Classification of Sleep Stages Using Density Transformations and Convolutional Neural Networks

github repository: https://github.gatech.edu/ralbright7/CSE6250project

Presentation video: https://youtu.be/tRNz9cRWXtg

Richard L Albright(ralbright7@gatech.edu), Karthick Govindaraju (karthick1288@gatech.edu), Jay Sumners(jsumners3@gatech.edu) , Onkar Mishra(omishra3@gatech.edu)

Abstract

We have implemented a 6-layer convolutional neural network (CNNs) with a 1-dimensional input, three convolution/pooling combinations, and three fully-connected layers for automatic sleep stage scoring based on EEG channels Fpz-Cz and Pz-Oz. The EEG signals were split into 30-second epochs and decomposed using one or a combination of power spectral density transformations along overlapping frequency bandwidths. The CNN was trained on the spectral density as the features of each epoch.. During initial training, we were able to achieve an overall validation accuracy of 89%; however the mean accuracy across different sleep stages were relatively lower at 66%, resulting in a test accuracy of 77%. Consistent with the literature, the data exhibited an imbalanced class structure favoring "wake" and differentiating "wake" from "light sleep" was difficult. Additional models were also tested.

Introduction

Sleep is one of the most important aspects of human wellbeing. Healthy sleep is a prerequisite for a healthy individual. Sleep disorders have been linked to 7 of the 15 leading causes of death in the U.S., including cardiovascular disease, malignant neoplasm, cerebrovascular disease, accidents, diabetes, septicemia, and hypertension [7]. Detecting and understanding sleep disorders is an important step in improving overall public health and sleep stage classification is a common tool for evaluating sleep patterns. Multiple studies have shown that sleep stages can be useful indicators of sleep disorders and accurately classifying the sleep stages plays a vital role in the diagnosis and treatment of these sleep disorders. Historically these sleep stage scorings have been performed by trained experts according to Rechtschaffen and Kales sleep staging criteria. Unsurprisingly, it would take experts several hours to annotate the sleep stages accurately, making the manual process of sleep staging very expensive. This leads to the need for automated sleep staging algorithms that have comparable accuracy to the manual scoring.

Problem Formulation

There have been several attempts to automate the sleep stage scoring process using deep learning algorithms, but they are mostly sequence-to-sequence models that attempt to score epochs in real-time. Our project's aim was to build a post-hoc model rather than a sequence-to-sequence model that can assist human sleep-stage assignment after data collection and classify epochs in isolation. We planned to achieve this by transforming EEG data in a way that is best suited for convolutional neural networks (CNN) and train the CNN to predict the sleep stages accurately.

This is in contrast to much of the literature where the focus has been on network architecture and hyperparameter tuning rather than feature construction.

