Stage classification of sleep data

Recorded Presentation Link: https://www.youtube.com/watch?v=LrxD_XXKcOQ

Gatech GitHub Link: https://github.gatech.edu/aawasthi32/cse6250Project

Anshuta Awasthi, M.S. [1], Jai Motwani, M.S. [2], Ng Wee Liang, M.S. [3,] Phat Nguyen, M.S. [4]

Georgia Institute of Technology, Atlanta, GA, USA

Abstract

Analysis and study of sleep patterns is very important for evaluating the sleep quality and sleep disorders. The sleep stage scoring has been done traditionally by human expert technicians over many years and needs a significant amount of time and human effort. The aim of this project is to build a system to generate the sleep scores from the polysomnographic (PSG) sleep data automatically. The proposed system first leverages the Big Data techniques to extract and transform the vast amount of PSG sleep data in a distributed environment. The transformed data, extracted from the single most important EEG channel is then used to train the machine learning models like Support Vector Machine, Random Forest Classifier and deep learning models like MLP and GRU, in order to predict the sleep stages. The study also compares the performance of these models and discusses the results of different approaches. The current best performance model is Variable Recurrent Neural Network with test accuracy score of 77.7%.

Introduction

Sleep disorders are found to be the cause behind many serious illnesses among people of all ages. Due to the sharp increasing trend in sleep disorders worldwide, the World Health Organization has termed it as "Global epidemic of sleeplessness" [9]. It is also a major contributing factor of healthcare and welfare cost in many countries [6].

Over the years, analysis and study of sleep patterns has been an important topic among researchers. EEG-polysomnography is the most widely used monitoring technique to record the sleep patterns, in which the brain activity and eye movement signals are recorded by various sensors attached to the body of the subject. According to sleep researchers of American Academy of Sleep Medicine (AASM), the average human goes through 5 stages during ideal night sleep. These 5 stages of sleep cycle are defined as: W (Wake), R (Rapid Eye Movement), N1 (Non-REM Stage 1), N2 (Non-REM Stage 2), N3 (Non-REM Stage 3). The entire PSG signal data of a full night sleep of each subject is partitioned into epochs of 30 seconds and one of the 5 sleep stages are assigned to each epoch [8]. Traditionally, the scoring of sleep stages is done manually by human expert technicians, which requires a significant amount of time and effort to label the sleep stages [1].

It is not possible to meet the demand of sleep pattern analysis by human expert technicians with ever increasing sleep disorders, all over the world. Hence, the reliable system to automate the sleep stage classification process is crucially on-demand.

The proposed study is performed on the ISRUC dataset, which consists of sleep data collected from 100 subjects with sleep disorders. Out of 11 extracted channels in the dataset, the single most important EEG channel, 'F4-A1' is used in the computation of sleep stages. Prior research shows that data collected from the sensor placed at 'F4-A1' gives the best classification accuracy [15]. The proposed system attempts to solve this multi-label classification problem by applying various ML models. Multiple Machine Learning models such as SVM, Random Forest, MLP and RNN were implemented to classify sleep stages. Design, Implementation details and results of these models are discussed in later sections of the draft.

Approach/Metrics

Software and Hardware Specifications

We are using the following software stack:

For big data extraction and processing: Python v3.7.9 - PySpark v3.1.1 - SciPy - Pandas

For Machine Learning and Deep Learning models: PyTorch v1.6 - Scikit-learn - Matplotlib v3.1.0 - CudaToolkit v9.2

The configuration of the machines where proposed system is getting executed and implemented is given below:

Machine 1 – Windows 10 – Intel Core i7 7700HQ 4 cores (8 threads) – 32GB RAM – 4GB NVIDIA 1050Ti.

Machine 2 – Windows 10 - Intel i7 CPU, 16GB DDR4 RAM, NVIDIA GeForce GTX 1060-6GB

Dataset

This dataset consists of randomly selected PSG recordings from the sleep data obtained from 3 different types of subgroups and the subject's sleep was recorded for around 8 hours night-time: 1) Subgroup-I: One data acquisition session per subject performed on 100 adults with sleep disorders. 2) Subgroup-II: Two data acquisition performed on different days on each of the 8 adults with sleep disorder. 3) Subgroup-III: One data acquisition performed per subject, performed on 10 healthy adults.

ISRUC data files are available in .mat file format [8] which are obtained after the raw data signal is passed through 2 filtering stages, in order to improve the quality and eliminate noise: (a) a notch filter for the removal of 50 Hz electrical noise and (b) a bandpass Butterworth filter with the lower and higher cutoff frequencies of 0.3 Hz and 35 Hz respectively. The data is then partitioned into epochs of 30 seconds at the frequency of 200 Hz and made available in. mat format. Each 30-second epoch is then assigned one of the 5 sleep stages manually by two experienced clinical technicians as per AASM rules. For the draft version we have trained models on randomly selected data of 20 Subjects from Subgroup I. For each subject, there are about 700-980, namely N, epochs which in turn have 6000 recordings. In other words, extracted data from a single channel('F4-A1') is a multivariate-time series of dimension Nx6000, where N is in the range 700 to 980. For 100 subjects, we will have approximately 80000 data instances with 6000 features. The epochs and corresponding hypnogram labels are available in a text file in the dataset (6,7,8). The plot shows the distribution of sleep stage classes in data, which shows some imbalance in classes. The percentage of occurrences of "W(Wake)" stage are more in patients with sleep disorders as compared to healthy patients. The below plot shows that 'W' and 'N2' sleep stages, are the majority classes present in the data while 'N1' and 'R' stages are present with least number of occurrences.

