smart systems are made of several components, each having a special purpose. These components include sensors for data acquisition, information transmitting elements, command-and-control units for making decisions, components for transmitting decisions, taken, and finally actuators for triggering required action. These systems have been used to address some

problems in diverse areas, such as energy, transportation, security, safety, healthcare, etc.[11].

In healthcare area, the main goal of utilizing smart systems is to improve patient management workflow. This improvement leads to the reduction in the burden of medical staff, consultation time, waiting lists and medical costs. Smart healthcare systems, as advanced technology, are classified into three main categories: remote health monitoring systems (RHMS), mobile health monitoring systems (MHMS) and wireless health monitoring systems (WHMS). Among these three categories, WHMS refer to biosensors that can be worn by the subject and may also include one or both of RHMS and MHMS [12]. Wearable sleep monitoring systems are perfect examples of WHMS. The idea behind them is to use efficient, affordable and medically reliable systems for unsupervised at-home monitoring of patients' sleep data. Typically, wearable sleep monitoring systems have an algorithm that performs automatic sleep stage classification for evaluating sleep quality [13]. This work is developed in such a context to improve the performance of the classification accuracy using deep learning techniques.

3 State of The Art

Unlike some of the machine learning areas such as natural language processing and object classification, the potential of deep learning techniques is not fully explored in automatic sleep stage classification. This fact is also noticeable when it comes to the feature transformation for sleep scoring. To the best of our knowledge, there are few research works in this area and in the following we will try to briefly review them.

In [14], the main idea is using hybrid deep learning models to increase the performance of sleep stage classification. Deep Belief Networks (DBNs) are applied on 28 hand-crafted feature set for unsupervised generation of higher level features. For classification another deep structure, namely, Long Short Term Memory (LSTM) is used. In this work, sleep stage classification is regarded as a time series and sequence classification problem, therefore the ability of LSTM models for recognizing the patterns from a sequence of events is mentioned as the reason for using this classifier. The proposed algorithm is tested on two datasets of sleep recording and the features are extracted from EEG, EOG and EMG. The performance of the proposed algorithm (DBN +LSTM) is compared to three other sleep stage classification algorithms, namely DBN only, LSTM only and DBN with Hidden Markov Model (HMM). Simulation results show that two hybrid methods DBN+LSTM and DBN+HMM have significantly better performance than single DBN and single LSTM, while DBN+LSTM performs better than DBN+HMM for both datasets. It has been concluded that LSTM boosted the performance of DBN much better than HMM.

Tsinalis et al. [15] proposed an algorithm for sleep stage classification using time-frequency analysis based features and Stacked Sparse Autoencoders (SSAE). For each epoch, in total 557 features are extracted from the time-frequency representation of the single channel EEG signal. Complex Morlet wavelet is used for creating time-frequency representation. In this paper, it has been showed that the

classification accuracy can be improved by including the features from neighboring epochs. According to the simulation results, the proposed method leads to 1-2% scoring improvement compared to four other sleep scoring algorithms, that don't use deep learning. They succeeded to reduce the gap between the mean performance between S1 (as the most misclassified stage) and all other stages. Also, the adverse effect of inherent class imbalance in sleep data on the classification accuracy is highlighted. Authors tried to alleviate this effect by creating a balanced dataset in which all stages are equally represented.

In [16], the main focus is on generating meaningful data representations from unlabeled data. For this purpose, the performance of a two-layer DBN with 200 hidden units was compared with another feature transformation algorithm consisting of conventional methods (PCA + Sequential Backward Selection (SBR) + Gaussian Mixture Model (GMM)). These feature selection methods are applied on a vector of 28 hand-crafted features. Newly generated feature sets are classified by using Softmax classifier and Hidden Markov Model (HMM) respectively. Experimental results showed that DBN-based feature transformation performs much better the other method for sleep stage classification

Dong et al. in [17], proposed a practical approach was proposed for mitigating the limitations of single-channel automatic sleep stage classification using Mixed Neural Network (MNN). MNN is a deep learning-based feature transformation and classification technique and is composed of a Rectifier Neural Network (RNN), a Long Short-Term Memory (LSTM) and a Softmax regression. The input to this system is a feature vector with time-frequency domain, statistical and time domain features. Considering temporal dependency of sleep stages to each other, in addition to the features of the current epoch, the features from previous EEG epochs are also fed to the system. In this paper, several alternative electrode placements are explored and finally a convenient single forehead EEG channel together with an EOG channel configuration is proposed for the low-cost, at-home sleep monitoring applications.