Approach and Implementation

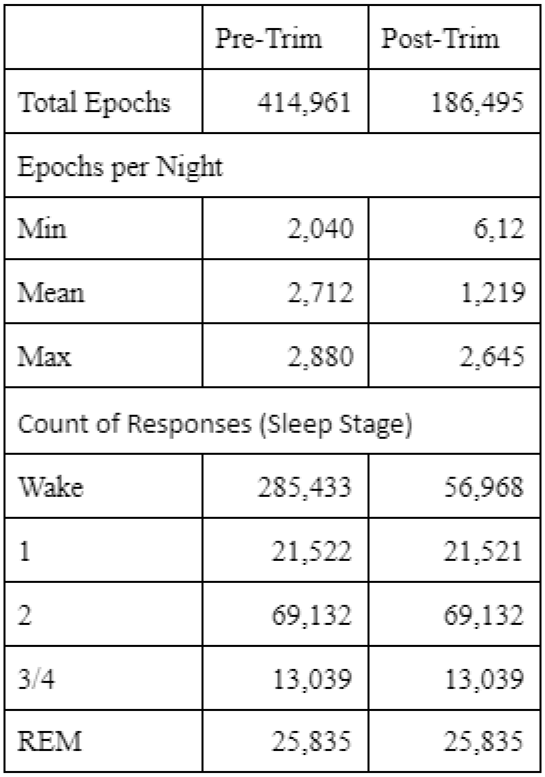
Data

The *sleep-cassette* portion of the Sleep-EDF Database[10] was used for model development. Obtained from a 1987-1991 study, the data includes EEG signals in Fpz-Cz and Pz-Oz channels from 78 patients over two nights each. The population included only Caucasian patients not taking sleep-related medications at the time of the study.

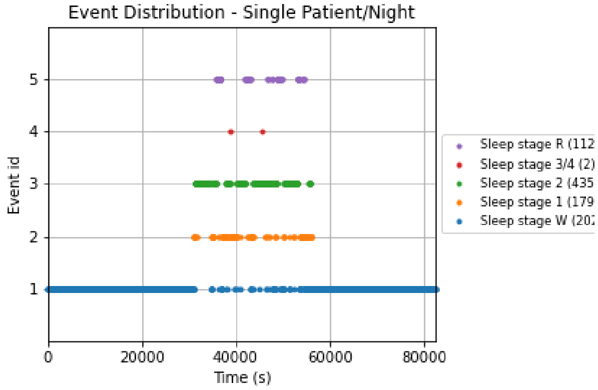
Total Patients	Age Range Patient Counts				Sex	
	25-39	40-59	60-79	80-101	Male	Female
78	20	21	21	16	37	41

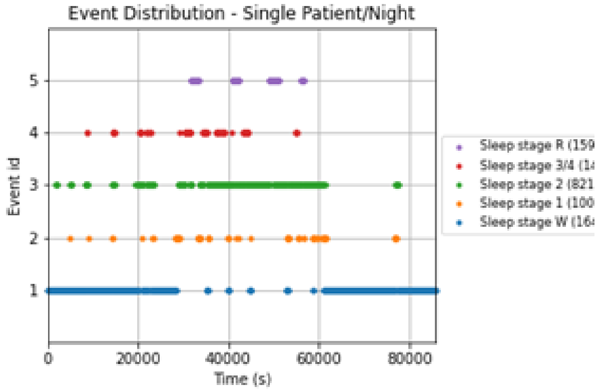
The data consisted of approximately 20 hours (split over the two nights) of EEG data and sleep-stage annotations at 30-second epochs for each patient recorded in the patient's home via cassette tape. There were a few known issues, including night 1 for two patients and night 2 for one patient. Bad epochs were dropped, although there were few. While the data was captured up to 100Hz, EEG data above 50 Hz was inconsistent across patients.

Due to the imbalance in the sleep stages favoring the wake stage, the team decided to trim the sleep data by cutting all the epochs with the label "wake" 15 minutes before the onset of sleep and 15 minute after the last sleep stage in the time series.

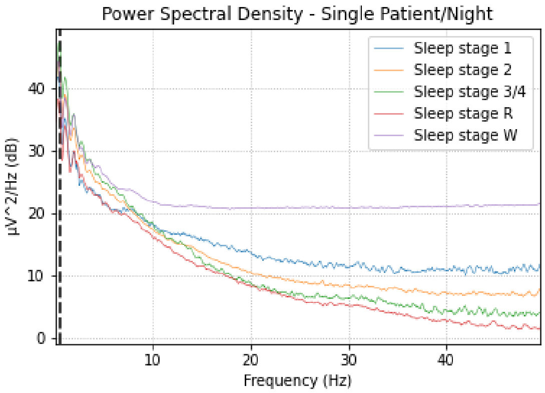


	Pre-Trim	Post-Trim
Total Epochs	414,961	186,495
Epochs per Night		
Min	2,040	6,12
Mean	2,712	1,219
Max	2,880	2,645
Count of Responses (Sleep Stage)		
Wake	285,433	56,968
1	21,522	21,521
2	69,132	69,132
3/4	13,039	13,039
REM	25,835	25,835





Data Pre-Processing



Power spectral density transformations were selected as a way to reduce the noise and convolution of the raw EEG signal (although some studies used the raw signal[2]). Using *pyspark* to load and transform the EEG signals in a distributed manner significantly reduced the processing time. Additionally, the *mne* package in python provided tools for loading and processing data from EDF files.