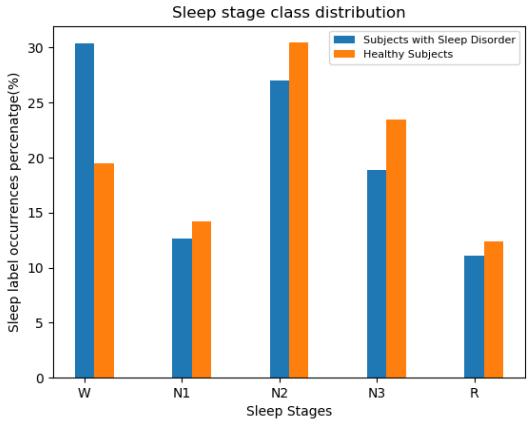


Figure 1. Sleep stage class distribution

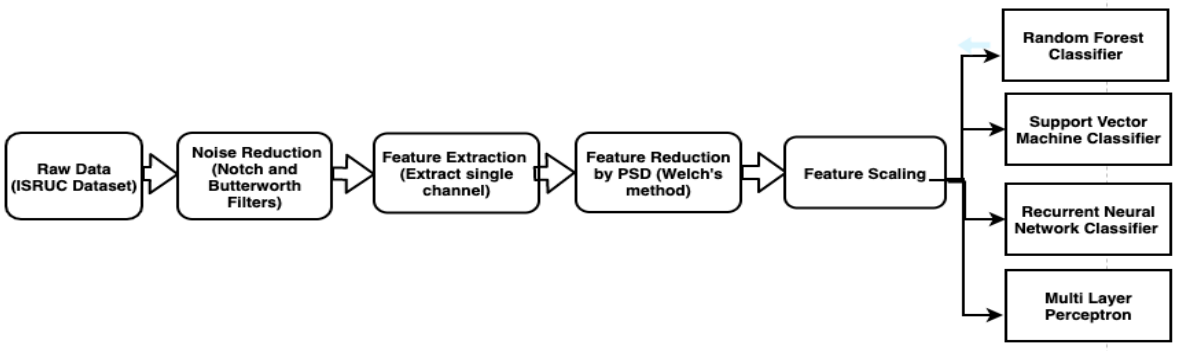


Figure 2. Sleep stage data ETL and model training pipeline

Data Processing Pipeline

The system has a data pre-processing pipeline created in PySpark to utilize parallel and distributed Apache Spark engine in order to process the vast amount of time series data. The first step of the pipeline reads the data from the Matlab .mat files and the corresponding labels from hypnogram text files into X and y PySpark Data Frames respectively. The last 30 labels from y data frame are removed since the corresponding data files are 30 epochs smaller than their corresponding hypnograms. These last 30 epochs are removed to reduce noise from the data. In the next step of the pipeline, data from only the single most important channel ('F4-A1') is extracted. The raw data sampled at the frequency of 200Hz for 30 sec had 6000 values of signal amplitude for each epoch. The strength of the signal is measured in terms of Power Spectral Density using Welch's method [18]. PSD is a very relevant metric to classify sleep stages as it measures the band power of 5 frequencies (delta, theta, alpha, sigma and beta) that are dominating the stages of sleep in the EEG signal. Scaling of features (frequency band powers) obtained is done by averaging them over the short segments of windows, which drastically reduces the variance in the data [10]. The figure below shows transformation of raw signal epoch to Welch's periodogram. The extracted and processed data is then used to train ML and DL models using libraries like scikit learn and PyTorch.