4 Materials and Methods

Figure 1 shows an overview of proposed sleep stage classification algorithm with proposed feature transformation scheme.

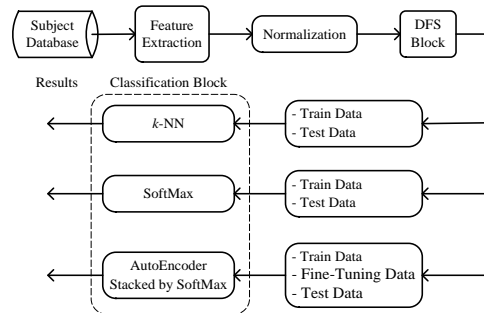


Fig. 1. Flowchart of the proposed algorithm.

4.1 Data

In this paper, we used a publically available dataset, called ISRUC-Sleep [18]. The data were acquired from 10 healthy adults, including 9 male and 1 female subjects aged between 30 and 58. The recordings were done in Sleep Medicine Centre of the Hospital of Coimbra University (CHUC) during an all-night session (eight hours). The PSG recordings of each subject were scored by two experts according to the AASM manual [1]. All EEG, EOG and chin EMG recordings were performed with a sampling rate of 200 Hz. The quality of the PSG recordings of this dataset have improved through a pre-processing step. In this pre-processing step, 1) a notch filter is applied to eliminate 50 Hz electrical noise, 2) EEG and EOG recordings are filtered using a bandpass Butterworth filter with a lower cut-off frequency of 0.3 Hz and higher cut-off frequency of 35 Hz, and 3) EMG channels are filtered using a bandpass Butterworth filter with a lower cut-off frequency of 10 Hz and higher cut-off frequency of 70 Hz. For the evaluation of our proposed method, we used C3-A2 EEG channel, right EOG and chin EMG channels. In this paper, we used all the data from 10 healthy subjects from ISRUC-Sleep dataset. The number of epochs for these 10 subjects is 954, 941, 824, 794, 944, 853, 814, 1000, 969, and 796. Totally we have 8889 epochs from this database and to avoid overfitting we used all of them.

4.2 Feature Extraction

All signals used in this study, were divided into 30-second epochs. A set of features were extracted from each epoch of EEG, EOG and EMG recordings of each subject. This feature set includes 49 features that can be considered as time, frequency, joint time-frequency domain, entropy-based and nonlinear types. Table 1 shows a summary of these features. For more detail see [19].

Table 1. Summary of the features extracted from PSG recordings.

Signal	Category	Feature Name
EEG	Time Domain (F1 to F12)	Statistical Features (Minimum Value, Maximum Value, Arithmetic Mean, Standard Deviation, Variance, Skewness, Kurtosis, Median), Zero-crossing Rate, Hjorth Parameters (Activity, Mobility and Complexity) [20].
	Time-Frequency Domain (F13 to F26)	Features Extracted from Wavelet Packet Coefficients including Energy of α , δ , $\beta 1$, $\beta 2$, θ and Spindle bands, Total Energy of all bands, Energy ratio of $(\frac{\alpha}{\delta+\theta}, \frac{\delta}{\alpha+\theta}, \frac{\theta}{\alpha+\delta}, \frac{\delta}{\theta}, \frac{\alpha}{\theta})$, Statistical Features (mean and standard deviation of coefficients in all of the bands).

Signal	Category	Feature Name
	Entropy (F27 to F30)	Spectral Entropy, Rényi Entropy, Approximate Entropy, Permutation Entropy [20].
	Non-linear (F31 to F36)	Petrosian Fractal Dimension, Mean Teager Energy, Mean Energy, Mean Curve Length, Hurst Exponent [20], ISD.
EOG	Time Domain (F37 to F41)	Mean, Maximum, Standard Deviation, Skewness, Kurtosis [21].
	Non-linear (F42)	Energy [21].
EMG	Frequency Domain (F43 to F46)	Total Power in the EMG Frequency Spectrum, Statistical Features of EMG Frequency Spectrum (Maximum, Mean, Standard Deviation) [21].
	Non-linear (F47 to F49)	Energy, Ratio of the EMG Signal Energy for the Current Epoch and Previous Epoch, Ratio of the EMG Signal Energy for the Current Epoch and Next Epoch [21].

