University of Colorado Denver

Project Report
Sleep Stage Classification

Reported by:

Erfan Jafari Khadem Zavareh

Supriya Ramachandra

Swayanshu Shanti Pragnya

University of Colorado Denver

Problem Statement and Background

In this project, we have implemented different machine learning and deep learning algorithms to automatically classify sleep stages i.e, to Wake, N1, N2, N3, and REM on windows of 30 seconds of raw data and compared the results. The dataset source and its characteristics are described below:

Source: https://www.physionet.org/content/sleep-edfx/1.0.0/

The source database contains 197 whole-night PolySomnoGraphic sleep recordings, containing EEG, EOG, chin EMG, and event markers. Some records also contain respiration and body temperature. The ***PSG.edf** files are whole-night polysomnographic sleep recordings containing EEG (from Fpz-Cz and Pz-Oz electrode locations), EOG (horizontal), submental chin EMG, and an event marker. The ***Hypnogram.edf** files contain annotations of the sleep patterns that correspond to the PSGs. These patterns (hypnograms) consist of sleep stages W, R, 1, 2, 3, 4, M (Movement time), and ? (not scored). All hypnograms were manually scored by well-trained technicians.

According to the AASM manual (https://aasm.org/clinical-resources/scoring-manual/), sleep EEG consists of 5 stages. Each of the five stages is defined below.
-W: Awake state (stage W) is characterized by alpha or faster frequency bands occupying more than 50% of the epoch, frequent eye movements, and high EMG tone.
-N1: Stage N1 is scored when alpha occupies more than 50% of epoch while theta activity, slow rolling eye movements, and vertex waves are evident.
-N2: Stage N2 is scored when sleep spindles or K-complexes (less than 3 min apart) are noted.
-N3: Stage N3 is characterized by delta activity detected in over 20% of the epoch length.
-REM: Upon sleep scoring an epoch is marked as REM when saw-tooth waves along with rapid eye movements as well as lowest EMG signals are observed through each epoch.

Bands	Frequencies (Hz)	Amplitude (μV)
Delta (δ)	0–4	20–100
Theta (θ)	4–8	10
Alpha (α)	8–13	2–100
Beta (β)	13–22	5–10
Gamma (γ)	> 30	-

The informal success measures that we planned to use are the Confusion Matrix to calculate accuracy, classification report, log loss.

Why is sleep stage classification important and its impact?

Sleep is the primary function of the brain and plays an essential role in an individual's performance, learning ability, and physical movement. Effective diagnosis and treatment of patients with sleep-related complaints is currently an urgent and heavily researched topic in the healthcare community. Sleep stage classification is one of the most critical steps in effective analysis of sleep patterns to identify sleep-related conditions that include fatigue, drowsiness, or sleep disorders, such as apnea, insomnia, or narcolepsy. Classic approaches involve trained human sleep scorers, utilizing a manual scoring technique, according to certain standards. Visual

inspection undertaken by sleep experts is a time-consuming and burdensome task. Over the years, researchers have been trying to find more efficient, reliable, and accurate methods for the classification process. Computer-assisted sleep stage classification systems are nowadays considered essential for both sleep-related disorders diagnosis and sleep monitoring. Over the years, the models that have been developed have been successfully applied to the sleep scoring problem. Even with that success, the research is ongoing aiming at improving these automated classification schemes, as the results are not yet satisfactory to be used as a standard procedure in clinical studies.