Welch transformed EEG signals clipped to 0.5-49.5 Hz (the most consistent range between patients) at 0.5 or 1.0 Hz intervals were used in initial model development. The Welch transformation was chosen since it allows, natively, mean or median averaging over the bandwidth. The expectation was that since EEG signals lead into the next epoch, that the median may provide a more predictive central tendency--being less susceptible to skewness. Additional transformations (e.g. multitaper, morlet) and their combinations were tested, including 2D spectrographs. A summarization of the ETL process follows:

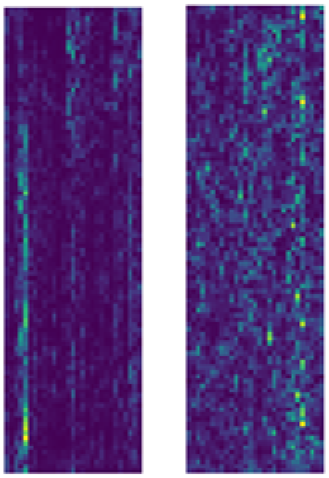
1. Group PSG and Hypnogram paths by patient and night.
2. Using *pyspark* to work distributed and the toolset from *mne*, for each pair of PSG and Hypnogram files:
   a. Load raw signal from PSG and annotation from Hypnogram.
   b. Apply annotations to raw and set channels.
   c. Extract events from annotated raw signals.
   d. Trim the beginning and end wake periods to within 15 minutes of sleep for each session.
   e. Applied a finite impulse response bandpass filter with cutoffs of 0.5Hz and 49.5Hz.
   f. Set max time (*tmax*) as 30.0 -1 / sfreq of the raw object (sets length of epoch).
   g. Convert raw to epochs using events, event ids, tmax, etc.
   h. Drop bad epochs.
   i. Apply indicated power spectral transformation, including aggregation (mean or median).
   j. Separate PSDs at each bandwidth and remove nans.

Data Pre-Processing: 1D Spectral Densities

Six different 1-dimensional transformations were conducted on the Epoch data including four different power spectral density transformations using two methods (welch and multitaper). Frequency bands were divided into 49 bands between 0.5Hz and 49.5Hz at 0.5Hz increments. The final result was an array of 196 values for each epoch.

We also performed 2 time frequency transformations using Morlet Wavelets. Using the same frequency bands, the time axis for each epoch was broken into 0.25 second increments. The mean or the median of the frequencies axis were taken for each band. The mean or median was then taken again on those aggregate band values for each 0.25 second increment. The final result was an array of 120 values for each epoch. All transformations were performed on each 30 second epoch with no overlap.

Finally, a combination of morlet wavelets and welch/multitaper transformations were tested by appending the columns. The goal having been to provide additional separation between the densities of each stage.



Data Pre-Processing: 2D Spectrograph

An additional welch transformation of the data was performed between 0.5 Hz and 49.5 Hz with 2 second wide transforms and 1 second overlaps within each 30 second epoch. The mean of the resultant stack of 2 second transforms was then normalized and converted into a rasterized 2D image (a heatmap) using matplotlib's *viridis* colormap. The RGB conversion of these images served as inputs to the 2D CNN.