4.3 Normalization

In order to standardize the range of Features, scaling normalization method was applied. Each feature (x_{ij}) is independently normalized by applying the following equation:

$$x'_{ij} = \frac{x_{ij} - \min(\mathbf{x}_i)}{\max(\mathbf{x}_i) - \min(\mathbf{x}_i)} \quad (1)$$

where \mathbf{x}_i is the vector of i^{th} feature.

4.3 Discriminative Feature Selection

There are many potential advantages in removing the features before final modelling and classification. Fewer features mean lower computational complexity. Also, some features may reduce the performance by their corrupt distributions. Consider a feature that a single value for all of the samples. According to [22], this feature is called “zero-variance predictor”. Even if it has little effect on the next step, this feature should be discarded from feature set, because it has no information. Similarly, some features may have few unique values that occur with low frequency. These features are called “near-zero variance predictors”. Kuhn et al. [22] defines two criteria for detecting near-zero variance features as follows:

- a) The ratio of unique values to the number of samples is low, for example 10%.
- b) The ratio of the frequency of the most dominant value to the frequency of the second dominant value is high, for example 20.

Using these two criteria, we applied Discriminative Feature Selection (DFS) to remove the features that didn't have enough discriminative power. As a result, the dimension of feature set reduced to 37.

4.4 Stacked Sparse Autoencoder (SSAE)

An autoencoder is a special type of neural network whose output values are equal to the inputs. An autoencoder typically consists of an encoder and a decoder and it is trained in an unsupervised manner using backpropagation. During training, a cost function that measures the error between input and output of the autoencoder is optimized, in other words, the autoencoder tries to learn the identity function. By applying special constraints on the network such as the number of hidden units, an autoencoder can learn new representation or coding of the data [23]. Suppose the input vector to the autoencoder is a set of un-labelled data $\mathbf{x} \in \mathbb{R}^{D_x}$. This vector is encoded to another vector $\mathbf{z} \in \mathbb{R}^{D_h}$ in the hidden layer as follows:

$$\mathbf{z} = h^1(\mathbf{W}^1\mathbf{x} + \mathbf{b}^1) \quad (2)$$

where h^1 is the transfer function of the encoder, \mathbf{W}^1 is the weight matrix and \mathbf{b}^1 is the bias vector of the encoder. Then, the autoencoder tries to decode this new representation back to the original input vector as follows:

$$\hat{\mathbf{x}} = h^2\mathbf{z} = h^2(\mathbf{W}^2\mathbf{z} + \mathbf{b}^2) \quad (3)$$

where h^2 is the transfer function of decoder, \mathbf{W}^2 is weight matrix and \mathbf{b}^2 is bias vector of the decoder. Sparse autoencoder is a specific type of autoencoder in which in order to encourage the sparsity of the output of the hidden layer, a constraint is imposed on the number of active hidden neurons. The cost function of sparse autoencoder is slightly different from the original autoencoder as follows:

$$E = \underbrace{\frac{1}{N} \sum \sum (\mathbf{x} - \hat{\mathbf{x}})^2}_{\text{mean squared error}} + \underbrace{\lambda \Omega_{\text{weights}}}_{\text{weight regularization}} + \underbrace{\beta \Omega_{\text{sparsity}}}_{\text{sparsity regularization}} \quad (4)$$

where N is length of the input vector, λ is the weight regularization parameter β is the sparsity regularization parameter [24].

A Stacked Sparse Autoencoder (SSAE) is a neural network with several sparse autoencoders. In this architecture, the output of each autoencoder is fully connected to the inputs of the next autoencoder. Greedy layer-wise training strategy is usually used for training SSAE. After the training of each layer is complete, a fine tuning is usually performed for enhancing the learned weights using backpropagation algorithm. Fine tuning can greatly improve the performance of the stacked autoencoder [23].

4.5 Softmax Classifier

After SSAE, Softmax classifier is stacked to the network as the output layer. Softmax classifier has a probabilistic interpretation of each. It is the generalization of binary Logistic Regression classifier to multiple classes. In sleep stage classification, the number of output classes is equal to the number of sleep stages.