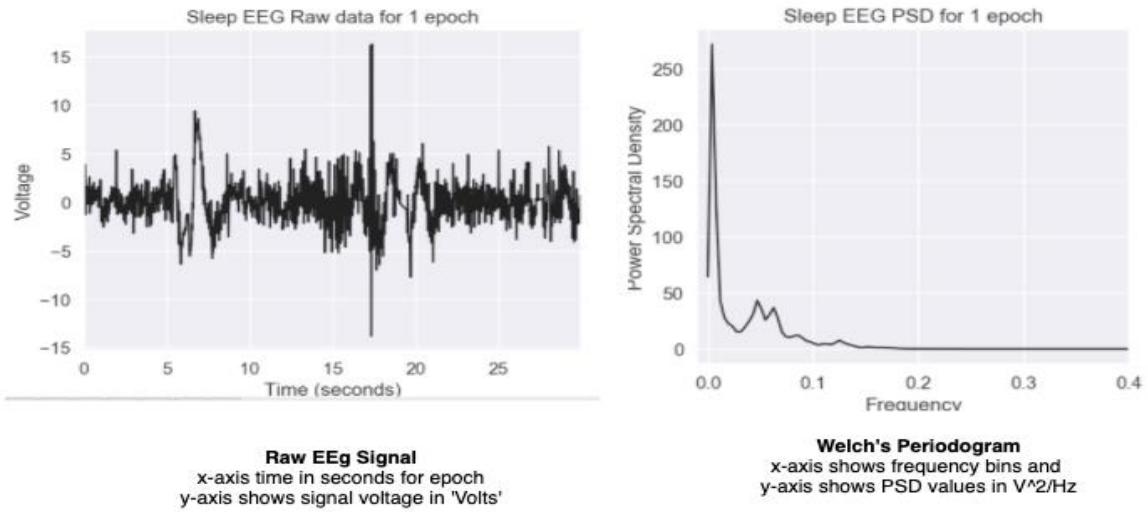


Figure 3. Raw EEG signal voltage and Welch's Periodogram

Model Training

Following models have been trained on the given data and results are analyzed over a set of metrics. For the Project Draft, our team choose to train with Accuracy as metrics. Learning curves and confusion matrices have also been plotted. Learning curves for the support vector machine and random forest model are obtained by splitting the dataset 5 times in training and test data and the accuracy scores of training and test subsets are computed by averaging over 5 runs. The groupKFold cross-validated training and test scores are plotted in the curves [14]. Our team used groupKFold so that data points from a particular patient belong exactly to either the train or the validation datasets but not both. For both deep learning models, the learning curves are constructed by appending the training and validation loss after each epoch and best fitted model is saved and used for computation of test accuracy.

Support Vector Machine: Support Vector Machine algorithms are mainly used for binary classification and do not support multi-class classification natively. In order to apply these methods to solve multiclass problems, one-vs-one strategy has been followed in sklearn. One-Vs-one(OvO) strategy allows these algorithms to classify the data into more than 2 classes by splitting the multi-class classification dataset into binary classification problems. This strategy attempts to fit one classifier per class pair. At prediction time, the class which received the most votes, is selected. Since it requires to fit n_classes * (n_classes - 1) / 2 classifiers, this method is usually slower [3]. In the next step, support vector machine (SVM) classifier with 'rbf' kernel is defined with the value of parameter decision_function_shape as 'ovo' for one-vs-one classifier. The model with the best hyperparameter is chosen with GridSearchCV and the results are computed on the best estimator. After data is processed through the pipeline mentioned above, the model is trained on data from 20 subjects and could achieve the test accuracy of 60.2%

Random Forest: Multiclass Random Forest is also implemented on the subset of 20 subjects (16 randomly selected from the train dataset, 2 randomly selected from the validation dataset, and 2 randomly selected from the test dataset). First, we fit a Min-Max-Scaler model to the full training dataset and then transform the sub-datasets of patients mentioned above. Due to the vast number of features (columns), our team also decide to perform PCA on the scaled datasets. We used the first 500 largest principal components in order to obtain > 90% variance explained by the train sub-datasets: 92.87% specifically. Then, we ensure that all the recordings from a single patient belong to exactly either the "train" datasets or the "cross-validation" datasets by utilizing sklearn's GroupKFold and a 5-fold cross-validation. Training Random Forest on the sub-train datasets (16 patients) and predicting on the sub-test dataset (2 patients) gave a mean cross-validation accuracy of 45.39% and a test accuracy of 47%. We additionally train the Random Forest model on the full min-max-scaled PCA-dimension-reduced training datasets. The first 500 largest principal components obtain 92.7% variance explained for the full training dataset. Training Random Forest on the full train dataset (90 patients) and predicting on the full test dataset (10 patients) gave a mean cross-validation accuracy of 48.92% and a test accuracy of 49% respectively.

In order to improve mean cross-validation and test accuracies, our team try applying the Welch's method on computing the power spectral density, PSD, on each subject's 30-second epochs [18]. Specifically, the Welch's PSD approach reduces the number of features from 6000 to 38 features by keeping only the features which have satisfied the specified minimum and maximum frequency: 0.5 and 30 respectively. Then, we reduce the resulting 38 features to 5 specified frequency bands namely: delta, theta, alpha, sigma, beta to be able to predict sleep stages from the EEG signals. Here, we decide to average the number of segments using either mean or median and test their performances. As a result, the fully trained Random Forest on PSD-transformed data using mean number of segments achieved a test accuracy of ~60% while that of median number of segments was ~58%. This was sufficient improvement from the previous Random Forest models on min-max-scaled PCA-transformed data.

Deep Learning Model (Multi-Layer Perceptron) MLP: Multilayer Deep Neural Network is implemented on 30 subject's data. 20 subjects are used for training of model. 5 subjects are used for cross validation and 5 subjects for testing of model. Initially, we tried training the MLP model by taking F4_A1 Channel data extracted and transformed (Min-Max Scaled) from PySpark pipeline. We tried using SparkTorch and PetaStorm [12, 17] libraries to directly consume PySpark DataFrames in PyTorch model. But we encountered many version incompatibilities issues with these libraries. We saved the subject data for F4_A1 channel, extracted and transformed using PySpark. For each subject This data had dimension of (850-900) Epochs and 6000 input features. We trained our MLP DL model with multiple attempts on this data. We could not get the accuracy above 30-32%.

