Sleep Stage Classification

TEAM BAKA

ABHISHEK SAHU–ALOK KUMAR–BINAY KUMAR–SAI KIRAN
21111002 21111008 21111021 21111061

INTRODUCTION

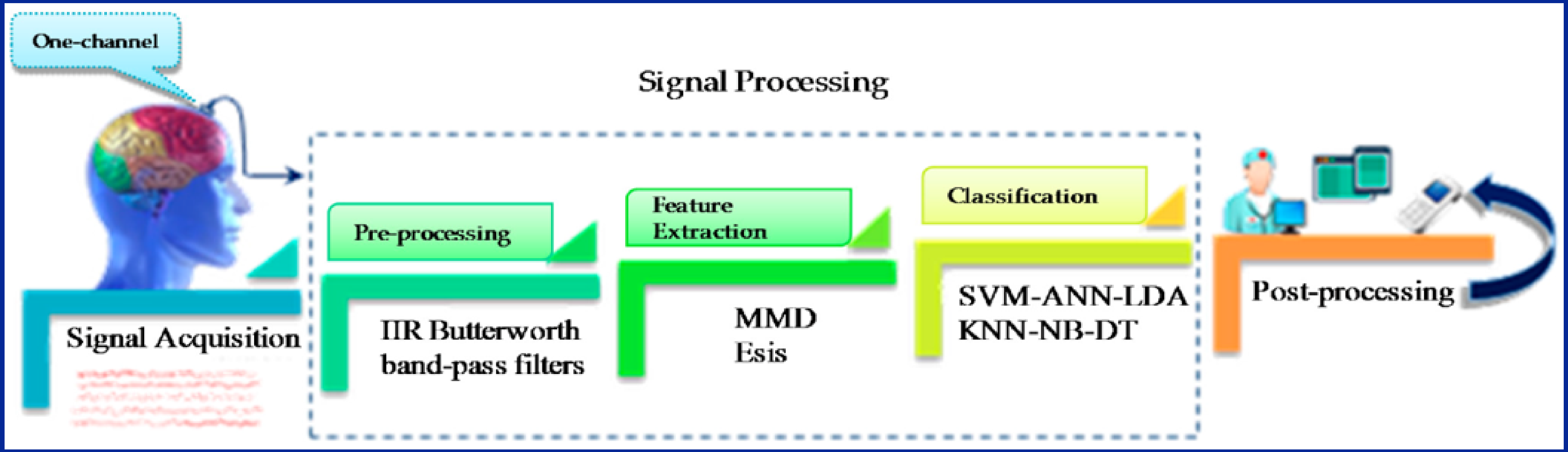
Sleep is the primary function of the brain and it plays an essential role in the performance, learning ability, physical movement and the Mental Health of an Individual.

Classification of sleep stages from EEG signals accurately is important due to the following reasons

1. To identify sleep-related conditions that include fatigue, drowsiness, or sleep disorders, such as Apnea, Insomnia or Narcolepsy
2. Manual sleep scoring is a time-consuming and prone to human error.
3. Expert personnel for manual scoring may not be available at all time. In such cases, smart EEG machines reduce the dependence on them.

PROBLEM STATEMENT

- The idea is to develop an algorithm of sleep staging classification, which is able to differentiate between Wake, N1, N2, N3 and REM on windows of 30 seconds of raw data.
- The Sleep Dataset contains 197 whole-night Polysomnographic sleep recordings, containing EEG, EOG, chin EMG, and event markers. Some records also contain respiration and body temperature.