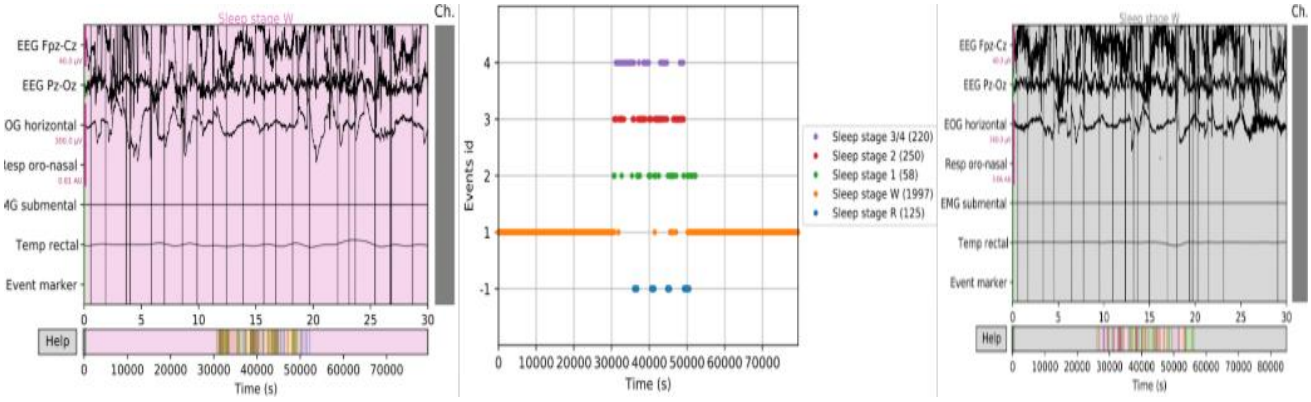
Methods

Learning Process:

Sleep Data in terms of EEG → Data Preprocessing based on 30 sec Time frame→ Feature Extraction → 5 stage classification

Some of the classification methods that we explored are Random forest, SVM, LTSM (Class of Artificial Neural network). Below are the process and methods that we used:

1) **Data Preprocessing** – Both EDF and EDF+ files were read using MNE library. The fetched data was stored in the NumPy array format (representing the EEG signals and its labels) and saved into .NPZ files. In this step a band-pass filter (Butterworth of order eight) with pass-band bandwidth of 0.5–40 Hz was applied to enhance EEG quality signal. Noise in the EEG signal caused due to movement was also removed by using a filter of this range, as this noise usually occurs at a much higher frequency.
   Before Data Preprocessing it was vital to know all the channels and Classification for individual person's data. Below is the EEG plot of the data using MNE library



2) **Feature Extraction** – We have done some feature extractions as follows:
   - PSD or *Power Spectral Density* was used to extract the features from filtered data. PSD is the measure of the signal's power content over frequency. A PSD is typically used to characterize broadband random signals. The amplitude of the PSD is normalized by the spectral resolution employed to digitize the signal.
   - PFD or *Petrosian Fractal Dimension*, which can be calculated using the following:

$$PFD = \frac{\log_{10} N}{\log_{10} N + \log_{10}(N/(N + 0.4N_\delta)}$$

Where N is the length of time series, and N_δ is the number of changes in the sign of the signal's derivative.

- Hjorth Parameters, which includes Activity, Mobility, and Complexity. These are the normalized slope descriptions in EEGs.
- Hurst Exponent, which also called rescaled range statistics, is a measure of long-term memory of the times series. It is a scalar value between 0 and 1.
- DFA or *Detrended Fluctuation Analysis*, is used to determine the self-affinity of the time series. It is a scalar value between 0 and 1.

3) **LSTM** (Long short-term memory): is an artificial recurrent neural network (RNN) architecture used in the field of deep learning. Unlike standard feedforward neural networks, LSTM has feedback connections. It can not only process single data points but also entire sequences of data and using this method accuracy of 78% on 50subjects and 60% for 151 subjects was achieved. As the data used in the project was EEG wave data we tried to explore an algorithm other than Machine learning algorithms to classify sleep stages in a test dataset.
4) **Random Forest** – It is an ensemble tree-based learning algorithm. The Random Forest Classifier is a set of decision trees from a randomly selected subset of the training set. It aggregates the votes from different decision trees to decide the final class of the test object. We implemented a random forest because it achieved the highest accuracy compared to the other machine learning algorithms in both our project and the research paper we referred to. We used 80% of the data to train the model and classify sleep stages on the 20% test data.
5) **Support vector machine (SVM)** – The objective of the support vector machine algorithm is to find a hyperplane in N-dimensional space (N — the number of features) that distinctly classify the data points. Similar to random forest test train split of data, we used multi-class SVM to classify sleep stages.
6) **KNeighborsClassifier, Kmean**s, etc. were the other models that we explored in our project to study their behavior while classifying sleep stages