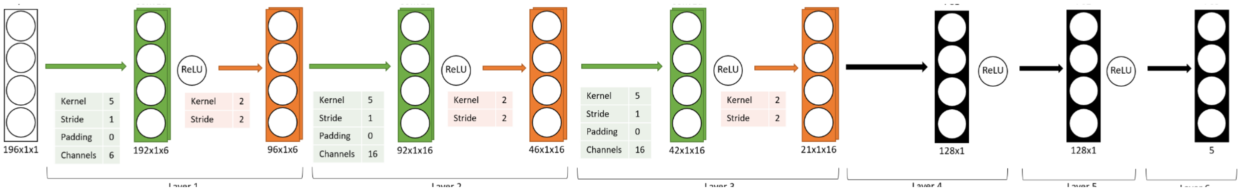
Convolutional Neural Network

A CNN typically consists of successive convolutional layers and pooling layers followed by some number of fully connected layers. The optimization algorithm that we are using to train the network is the popular Stochastic Gradient Descent (SGD). We used ReLU activation function between the convolutional layers due to the fact that we have 3 pairs of convolutional and pooling layers, three fully-connected layers where initial two fully connected layers were followed by a dropout layer. ReLU was used as an activation function to provide non linearity in CNN and reduce the computational intensity without compromising the performance of the CNN. The last pooling layer is followed by 3 fully connected layers to slowly reduce the number of features.

Two Networks

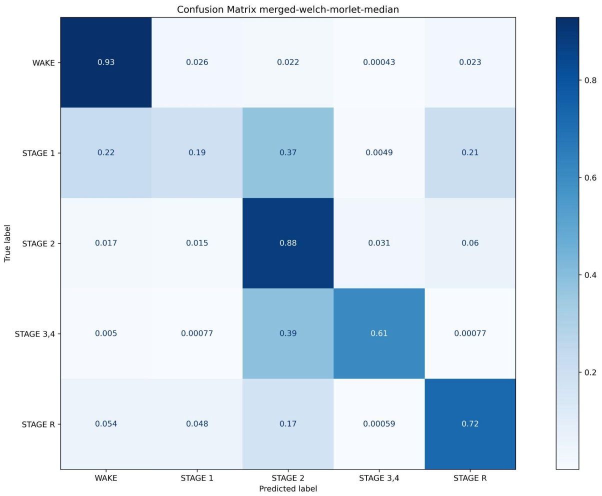
Using this architecture, we built two CNN models, 1D CNN and 2D CNN, to train the 1D transformed epoch data and the spectrogram images respectively.

The input data consists of a (# epochs, # frequency bins) and passed to the model in mini-batches of 32 epochs. Since we are utilizing SGD as an optimization algorithm, relatively small mini-batches may permit the model to converge faster (although SGD never truly converges). The input dataset is split into 60% training, 20% validation, and 20% testing. On larger datasets and under further development, training may take up a larger percentage of the overall data.



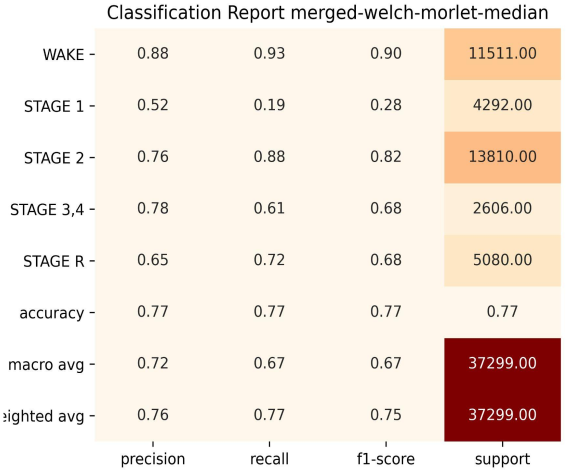
Experimental Evaluation

The 1D CNN was trained with all the 4 one dimensional transformed sleep data. The 1D CNN model performed well in classifying all the sleep stages except the stage 1 which was expected because according to available research on the subject stage 1 is the hardest stage to classify and even manual classification of stage 1 is inconsistent between different expert scorers. The metrics of the best performing 1D CNN model is attached in Fig3a and Fig 3b. Accuracy of validation data was used as performance metrics to choose hyperparameters for training. When we increased n_epoch to 100, we found that the model was underfitting and we found moderate differences in the performance of the model between the training set and the validation set. Hence we added a dropout layer which improved the performance of the model on the validation set. We started with a dropout rate of 0.5 but later reduced the drop out rate to 0.2 which gave better performance. We could find that with addition of the dropout layer, there was improvement in validation performance and thus the model was able to generalize better with validation data.