LITERATURE REVIEW

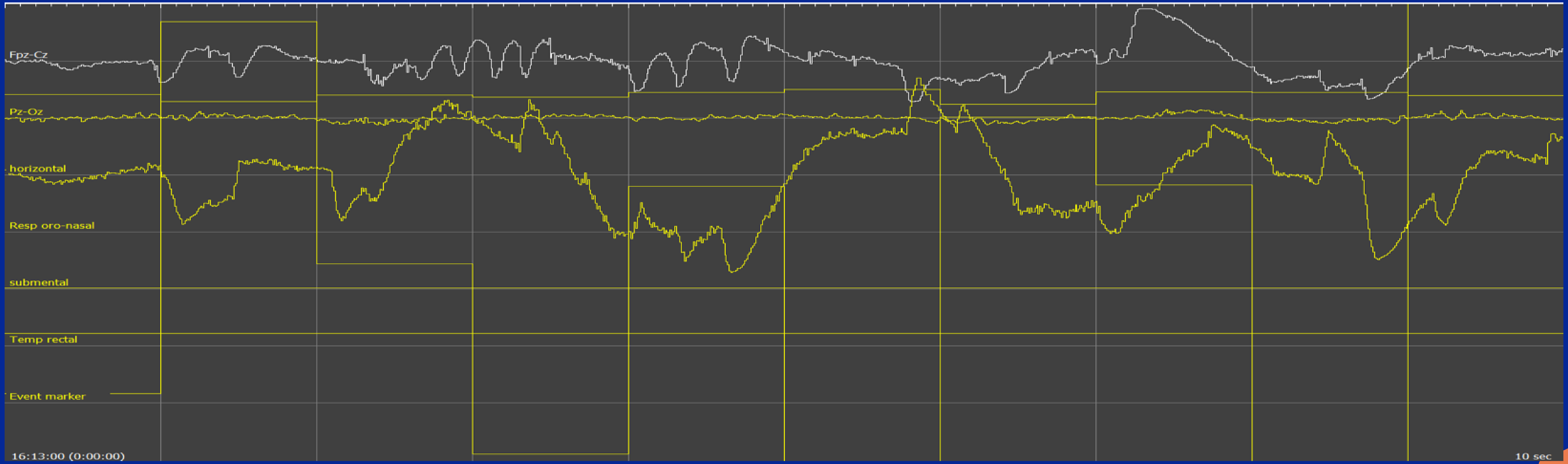
S.No	Author name	Classifier	EEG Dataset	Accuracy
1	Khald A.I.Aboalayon et al.	Multi-Class SVM	13 subjects	92%
2	Salih Gunes et al.	K-means clustering based feature weighting	5 subjects	82.21%
3	Luay Frauwan et al.	Random forest	16 subjects	83%
4	Aske B.Klok et al.	Random forest	100 subjects	92.6%
5	Plateleimon Chriskos et al.	KNN	23 subjects	81.69%
		SVM		89.07%
		Neural Networks		82.79%
6	Yang Yang et al.	CNN-LSTM	116 subjects	83%
7	Nicola Michielli et al.	LSTM-RNN	10 subjects	78%
8	Zhihong Cui et al.	CNN and Fine-Grained Segments	116	92.2%
9	Orestis Tsinalis et al.	CNN	20 subjects	82%

DATA COLLECTION

All the Data files collected are in EDF and EDF+ files.

Source: https://www.physionet.org/content/sleep-edfx/1.0.0/
DESCRIPTION: The source database contains 197 whole-night PolySomnoGraphic sleep recordings(PSG.edf), containing EEG (from Fpz-Cz and Pz-Oz electrode locations), EOG, chin EMG, and event markers. Here is a glimpse of the edf file.