In order to improve our features, we applied Welch's method for computing PSD as shown in Figure.2 [18]. It reduced the number of features from 6000 to 129. We did not try to reduce the features further as DNN are capable of learning from raw data or higher number of features. Our MLP had 3 hidden layers of 64, 32 and 16 Neurons respectively. There were also input and output layers. Sigmoid function was applied after each layer except for output layer. We could get test accuracy of 64.67 % by training the model for 200 Epochs with learning rate of 0.0001. We also tried adding dropout layers and making classification weight balanced but it did not improve the accuracy significantly. Due to compute limitations of our machines, we could not train our model on all the data.

Deep Learning Model (Variable RNN): A bidirectional GRU (gated recurrent unit) RNN is implemented on 100 subjects' data. 81 subjects were used for training the model. 9 subjects are used for validation and 10 subjects were used for testing the model. Cross validation was not done due to limitations of the team's member hardware and time constraints. The data from PySpark was converted to parquet format before it is read using Pandas, where is transformed into a list (subjects) of list (epochs) of list (features). Finally, the list is converted into a 3D tensor (subjectIDs x Epochs x features). Similar to training the MLP model, we initially tried using Min Max Scaling to scale the features but could not get the accuracy above 40%. Eventually, we used the Welch's method which reduced the number features to 129 and the accuracy improved beyond 50% [18]. We then proceed to test different architectures with different arrangement of fully connected neural network layer and GRU layer and finally settled with the following: a fully connected neural network layer of 32 neurons, followed by a 3-layer bidirectional GRU of 32 neurons each and finally a fully connected layer of 64 neurons. Relu function was applied after the first fully connected layer. After training model for 100 Epochs with learning rate of 0.001, we could get a test accuracy of 77.7%.

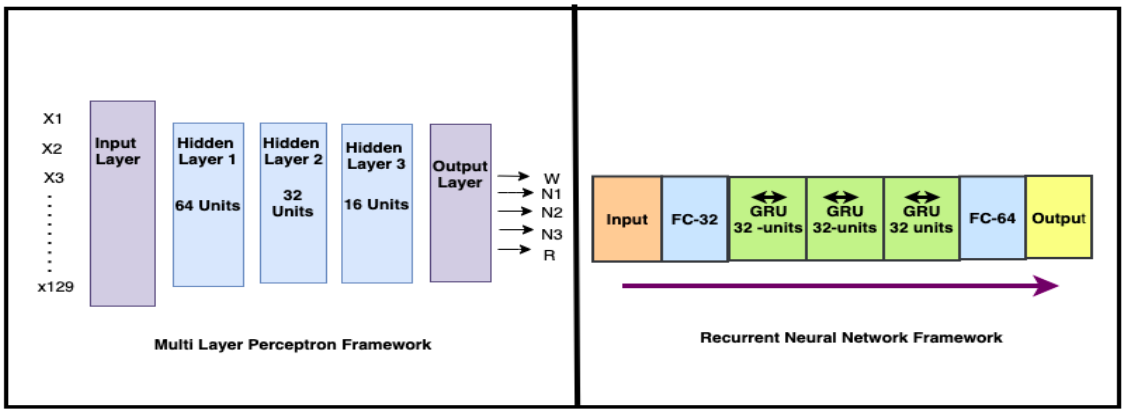


Figure 4. Our Proposed MLP and RNN Model Overview

Experimental Results

Experimented model performance across various evaluation metrics is reported below. These are the best possible values achieved till the submission date on the given models. **Detailed learning curves of each model are included in Appendix A.(Figure 7)**

Table 1. Model Summary Train/Test Performance Metrics

		Accuracy	F1 Score	Recall	Precision	Cohen's Kappa
Variable Recurrent Neural Network	Training	0.817	0.807	0.817	0.813	0.756
	Test	0.777	0.772	0.777	0.772	0.703
Random Forest	Training	0.632	0.670	0.671	0.702	0.628
	Test	0.601	0.533	0.550	0.540	0.470
Multi-Layer Perceptron	Training	0.702	0.692	0.702	0.699	0.601
	Test	0.663	0.640	0.663	0.641	0.562
SVM	Training	0.607	0.547	0.607	0.644	0.461
	Test	0.602	0.551	0.602	0.573	0.458