Some of the parameters that we used in our project are:

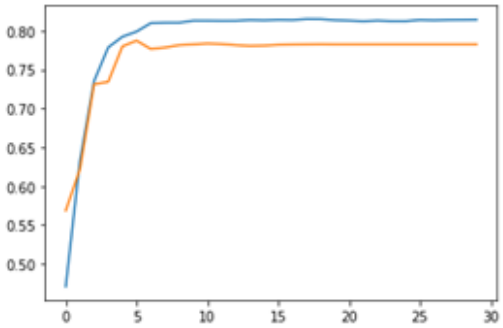
- Select_ch: to select the channels from the PSG files, only the Fpz-Cz EEG channel was used in our project. Using two channels requires a different approach.
- Epochs – 30s epochs were used in the project following the standard of AASM and R&K manuals.
- In LSTM the number of epochs or the iteration of the model run on the data had a significant impact on the performance. For example, we ran 25 epochs on all the 151 subjects the accuracy of 61.60% was achieved and by running 30 epochs on 50 subjects an accuracy of 78.6% was achieved.
- In SVM the default kernel was 'linear' which had a poor impact on the classification and accuracy of only 20% was achieved but by setting the kernel to 'rbf' a significant performance improvement was achieved with an accuracy of 60.33%

Details on the methods that we used and did not work well for the physionet dataset:

In the research paper that we referred to, an accuracy of 90% was achieved using SVM, RF, and other machine learning algorithms. But in our case machine learning algorithms did not achieve the expected accuracy, we used confusion matrix and accuracy to interpret the performance. As per our understanding, some of the reasons could be the type of dataset used was different from the research paper we referred to. We could have achieved higher accuracy by implementing better feature extraction.

Results

1) LSTM: Accuracy and classification report was used to evaluate the performance of the model. The below accuracy plot shows the model executed on 50 subjects for 30 epochs. As seen in the plot there was a significant increase in the performance or accuracy of the model. For 151 subjects if we increase the number of epochs we could have achieved higher accuracy, but as the data size for each subject is large enough RAM (Random access memory) did not support the execution on all the subjects for more than 25 epochs which achieved an accuracy of 61%




```
>>> f1 score: 0.6706400589751825
              precision    recall  f1-score   support

           0       0.91      0.87      0.89      1257
           1       0.49      0.04      0.08       436
           2       0.91      0.80      0.85      2352
           3       0.81      0.92      0.86       581
           4       0.54      0.90      0.68       974

    accuracy                           0.79      5600
   macro avg       0.73      0.71      0.67      5600
weighted avg       0.80      0.79      0.77      5600
```

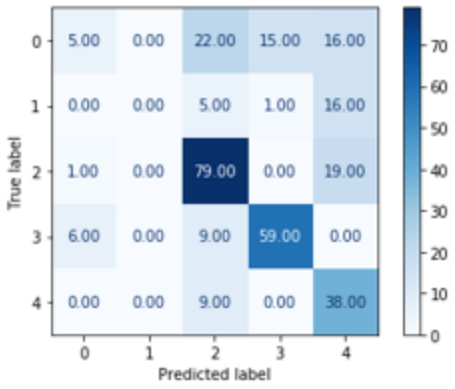
2) SVM: Performance of the SVM model increased significantly by using 'rbf' (Radial basis function) over 'linear' kernel. Accuracy measure, classification report, and confusion matrix were used to evaluate the model performance. An accuracy of 60% was achieved by running the model on 50 subjects.