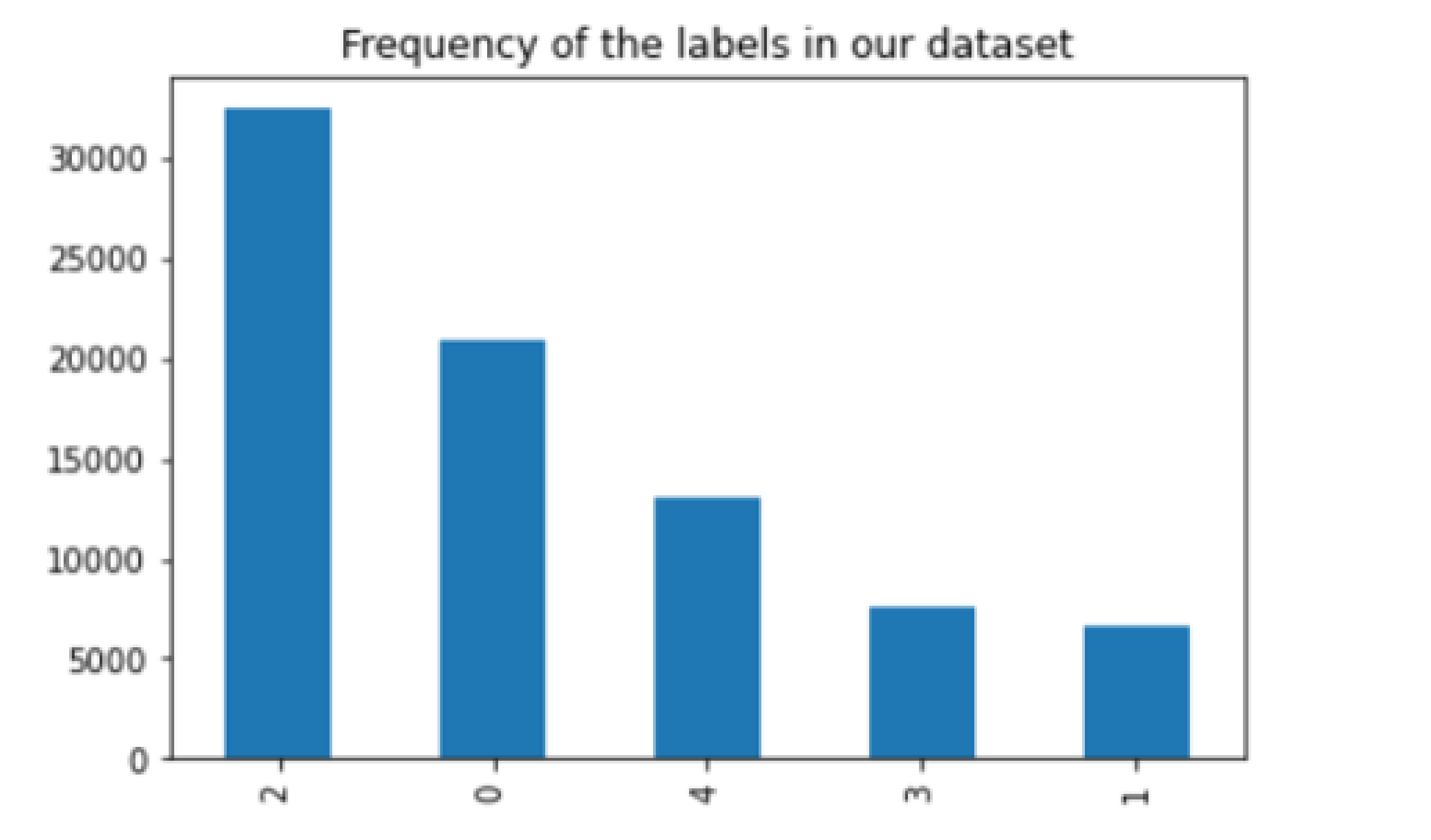
HERE IS A GLISMPSE OF THE EDF FILE



DATA PREPROCESSING

- 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.
- Next,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 was also removed by using a filter of this range, as this noise usually occurs at a much higher frequency.

- EEG Fpz-Cz are used for the purpose of this project as it gives higher accuracy (mentioned in the Literature)
- A 30 second epoch is taken and is divided into 3000 parts and a normalized dataset is prepared.



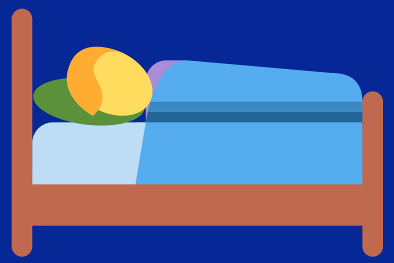
DATA PREPROCESSING

- We have divided the Dataset into Training and Testing Datasets in ratio 90:10.
- We have seen a problem of Class Imbalancing, to solve this problem we have done few things mentioned
   a. We have used an algorithm called SMOTE.
   b. SMOTE works by selecting close examples in the feature space by drawing a line between the standards in the feature space and drawing new samples at a point along that line.
   c. After previous step, we have classes having equal frequency.
- Then after the training data balanced, we have divided into training and validation dataset in ratio 80:20.