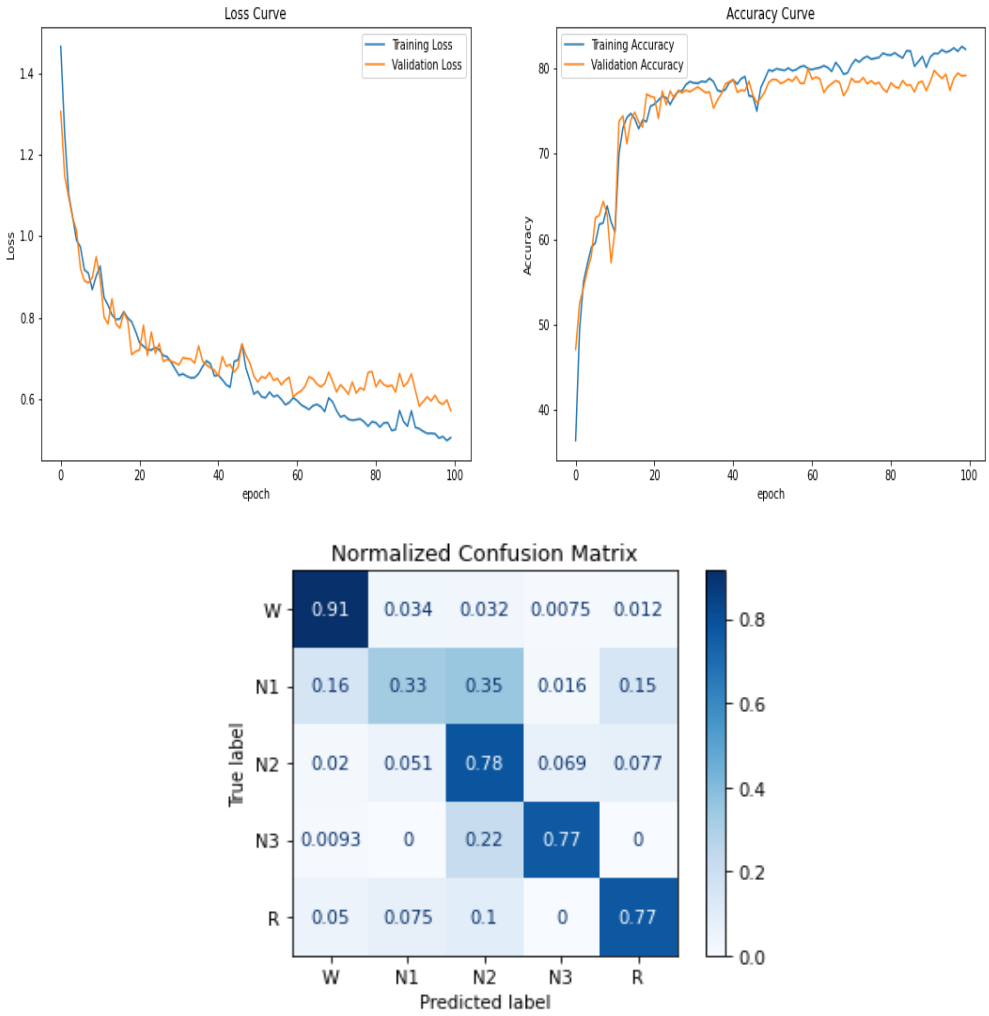


Figure 5. Loss/Accuracy Learning Curves and Normalized Confusion Matrix for the best performance model – RNN with Bidirectional GRU

Discussion

According to above results the highest test set accuracy of 77.7% is achieved by bidirectional RNN followed by MLP model with 66.3%. The machine learning models, RF and SVM are not far behind too and both the models are classifying the data points with approximately 60% accuracy.

The proposed study shows that sleep stages with higher percentage of occurrence are predicted better than the sleep stages with less no. of occurrences, namely the minority classes. In addition, even though the Random Forest test performance of using the mean segments, ~60%, and using the median segments, ~58%, is marginal, the true positive rate for class label 5, namely R (Rapid Eye Movement), decreases significantly for the median segments. Furthermore, the class label 1, namely N1 (Non-REM Stage 1),'s true positive rate was extremely low: in the range of 0-20%. This observation agrees with many published studies that N1 label had been tremendously challenging to be correctly classified [7, 15-16].

Previously when SVM was trained on only 20 subjects, the model achieved the test accuracy of 59.4% but it was severely overfitted as training set accuracy was significantly higher than test set accuracies. Training the model on the data from 100 subjects not only improved the accuracy marginally to 60.2% but also reduced the overfitting. Cross validation and hyperparameter tuning were already in place but increasing the data to train the model has helped the model generalized well.

From the confusion matrices of all the models, we can tell that the models performed poorly in classifying N1. Figure 5. below shows the confusion metrics from all the models, and it is quite evident that majority classes 'W' and 'N2' have been classified with 80-90 % of accuracy. 'N3' has also been classified reasonably well with around 80% predicted labels matching true labels. However, simple ML models (Random Forest and SVM) failed to classify classes 'N1' and 'R' properly. Hence bringing down the overall accuracy. While RNN with GRU has successfully predicted minority class 'R' with 77% label match and 'N1' with 33% which is significant improvement for higher overall accuracy than ML models such as SVM and Random Forest:

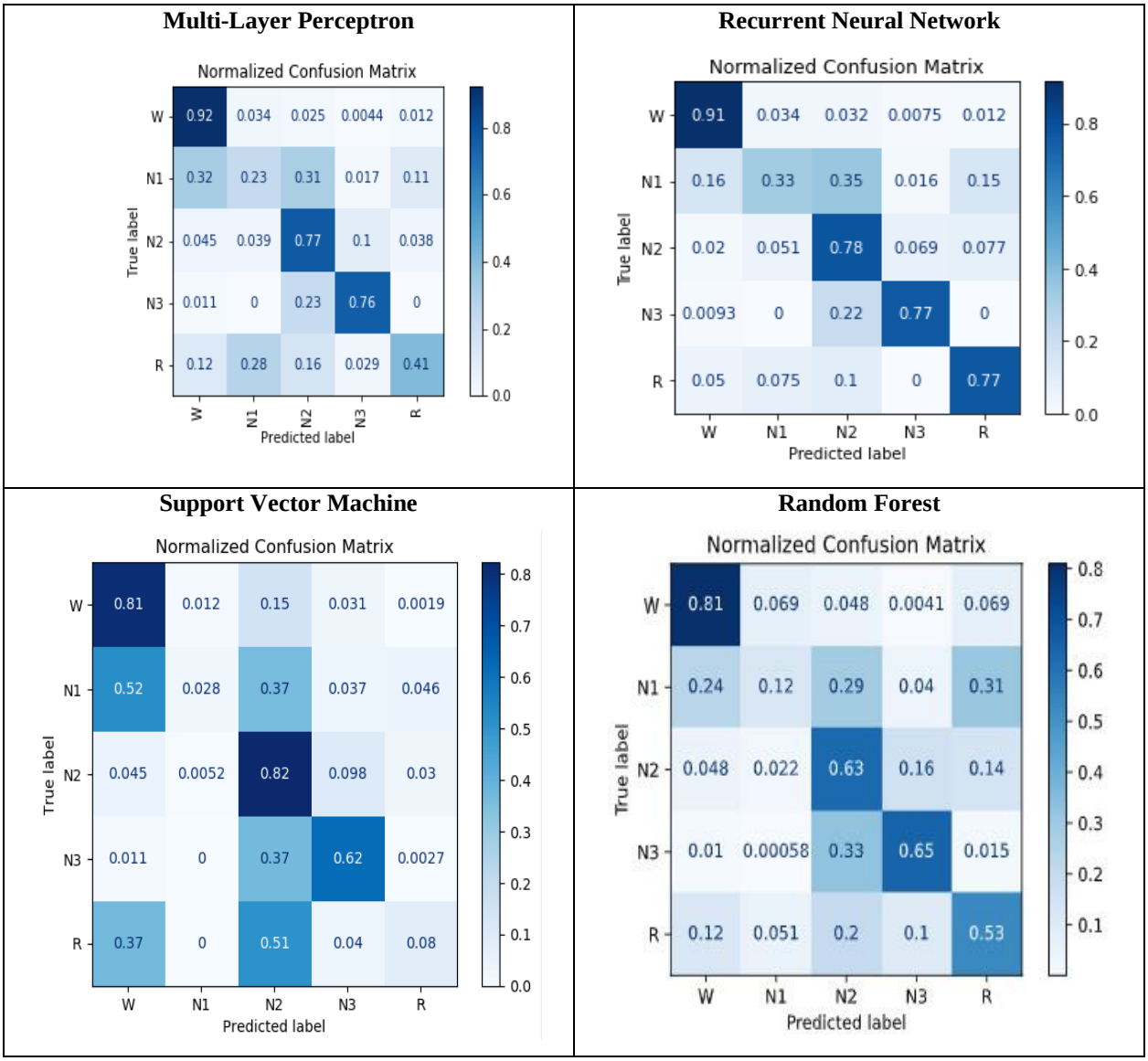


Figure 6. Normalized Confusion Matrices for the currently best test performance of each model

Conclusion/Optimization

The proposed models could not exceed the results obtained by state-of-the-art frameworks, but the system was able to achieve reasonable accuracies given the resource and time constraints.

The results also justify the fact that Recurrent Neural Networks with GRU layers perform best on multivariate time-series data, as they can capture the temporal dependencies in time series. In addition, gated cell architecture can hold information learned in previous time steps to make educated predictions on the test data.

On the other hand, machine learning models like Random Forest and SVM have provided decent accuracies on limited amount of data with limited processing power, while MLP and RNN needed high processing powers and learn from

large amount of data. It was also observed that linear models like Logistic Regression and SVM with 'linear' kernel are not performing well as the sleep data is highly correlated and non-linear.

To get better result from models, we should have more data possibly from other channels as well. It was a challenge to understand the sleep signal data and extracting the relevant subset of meaningful data from the vast amount of raw data available. With the limited time and resources at this stage, models are little short of the accuracy values achieved by state-of-the-art architectures. Also, we have extracted only the single most important channel to keep the model architecture simple and doable in the limited time but ignoring the data from other channels has contributed to information loss and can be a possible reason for low accuracy values. Further, in many previous studies, not only EEG channels but combinations of EEG signals along with, EOG and EMG signals have also contributed to better performance. In the future, we are planning to explore complex ML models by increasing number of hidden layers in Neural Networks and stacking different types of ML models to improve the accuracy further.