With SVM accuracy is: 0.6033333333333334

```
Classification report:
              precision    recall  f1-score   support

           0       0.09      0.42      0.14        12
           1       0.00      0.00      0.00         0
           2       0.80      0.64      0.71       124
           3       0.80      0.79      0.79        75
           4       0.81      0.43      0.56        89

    accuracy                           0.60       300
   macro avg       0.50      0.45      0.44       300
weighted avg       0.77      0.60      0.66       300
```

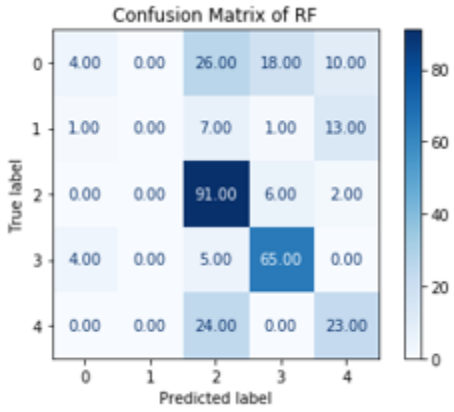


3) RF: Metrics evaluation and subjects used in this model were similar to SVM. An accuracy of 59% was achieved.

With Random forest, accuracy is: 0.59

Classification report:

	precision	recall	f1-score	support
0	0.07	0.44	0.12	9
1	0.00	0.00	0.00	0
2	0.92	0.59	0.72	153
3	0.88	0.72	0.79	90
4	0.49	0.48	0.48	48
accuracy			0.61	300
macro avg	0.47	0.45	0.42	300
weighted avg	0.81	0.61	0.69	300



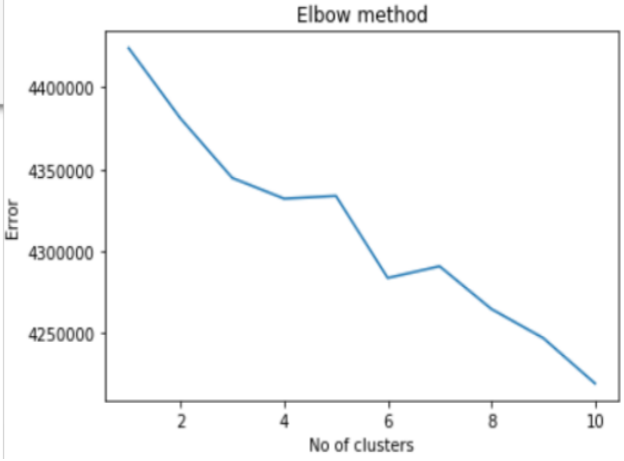
4) K_Means

```
kmeans5 = KMeans(n_clusters=5)
y_kmeans5 = kmeans5.fit_predict(xsample)
print(y_kmeans5)

kmeans5.cluster_centers_
```

```
2 2 2 2 3 3 3 2 2 3 2 2 3 2 2 1 2 1 2 2 2 1 2 2 2 2 2 2 2 2 2 3 2 2 2 2
2 1 3 2 3 2 2 2 2 1 2 2 2 2 2 1 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
2 2 2 2 2 2 2 2 2 2 2 2 2 3 2 2 2 1 2 2 3 2 2 2 3 2 2 3 2 1 2 2 2 2 2 2 2
2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 3 2 2 2 3 2
2 2 1 1 2 2 2 1 2 2 3 2 2 2 2 2 2 2 3 2 2 2 1 2 3 2 2 2 2 2 2 3 2 2 2 2 2 2
3 2 2 2 2 1 3 0 2 2 2 2 2 2 3 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2
2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 3 2 2 2 2 2 2 2 2
2 2 2 2 2 2 3 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 3 2 2 2 1 2 3 1 2 1 2
1 3 3 2 1 2 2 2 2 3 3 2 0 2 1 2 3 1 1 1 3 1 2 2 3 0 2 2 3 2 2 3 2 2 2 1 1
2 0 2 1 2 3 2 3 3 2 1 3 2 2 2 2 2 2 3 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 3 2 2 2 2 2 2 2]

array([[-0.22650129, -0.39525416, -0.49374154, ...,  1.3791858 ,
         1.6338761 ,  1.5813066 ],
       [-0.01234133, -0.02733597, -0.05687766, ..., -0.1411224 ,
        -0.2153845 , -0.24735919],
```