TOOLS USED

- Python 3.6
- Numpy
- Sklearn
- Tensorflow
- Libraries
  - MNE
  - pyEDFFlib
  - urllib
  - dhedfreader
  - matplotlib, seaborn (visulization)

PLATFORM: GOOGLE COLABS with GPU ENABLED



MODELS

- CNN-CNN MODEL
- CNN-LSTM MODEL
- RANDOM FOREST
- XGBOOSTING

In the next slides, we will discuss about the architecture

METHODOLOGY

CNN-CNN MODEL

In order to efficiently extract the features and transform the biologicalwaves to numbers , we included two CNN architectures in our model. we designed base model ,whichcan be used for both CNN-CNN and CNN-LSTM models with some minor changes .

DETAIL ABOUT BASE MODEL

layer	number	type	Kernel size
Input			
convo1D	16	Relu	10
convo1D	16	Relu	10
Maxpool1D			3
SpatialDropout1D			prob = 0.01
convo1D	32	Relu	5
convo1D	32	Relu	5
Maxpool1D			2
SpatialDropout1D			prob = 0.01
convo1D	128	Relu	3
convo1D	128	Relu	3
GlobalMaxPool1D			
Dropout			prob = 0.01
Dense	64	sigmoid	
Dropout			prob = 0.01

Detail of final CNN architecture

layer	number	type	Kernel size	Output
Input				(None, None, 3000, 1)
Time distributed				(None, None, 64)
convo1D	256	Relu	6	(None, None, 256)
Dropout			0.01	(None, None, 256)
convo1D	5	sigmoid	3	(None, None, 5)

PLATFORM: GOOGLE COLABS WITH GPU ENABLED
Loss Function: Sparse Categorical CrossEntropy
Optimizer: Adam
Metrics: Accuracy
Epoch=500
Overfitting checked by EarlyStopping which monitor **Validation Loss**

METHODOLOGY

CNN-LSTM MODEL

In order to efficiently extract the features and transform the biologicalwaves to numbers , we included one CNN architectures in our model. we designed base model, whichcan be used for both CNN-CNN and CNN-LSTM models with some minor changes .

DETAIL ABOUT BASE MODEL

layer	number	type	Kernel size
Input			
convo1D	16	Relu	10
convo1D	16	Relu	10
Maxpool1D			3
SpatialDropout1D			prob = 0.01
convo1D	32	Relu	5
convo1D	32	Relu	5
Maxpool1D			2
SpatialDropout1D			prob = 0.01
convo1D	128	Relu	3
convo1D	128	Relu	3
GlobalMaxPool1D			
Dropout			prob = 0.01
Dense	64	sigmoid	
Dropout			prob = 0.01

Detail of final LSTM architecture

layer	number	type	Kernel size	Output
Input				(None, None, 3000, 1)
Time distributed				(None, None, 64)
Bi-directional	100			(None, None, 200)
Dropout			0.1	(None, None, 200)
Dropout			.07	(None, None, 200)
conv1D	nlabel	sigmoid	1	(None, None, 5)

PLATFORM: GOOGLE COLABS WITH GPU ENABLED
Loss Function: Sparse Categorical CrossEntropy
Optimizer: Adam
Metrics: Accuracy
Epoch=500
Overfitting checked by **EarlyStopping** which monitor **Validation Loss**

METHODOLOGY

RANDOM FOREST MODEL

We have also deployed Random Forest for sleep pattern prediction.

RandomForest Architecture - We utilised 100 decision trees with max depth = 10.

XGBOOST MODEL

Another ensemble method used for sleep pattern prediction.