Challenges and Lessons Learned

Throughout this project, our team finds it challenging to understand the raw signal data and apply methods such as Welch's PSD to extract meaningful features and transform them into useful insights for the models. In addition, the vast amount of data from the 100 patients makes it impossible for our team to load and process all files entirely. Thus, we have to utilize the distributed data processing framework, PySpark, and choose the most important single channel shared across many datasets cited in published works. Lastly, constructing a uniform PySpark ETL pipeline to perform data engineering and training different Machine Learning and/or Deep Learning models prove to be tremendously difficult for our team due to module compatibility. For instance, frequency processing module may not have functionality available to apply with PySpark DataFrame.

Though we observed that increasing number of epochs, tuning learning rate, adding more layers and neurons and implementing complex Neural Network architecture is helping in improving accuracy, absence of high configuration GPU machine limited us from exploring them beyond 100 epochs. To overcome this problem, we divided the task of training different models among all four team members. We also limited number of subjects data to 20 in order to validate our models initially.

Along with the mentioned challenges, our team has the chance to learn processing real-world raw audio frequency data from distributed loading of vast amount of data and applying various transformation techniques. We also have the chance to update our knowledge about the state-of-the-art progress achieved by dedicated researchers around the world on Sleep Stage Automatic Classification. Finally, we can present our project in accordance with the standard requirements for a research paper.

Team Contributions

All four team members have contributed equally to all the required coding, write-up, and presentation-recording tasks for this project.

References

1. Berry RB, Budhiraja R, Gottlieb DJ, Gozal D, Iber C, Kapur VK, et al. Rules for Scoring Respiratory Events in Sleep: Update of the 2007 AASM Manual for the Scoring of Sleep and Associated Events. J Clin Sleep Med JCSM Off Publ Am Acad Sleep Med. 2012 Oct 15;8(5):597–619.

2. Chambon, S., Galtier, M., Arnal, P., Wainrib, G. and Gramfort, A. (2018)A Deep Learning Architecture for Temporal Sleep Stage Classification Using Multivariate and Multimodal Time Series. IEEE Trans. on Neural Systems and Rehabilitation Engineering 26: (758-769).

3. Duan, Kaibo et al. "One-Versus-One and One-Versus-All Multiclass SVM-RFE for Gene Selection in Cancer Classification." EvoBIO (2007).

4. Extracted Channels | ISRUC-SLEEP Dataset [Internet]. [cited 2021 Mar 12]. Available from: https://sleeptight.isr.uc.pt/?page_id=76

5. HR Colten, BM Altevogt. Institute of Medicine (US) Committee on Sleep Medicine and Research. Sleep Disorders and Sleep Deprivation: An Unmet Public Health Problem [Internet]. Washington (DC): National Academies Press (US); 2006 [cited 2021 Mar 13]. (The National Academies Collection: Reports funded by National Institutes of Health). Available from: http://www.ncbi.nlm.nih.gov/books/NBK19960/

6. Kales A, Rechtschaffen A, University of California LA, Brain Information Service, National Institute of Neurological Diseases and Blindness (U.S.). A manual of standardized terminology, techniques and scoring system for sleep stages of human subjects. Washington, DC: United States Government Printing Office; 1968.

7. Khalighi S, Sousa T, Pires G, Nunes U (2013) Automatic sleep staging: A computer assisted approach for optimal combination of features and polysomnographic channels. Expert Systems with Applications 40(17): 7046-7059.

8. Khalighi S, Sousa T, Santos J, Nunes U. ISRUC-Sleep: A comprehensive public dataset for sleep researchers. Comput Methods Programs Biomed. 2015 Nov 2;124.

9. Lyon L. Is an epidemic of sleeplessness increasing the incidence of Alzheimer's disease? Brain. 2019 Jun 1;142(6):e30–e30.

10. M. Otis,Jr. Solomon, PSD Computations Using Welch's Method, Dec. 1991

11. McHugh ML. Interrater reliability: the kappa statistic. Biochem Medica. 2012;22(3):276–82.

12. Petastorm Library page. https://github.com/uber/petastorm

13. Satapathy Santosh Kumar, Loganathan D. et. al. Automated Sleep Stage Classification Based on Multiple Channels of Electroencephalographic Signals Using Machine Learning Algorithm. 2020

14.Scikit Learn documentation:https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.learning_curve.html

15. Simões H, Pires G, Nunes U, Silva V (2010) Feature Extraction and Selection for Automatic Sleep Staging using EEG. ICINCO 3: 128-133.

16. Sousa T, Cruz A, Khalighi S, Pires G, Nunes U (2015) A two-step automatic sleep stage classification method with dubious range detection. Computers in biology and medicine 59: 42-53.