Confusion Matrix merged-welch-morlet-median

True label \ Predicted label	WAKE	STAGE 1	STAGE 2	STAGE 3,4	STAGE R
WAKE	0.93	0.026	0.022	0.00043	0.023
STAGE 1	0.22	0.19	0.37	0.0049	0.21
STAGE 2	0.017	0.015	0.88	0.031	0.06
STAGE 3,4	0.005	0.00077	0.39	0.61	0.00077
STAGE R	0.054	0.048	0.17	0.00059	0.72

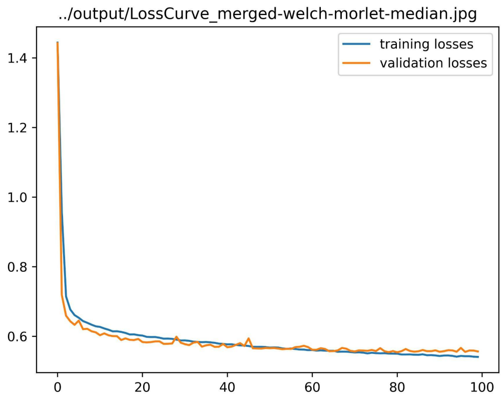


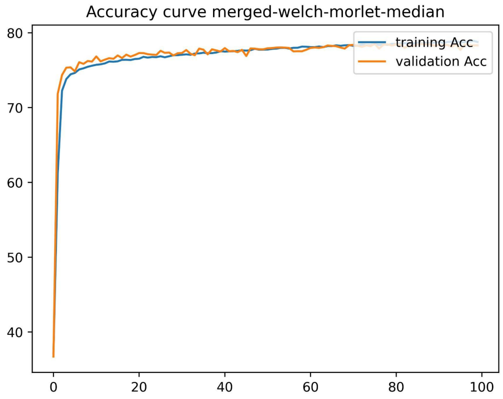
Classification Report merged-welch-morlet-median

	precision	recall	f1-score	support
WAKE	0.88	0.93	0.90	11511.00
STAGE 1	0.52	0.19	0.28	4292.00
STAGE 2	0.76	0.88	0.82	13810.00
STAGE 3,4	0.78	0.61	0.68	2606.00
STAGE R	0.65	0.72	0.68	5080.00
accuracy	0.77	0.77	0.77	0.77
macro avg	0.72	0.67	0.67	37299.00
ɘighted avg	0.76	0.77	0.75	37299.00

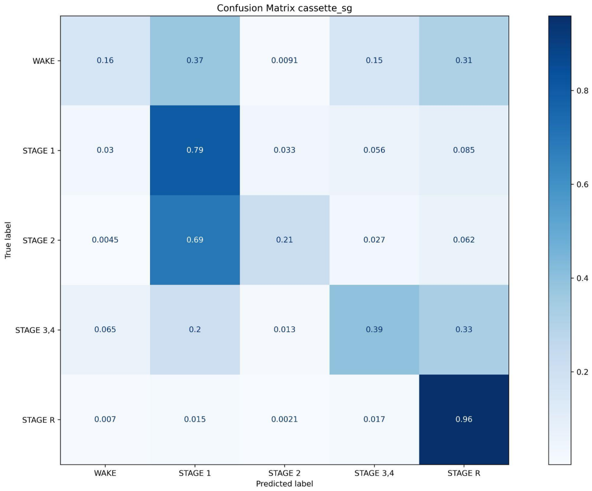
The overall accuracy of the model was 77% +/- 0.4% at 95% confidence interval for the 1D CNN model. As we can see from the confusion matrix above, the most accurately classified sleep stage was wake stage with 93% of wake epochs correctly classified, followed by 88% correctly classified for Stage 2. Similarly, Stage 1 is the most misclassified sleep stage.