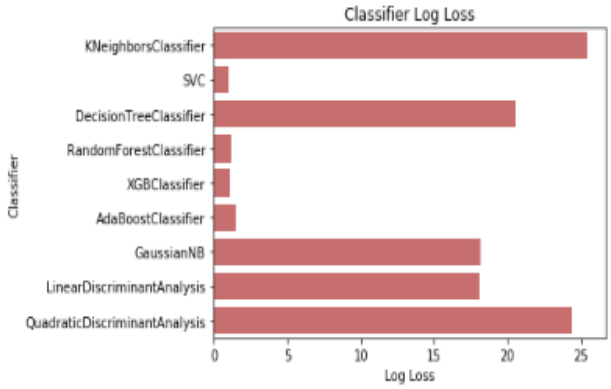


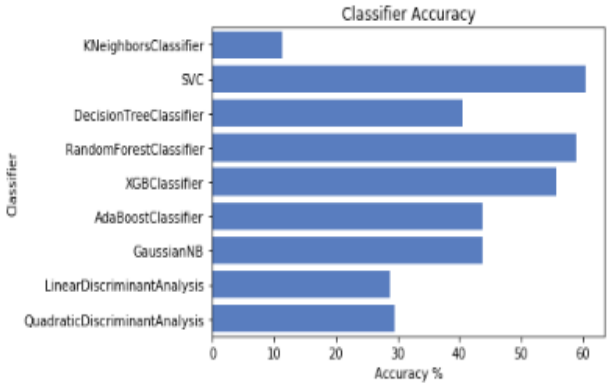
Due to inconsistent nonlinear data, the error has been increased with respect to cluster numbers.

Our baseline/Naïve solution was implementing one of the common and simplest classification algorithms called KNN (K-nearest neighbor) with which we achieved an accuracy of less than 20%. Since the accuracy achieved was very low we implemented SVM as this model achieved better accuracy even in the paper we referred to.

The **reason** for the low accuracy of KNN could be the nonlinear nature of the data and some time frames being zero, and also the hyperparameters of KNN is not suitable for exceptions like having 0 or noise. And complete noise elimination as it can be used for other inference like snoring problems. We then implemented SVM using linear where the performance was similar to KNN, however by using the ‘rbf’ kernel there was a significant improvement in the performance. The ‘rbf’ kernel outperformed because of the correct classification of nonlinear data. The nonlinear regularization factors helped to increase accuracy.

As shown in the below accuracy and log loss plot, the accuracy of random forest support vector machine classification outperformed other machine learning algorithms like KNN.





Tools

Below are the details of the tools used in the project:

- **MNE library**: The dataset used in our project was a wave or signal EEG data whose recordings were stored in the EDF file format. We used the MNE library to read both PSG and hypnogram files to extract their features. This library was helpful because unlike CSV or TXT files it's not possible to use pandas or numpy libraries to read EDF files.
- **TensorFlow**: is a symbolic math library, and is also used for machine learning applications such as neural networks. We used this library to implement the LSTM model.
- **NumPy**: this library supports large multi-dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays. We used this library to store the read EDF file data into a numpy array that stores both EEG signals data and their corresponding sleep stage labels.
- **Sci-kit:** we used this library to implement a few machine learning models and display its corresponding classification reports and accuracy.
- **Matplotlib, seaborn**: are a plotting library. In this project, we used this library to plot

Tools that we tried and ended up not using:

We initially used pyedf library to read EDF files but later we switched to the MNE library. The reason being that though the pyedf library read PSG files correctly, it was challenging to read and display all the features in the hypnogram files. And also, it failed to save the EDF format of the hypnogram file into TXT or CSV file format. And the PSG files were saved in CSV file for all subjects each of size 1GB. We then used the MNE library to read and plot both PSG and hypnogram files and saved the corresponding numpy arrays in NPZ files for each subject.