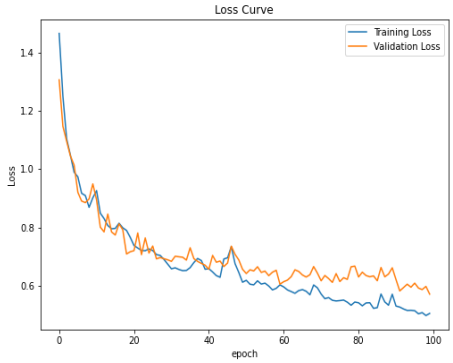
17. SparkTorch Library page. https://pypi.org/project/sparktorch/

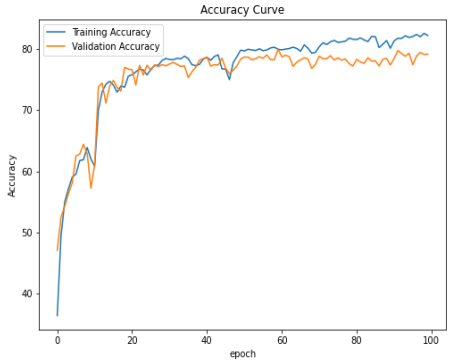
18. Welch's Method [Internet]. [cited 2021 Apr 18]. Available from: https://ccrma.stanford.edu/~jos/sasp/Welch_s_Method.html

Appendix A

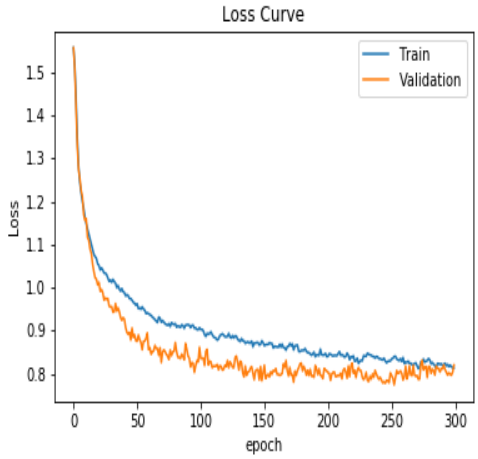
Learning Curves:

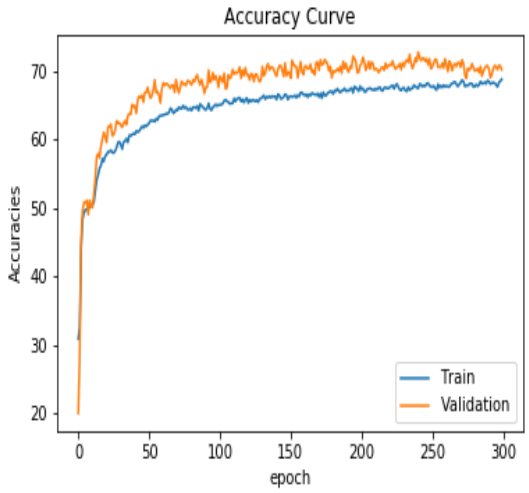
Bidirectional GRU:



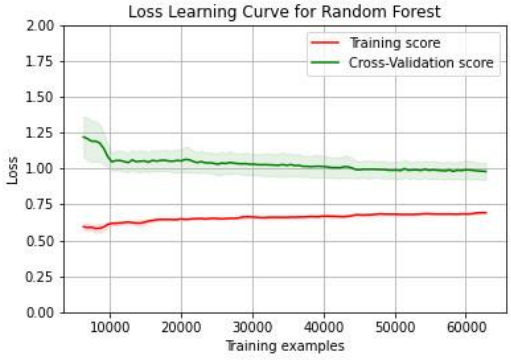


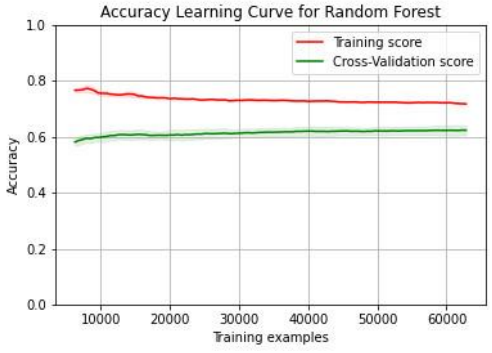
Multi-Layer Perceptron:





Random Forest:





Support Vector Machine:

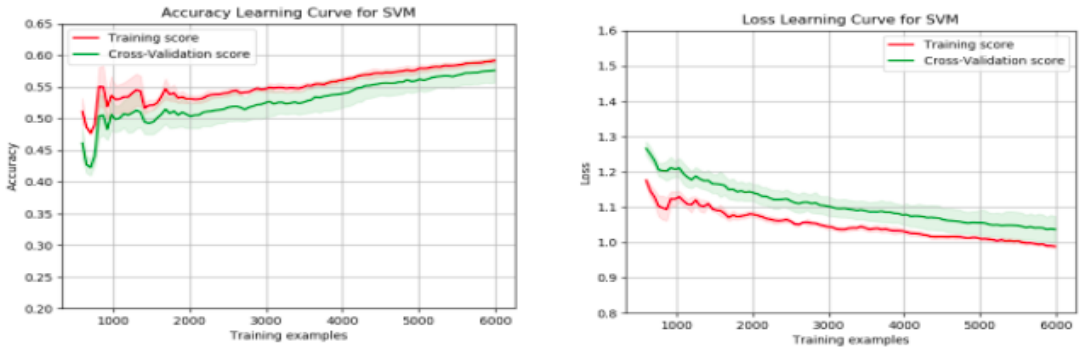


Figure 7. Loss/Accuracy Learning Curves for Various ML/DL models