5 Experimental Results and Discussion

For evaluating the performance of the proposed sleep stage classification algorithm, we used all data from 10 healthy subjects as described in section 4.1. The EEG, EOG and EMG signals of each subject were divided into 30-second epochs. After feature extraction and normalization, the feature sets were fed to DFS block to eliminate the near-zero variance features.

According to the criteria mentioned in section 4.3, 12 features were recognized as near-zero variance feature and removed from our sleep data model, as follows: maximum value (F1), minimum value (F2), variation (F5), median (F8), Petrosian fractal dimension (F31), permutation entropy (F30), Hjorth parameter (Activity) (F10), zero crossing number (F9), total power in the EMG frequency spectrum (F43), mean of power in the EMG frequency spectrum (F45), absolute energy of the time domain EMG signal (F47), maximum value of time domain EOG signal (F38).

After the feature vector was set, data were divided into two parts, training, testing, using 10-fold cross validation method. For fine tuning step of SSAE, part of training data was utilized. Our deep learning consists of three layers, a two-layer SSAE and a Softmax layer. The number of hidden units for the first and second layer of SSAE was 20 and 12, respectively.

For finding the best hyper-parameters for the autoencoders, we tried several models by adjusting sparsity regularization parameter, weight regularization parameter and the number of iterations. We used autoencoders with logistic sigmoid activation function for both of the layers. The performance of the proposed algorithm was compared with two other classifiers, Softmax and k -Nearest Neighbor (k -NN) classifier. The number of neighbors was set to 18 and Euclidean distance was used as a measure of distance for k -NN. To evaluate our systems' performance we used classification accuracy as the evaluation criterion.

Table 2 shows the individual sleep stage and overall classification accuracy extracted from confusion matrix for three different classifiers. The boldface numbers indicates the best performance.

Table 2. Results of the statistical analysis for comparison of each stage and overall accuracy.

Classifiers	Wake (%)	REM (%)	S1 (%)	S2 (%)	SWS (%)	Overall Accuracy (%)
Softmax	80	61.66	65	90	78.33	74.9
k -NN	85	66.66	61.66	70	83.33	73.33
SSAE	91	77	69	87	87	82.2

It is noticeable that SSAE method outperforms the other two classifiers in terms of overall accuracy. Also, for the individual sleep stages, in most of the cases SSAE discriminates the stages better. In addition to the higher performance, SSAE provides a considerable reduction in the dimension of the feature vector.

Considering that the second layer of SSAE had 12 hidden units, SSAE succeeded to decrease the dimension from 37 to 12, which means 67% reduction. Therefore, SSAE is a powerful tool to generate more descriptive features from original feature vector. In order to confirm the advantage of DFS block, the performance of SSAE-based sleep stage classification with and without this step was investigated. Without using DFS block, 49 original features were fed to SSAE.

The classification accuracy achieved in this way was 74.1% which is almost 8% less than accuracy with DFS block.

6 Conclusions and Future Works

Although feature transformation based on deep learning has already used in several machine learning applications, the advantages and potentials of applying these methods in sleep stage classification problem have not explored yet. This paper is a contribution in this regard. We proposed a method for dimension reduction and feature transformation based on SSAEs. The results show that SSAE can be considered as an appropriate tool for decreasing the complexity of sleep scoring issues. Future works will include comparing the performance of other conventional classifiers such as SVM and RF with SSAE in sleep stage classification.

Acknowledgment. This work was partially funded by FCT Strategic Program UID/EEA/00066/203 of UNINOVA, CTS.

References

1. Berry RB, Brooks R, Gamaldo CE, Harding SM, Lloyd RM, Marcus CL, et al. AASM - Manual for the Scoring of Sleep and Associated Events version 2.1. 2014.
2. Hamida ST-B, Ahmed B. Computer Based Sleep Staging: Challenges for the Future. 2013 7th IEEE GCC Conf. Exhib. IEEE; 2013. p. 280–5.
3. Koley B, Dey D. An ensemble system for automatic sleep stage classification using single channel EEG signal. *Comput. Biol. Med.* Elsevier; 2012;42:1186–95.
4. Şen B, Peker M, Çavuşoğlu A, Çelebi F V. A Comparative Study on Classification of Sleep Stage Based on EEG Signals Using Feature Selection and Classification Algorithms. *J. Med. Syst.* 2014;38:18.
5. Radha M, Garcia-Molina G, Poel M, Tononi G. Comparison of feature and classifier algorithms for online automatic sleep staging based on a single EEG signal. 36th Annu. Int. Conf. IEEE Eng. Med. Biol. Soc. IEEE; 2014. p. 1876–80.
6. Šušmáková K, Krakovská A. Discrimination ability of individual measures used in sleep stages classification. *Artif. Intell. Med.* 2008;44:261–77.
7. Hsu Y-L, Yang Y-T, Wang J-S, Hsu C-Y. Automatic sleep stage recurrent neural classifier using energy features of EEG signals. *Neurocomputing.* Elsevier; 2013;104:105–14.