These method are lightweight and resource friendly

RESULT

CNN-CNN MODEL

Classification Report

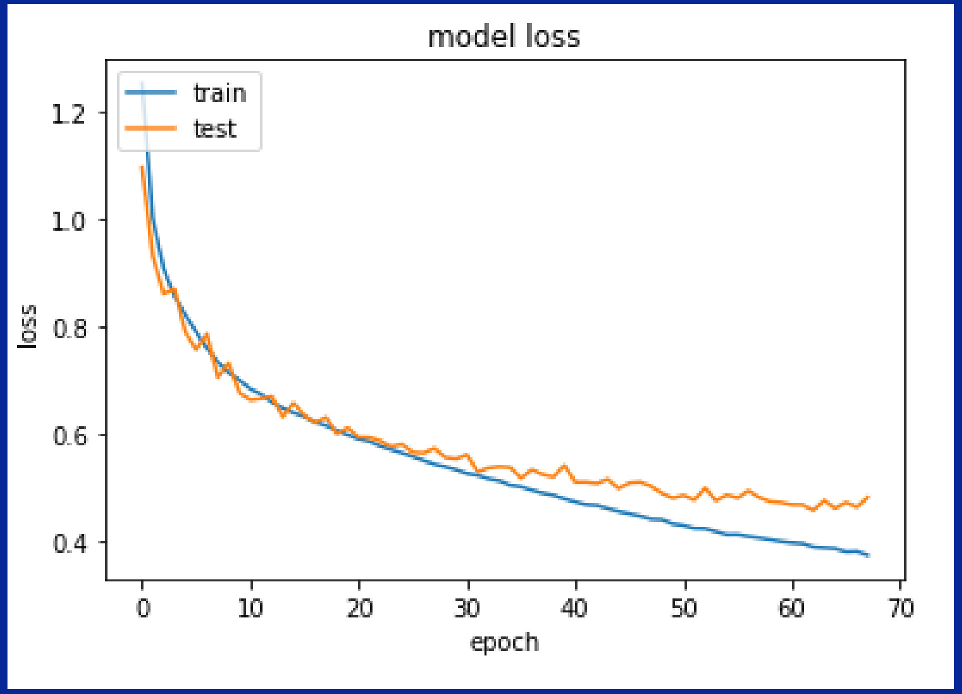
	precision	recall	f1-score
0	0.95	0.89	0.92
1	0.78	0.82	0.80
2	0.87	0.83	0.85
3	0.93	0.97	0.95
4	0.84	0.85	0.84
accuracy			0.87
macro avg	0.87	0.87	0.87
weighted avg	0.87	0.87	0.87

Training Data

	precision	recall	f1-score
0	0.92	0.86	0.89
1	0.72	0.74	0.73
2	0.80	0.77	0.79
3	0.90	0.95	0.92
4	0.78	0.79	0.79
accuracy			0.82
macro avg	0.82	0.82	0.82
weighted avg	0.82	0.82	0.82

Testing Data

Model Loss



RESULT

CNN-LSTM MODEL

Classification Report

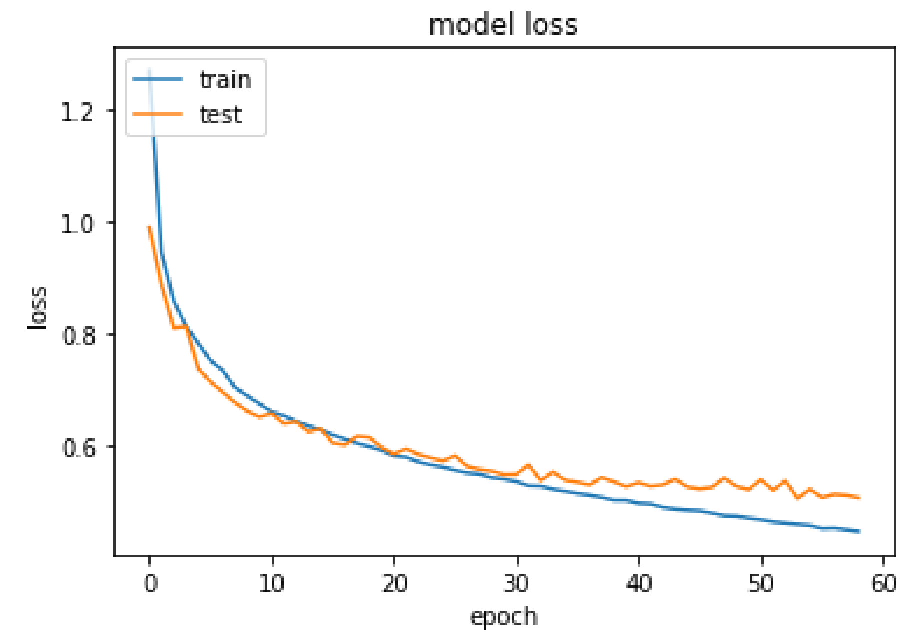
	precision	recall	f1-score
0	0.90	0.89	0.90
1	0.73	0.76	0.75
2	0.82	0.78	0.80
3	0.93	0.94	0.94
4	0.80	0.80	0.80
accuracy			0.84
macro avg	0.84	0.84	0.84
weighted avg	0.84	0.84	0.84