We used number of epoch as a hyperparameter. We started training our model n_epoch = 20 and kept it increasing till n_epoch = 100. Initially, with n_epoch = 20, the validation loss curve was still improving and there were chances of improving the model by increasing the number of epochs. Thus, we increased the number of epochs to 50 and then 100. The model trained well up to 100 epochs beyond which the accuracy and loss stopped improving and we decided to stop training models beyond 100 epochs.

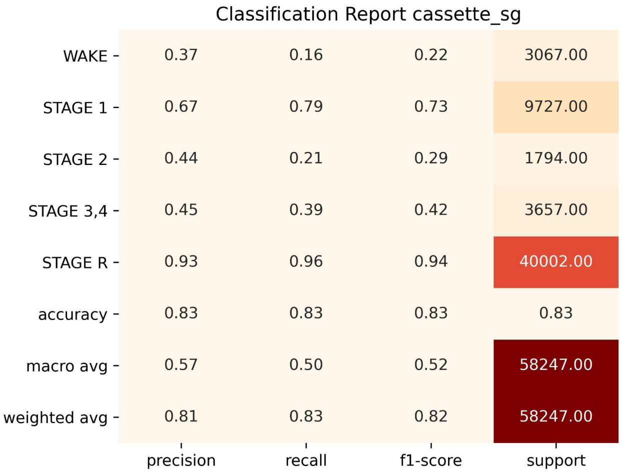




In an attempt to improve on the accuracy of the stage 1 class/label, the team decided to build a 2D CNN model and train it using the spectrogram images of the epochs. As it can be seen in the above confusion matrix and the classification report (below), even though the model seems to classify stage 1 well, it has high bias toward stage 1 and it could not be used to do an ensemble along with 1D CNN to improve the classification accuracy for all stages.



True label \ Predicted label	WAKE	STAGE 1	STAGE 2	STAGE 3,4	STAGE R
WAKE	0.16	0.37	0.0091	0.15	0.31
STAGE 1	0.03	0.79	0.033	0.056	0.085
STAGE 2	0.0045	0.69	0.21	0.027	0.062
STAGE 3,4	0.065	0.2	0.013	0.39	0.33
STAGE R	0.007	0.015	0.0021	0.017	0.96


Classification Report cassette_sg

	precision	recall	f1-score	support
WAKE	0.37	0.16	0.22	3067.00
STAGE 1	0.67	0.79	0.73	9727.00
STAGE 2	0.44	0.21	0.29	1794.00
STAGE 3,4	0.45	0.39	0.42	3657.00
STAGE R	0.93	0.96	0.94	40002.00
accuracy	0.83	0.83	0.83	0.83
macro avg	0.57	0.50	0.52	58247.00
weighted avg	0.81	0.83	0.82	58247.00

Discussion

Most of the transformations performed well, nearly equally so. Welch transformations and welch-morlet combinations had the best accuracy. Given nearly matched performance between mean and median aggregation, we settle on using median to safeguard against outliers. Unfortunately, the 1D models had difficulty separating “light sleep” from other types of sleep -- a consistent problem in the literature. The spectrograph performed well, but exhibited significant bias towards “light sleep”.

There were roadblocks throughout the project that future projects are also likely to encounter. First, since the data is recorded nearly constantly, preprocessing can be intensive. We overcame this limitation by distributing the load and transformation stages and by using transformations that would limit the dimensionality of the data.