8. Dursun M, Gunes S, Ozsen S, Yosunkaya S. Comparison of Artificial Immune Clustering with Fuzzy C-means Clustering in the sleep stage classification problem. 2012 Int. Symp. Innov. Intell. Syst. Appl. IEEE; 2012. p. 1–4.
9. Rempe M, Clegern W, Wisor J. An automated sleep-state classification algorithm for quantifying sleep timing and sleep-dependent dynamics of electroencephalographic and cerebral metabolic parameters. *Nat. Sci. Sleep*. 2015;7:85.
10. Fukumizu K, Bach FR, Jordan MI. Kernel Dimensionality Reduction for Supervised Learning. *J. Mach. Learn. Res.* 2004;5:73–99.
11. CORDIS Archive : European Commission : CORDIS : FP7 : ICT : Micro / nanosystems : Calls [Internet]. [cited 2016 Dec 17]. Available from: http://cordis.europa.eu/fp7/ict/micro-nanosystems/calls_en.html
12. Baig MM, Gholamhosseini H. Smart Health Monitoring Systems: An Overview of Design and Modeling. *J. Med. Syst.* Springer US; 2013;37:9898.
13. Tsinalis O, Matthews PM, Guo Y. Automatic Sleep Stage Scoring Using Time-Frequency Analysis and Stacked Sparse Autoencoders. *Ann. Biomed. Eng.* 2016;44:1587–97.
14. Giri EP, Fanany MI, Arymurthy AM. Combining Generative and Discriminative Neural Networks for Sleep Stages Classification. 2016;
15. Tsinalis O, Matthews PM, Guo Y. Automatic Sleep Stage Scoring Using Time-Frequency Analysis and Stacked Sparse Autoencoders. *Ann. Biomed. Eng.* Springer US; 2016;44:1587–97.
16. Långkvist M, Karlsson L, Loutfi A, Loutfi A. Sleep Stage Classification Using Unsupervised Feature Learning. *Adv. Artif. Neural Syst.* Hindawi Publishing Corporation; 2012;2012:1–9.
17. Dong H, Supratak A, Pan W, Wu C, Matthews PM, Guo Y. Mixed Neural Network Approach for Temporal Sleep Stage Classification. 2016;
18. Khalighi S, Sousa T, Santos JM, Nunes U. ISRUC-Sleep: A comprehensive public dataset for sleep researchers. *Comput. Methods Programs Biomed.* Elsevier Ireland Ltd; 2016;124:180–92.
19. Najdi S, Gharbali AA, Fonseca JM. A Comparison of Feature Ranking and Rank Aggregation Techniques in Automatic Sleep Stage Classification Based on Polysomnographic Signals. 4th Int. Conf. IWBBIO. Granada: Springer International Publishing; 2016. p. 230–41.
20. Şen B, Peker M, Çavuşoğlu A, Çelebi F V. A Comparative Study on Classification of Sleep Stage Based on EEG Signals Using Feature Selection and Classification Algorithms. *J. Med. Syst.* 2014;38:18.
21. Dursun M, Gunes S, Ozsen S, Yosunkaya S. Comparison of Artificial Immune Clustering with Fuzzy C-means Clustering in the sleep stage classification problem. 2012 Int. Symp. Innov. Intell. Syst. Appl. 2012;1–4.
22. Kuhn M, Johnson K. *Applied Predictive Modeling*. New York, NY: Springer New York; 2013.
23. Ng A, Ngiam J, Foo C, Mai Y, Suen C. UFLDL Tutorial. http://ufldl.stanford.edu/wiki/index.php/UFLDL_Tutorial. 2010;
24. Olshausen BA, Field DJ. Sparse coding with an overcomplete basis set: A strategy employed by V1? *Vision Res.* 1997;37:3311–25.