Training Data

	precision	recall	f1-score
0	0.87	0.87	0.87
1	0.67	0.70	0.68
2	0.75	0.73	0.74
3	0.91	0.92	0.92
4	0.75	0.73	0.74
accuracy			0.79
macro avg	0.79	0.79	0.79
weighted avg	0.79	0.79	0.79

Testing Data

Model Loss



RESULT

RANDOM FOREST MODEL

Classification Report

	precision	recall	f1-score
0	0.92	0.58	0.71
1	0.47	0.73	0.57
2	0.58	0.67	0.62
3	0.86	1.00	0.93
4	0.43	0.22	0.29
accuracy			0.64
macro avg	0.65	0.64	0.62
weighted avg	0.65	0.64	0.62

Training Data

	precision	recall	f1-score
0	0.77	0.34	0.47
1	0.46	0.71	0.56
2	0.46	0.48	0.47
3	0.68	0.99	0.81
4	0.38	0.20	0.26
accuracy			0.54
macro avg	0.55	0.54	0.51
weighted avg	0.55	0.54	0.51

Testing Data

RESULT

XGBOOST MODEL

Classification Report

	precision	recall	f1-score
0	0.79	0.43	0.56
1	0.55	0.70	0.61
2	0.58	0.58	0.58
3	0.77	0.94	0.85
4	0.49	0.47	0.48
accuracy			0.62
macro avg	0.64	0.62	0.62
weighted avg	0.64	0.62	0.62

Training Data

	precision	recall	f1-score
0	0.76	0.39	0.52
1	0.52	0.66	0.58
2	0.52	0.52	0.52
3	0.74	0.93	0.82
4	0.45	0.43	0.44
accuracy			0.59
macro avg	0.60	0.59	0.58
weighted avg	0.60	0.59	0.58

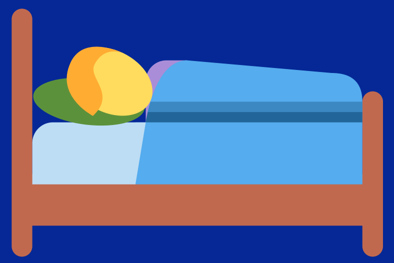
Testing Data

CHALLENGES

- Lots of noises present in collected data. Had to use advance signal processing techniques to filter them out
- There were many external factor affecting sleep pattern of the subject apart from the body EEG
- Training the model has been the major challenge as the access for fast GPU is limited. We could train it only a couple of times in a day.

CONCLUSION

- Sleep pattern datasets are usually very large and complex involves very intricate detailsthat need to be extracted as features.
- 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 82%, 89%, 59% for LSTM, CNN and RF, we believe that a better performance could have been achieved by better feature extraction and filtering of the data.



INDIVISUAL CONTRIBUTION

- **Abhishek**: Data Collection: 40%, Preprocessing: 30%, Model: CNN-LSTM model, Report: 30%
- **Alok**: Literature Review: 30%, Data Collection: 40%, Model: CNN-CNN model, Report: 30%
- **Binay**: Literature Review: 40%, Preprocessing: 40%, Model: XGBoost model, Report: 20%
- **Kiran**: Literature Review: 30%, Data Collection: 20%, Preprocessing: 30%, Model: Random Forestmodel, Report: 20%

Demonstration

Questions or comments ?

That Ends The Presentation

THANK YOU