Lessons Learned

The dataset used in this project is a physionet dataset which contains PolySomnoGraphic sleep recordings of subjects, we mainly focused on classifying sleep stages by considering a single channel of EEG Fpz-Cz. Unlike normal CSV or text file the format of the data files i.e. EDF (European data format) used in this project were new to all of us and we had to spend more time on initial preprocessing of these data like understanding the file format, its annotations, exploring

different libraries to read these files. As the data was not just in simple numbers or characters it was challenging to understand the few terms like an epoch, window size, sleep stages, frequencies of the wave, and what is the noise in such data. After successful representation of the signals data into numpy arrays, we got a better idea of the representation of EEG signals and how the labels i.e, sleep stages can be used to train and test the data to classify the signals.

Our prior research on the data and some existing works done on sleep stage classification motivated us to choose the models and feature extraction methods. PSD/Power Spectral Density was calculated by using the Welch and Burg Method to extract the features from filtered data. The models that we implemented are LSTM, Random forest, Support vector machine, KNN.

The main challenge that we faced while executing models is the large data of each subject. And the RAM did not support the execution of all the subjects as the system hanged.

LSTM: while implementing this model we understood the impact parameters tuning has on its performance. When the model was run on all the 151 subjects for 10 epochs the accuracy achieved was around 50%, however, for 25 epochs an accuracy of 61% was achieved. Since the RAM did not support large epochs, we considered only 50 subjects to train and test the model by running 30 epochs, an accuracy of 78% was achieved.

Support vector machine: we know that SVM is one of the well-known classification models, but initially we achieved only an accuracy of 30% because when we specify a linear kernel on SVM we imply that we are expecting that a line will separate the data, but generally we have a highly non-linear dataset. Hence the low accuracy score. By changing the kernel to 'rbf' and running SVM on the same 50 subjects we were able to achieve an accuracy of 60%.

Random forest: by implementing the random forest model we were able to achieve only an accuracy of 59% which was lower than we expected to achieve.

And by training the dataset using machine learning models like KNN, K Means we understood that the accuracy achieved for the EEG type of datasets is very low compared to other SVM, RF, and LSTM models. And the performance of the models depends on the characteristics of the data and the expected output like classification, clustering or prediction.

And one important lesson we learned is that understanding the characteristics of data and the initial preprocessing of data and feature extraction plays an important role in achieving a good performance of the model. Though we were able to achieve an accuracy of 78%, 60%, 59% for LSTM, SVM and RF, we believe that better performance could have been achieved by better feature extraction and filtering of the data.

Team Contributions

Supriya Ramachandra: Data Preprocessing: 60%, Feature Extraction: 15%, LSTM: 70%, SVM: 30%, other machine learning models: 10%, code integration: 15%, Project report & Presentation:40%

Erfan Jafari Khadem Zavareh: Data Preprocessing:20%, Feature Extraction:70%, LSTM: 20%, SVM: 10%, other machine learning models: 15%, code integration:70%, Project report & Presentation:25%

Swayanshu Shanti Pragnya: Data Preprocessing:20%, Feature Extraction:15%, LSTM:10% SVM: 60%, other machine learning models: 75%, code integration: 15%, Project report & Presentation:35%

University of Colorado Denver

References:

[1] Rechtschaffen, Allan. "A manual for standardized terminology, techniques and scoring system for sleep stages in human subjects." Brain information service (1968)

[2] Aboalayon, K.A.; Faezipour, M. Multi-class SVM based on sleep stage identification using EEG signal.

In Proceedings of the IEEE Healthcare Innovation Conference (HIC), Seattle, WA, USA, 8–10 October 2014;