Second, class imbalance is common whereby “wake” is significantly overrepresented in the data. We found that trimming the start and end epochs worked well, but the timeframe chosen (15 minutes) was arbitrary. Different datasets may or may not have this imbalance or may need a different trimming length.

Finally, since sleep changes slowly from epoch-to-epoch, there may be only slight differences in the ending epoch of one stage and the starting epoch of another stage. Overall, it is more likely that a REM epoch will lead to another REM epoch. This makes predicting epochs in isolation difficult.

Conclusion

Our study showed that density transformations can provide significant differentiating information about sleep stages at the epoch level. The exact density transformation may have little impact on their own, but that spectrographic representation of the transformation provides additional differentiating information. Additionally, the simple combination of transformation can further differentiate stages; although the effect is small.

While not the primary objective, we showed that it is possible to differentiate “light sleep” from other stages using spectrographs. We believe an ensemble model may balance out the biases of each model and provide better accuracy. Other color schemes for the spectrographs may also provide better differentiable sleep stages in the RGB spectrum.

In conclusion, we showed that a relatively shallow 1D CNN can provide reasonably good performance in sleep stage scoring when evaluating epochs in isolation. We also showed that spectrographic representations of the same data can provide differentiation between “light sleep” where the 1D CNN model (and the state-of-the-art) struggles.

References

1. O. Tsinalis, P. M. Matthews, Y. Guo, and S. Zafeiriou. Automatic sleep stage scoring with single-channel eeg using convolutional neural networks. arXiv preprint arXiv:1610.01683, 2016.
2. S. Biswal, J. Kulas, H. Sun, B. Goparaju, M. Brandon Westover, M. T. Bianchi, and J. Sun. SLEEPNET: Automated sleep staging system via deep learning. 26 July 2017
3. M. Zhao, S. Yue, D. Katabi, T. S. Jaakkola, and M. T. Bianchi. Learning sleep stages from radio signals: A conditional adversarial architecture. In International Conference on Machine Learning, pages 4100–4109, 2017
4. R. U. D. K. Linda Zhang, Daniel Fabbri. Automated sleep stage scoring of the sleep heart health study using deep neural networks. 2019.
5. I. Al-Hussaini, C. Xiao, M. B. Westover, and J. Sun. Sleeper: interpretable sleep staging via prototypes from expert rules. In Machine Learning for Healthcare Conference, pages 721–739, 2019.
6. O. Tsinalis, P. Matthews, and Y. Guo. Automatic Sleep Stage Scoring Using Time-Frequency Analysis and Stacked Sparse Autoencoders: Annals of Biomedical Engineering, Vol. 44, No. 5, May 2016 pp. 1587–1597
7. Chattu VK, Manzar MD, Kumary S, Burman D, Spence DW, Pandi-Perumal SR. The Global Problem of Insufficient Sleep and Its Serious Public Health Implications. *Healthcare (Basel)*. 2018;7(1):1. Published 2018 Dec 20. doi:10.3390/healthcare7010001
8. Zhang GQ, Cui L, Mueller R, Tao S, Kim M, Rueschman M, Mariani S, Mobley D, Redline S. The National Sleep Research Resource: towards a sleep data commons. J Am Med Inform Assoc. 2018 Oct 1;25(10):1351-1358. doi: 10.1093/jamia/ocy064. PMID: 29860441; PMCID: PMC6188513.
9. 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. 1997 Dec;20(12):1077-85. PMID: 9493915.
10. Goldberger, A., L. Amaral, L. Glass, J. Hausdorff, P. C. Ivanov, R. Mark, J. E. Mietus, G. B. Moody, C. K. Peng, and H. E. Stanley. "PhysioBank, PhysioToolkit, and PhysioNet: Components of a new research resource for complex physiologic signals. Circulation [https://physionet.org/content/sleep-edfx/1.0.0]. 101 (23), pp. e215–e220." (2000).