Detection of Alzheimer's Disease using machine learning models.

CPSC 6300: Applied Data Science

Spring 2023

Sahana Surapureddy
ssurapu@clemson.edu
LSTM and Matrix Profiling
Introduction
Summary of Machine Learning Models
EDA

Akshitha Gunupati
agunupa@clemson.edu
FNN and Random Forest
Summary of EDA
Visualization
Summary and Conclusion

1 Introduction

1.1 Problem Statement:

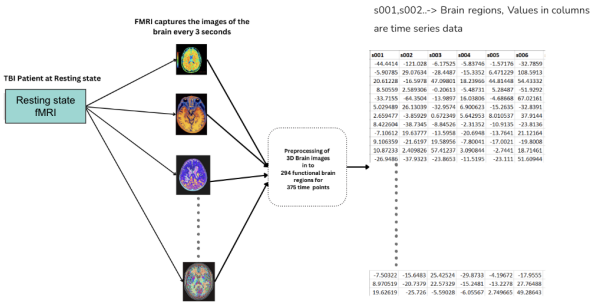
A patient with a TBI (Traumatic Brain Injury) status as positive may have a higher risk of pruning to Alzheimer's in future. This TBI has long term effects on human beings. TBI can be considered as one of the factors in detecting Alzheimer's. Our main motive is to build a classification model that can accurately predict if a patient has TBI or not based on time series data of brain regions of both TBI-Pos and TBI-Neg patients. This prediction of TBI acts as a crucial step in reaching closer to Alzheimer's detection in patients. Because it can aid in early diagnosis, care, and prevention. Building this model is significant because it will also aid in overcoming the difficulties that neural networks have when working with complex time series data and restricted data set features, making distinguishing between TBI-positive and TBI-negative individuals challenging.

1.2 Motivation:

We started analyzing multiple reasons that cause Alzheimer's and found out that a few factors like age, lifestyle, brain injuries and cardiovascular diseases might lead to Alzheimer's. We choose to analyze traumatic brain injury for our project. TBI can be considered as one of the factors in detecting Alzheimer's. A person affected with TBI might have cognitive impairment, emotional issues, and physical limitations and also disruptions in their brain functions. These long-term effects can be avoided with early identification and treatment, which also enhances patient outcomes. These multiple side effects of TBI have motivated us to build classification models that would provide a solution to many of the issues dealing with TBI. So, understanding a patient's TBI status might therefore assist healthcare practitioners in identifying those who are at higher risk of acquiring Alzheimer's and take preventative actions accordingly.

1.3 Explanation of Data set

The data is gathered from a dataset called ADNI. ADNI stands for Alzheimer's Disease Neuro Imaging Initiative. This takes the time series data information from Resting State Functional Resonance Magnetic Imaging (rs-fMRI). Below is a sample pictorial representation of how data is being collected.



In the above picture, rs-fMRI collects the data from a resting person. For every 3 seconds a 3D brain image is created and data is collected. Then it performs the preprocessing and segments the 3D image of the brain into 294 brain regions. This happens until 375 time points which in turn we call as time series data. This available data in ADNI after performing rs-fRMI is collected for our classification purpose from both TBI-Pos and TBI-Neg patients. We have 20 TBI-Positive patient's data and 16 TBI-Negative patient's data. Each patient's data is provided in the form of a csv file containing time

series data for those 375 time points and 294 functional brain regions.

2 Summary of your EDA

2.1 Unit of analysis:

In our project we are comparing TBI-positive to TBI-negative groups and building the model. Column names like s1, s10... in our data set stands for different brain regions. We have data of both TBI-Positive and TBI-Neg Patients data collected for 294 brain regions at 375 time points. Each row represents the time series data at each time point for 294 functional brain regions.

2.2 Total Number of Observations:

We have a total of 20 TBI-Positive patients and 16 TBI-Negative patients time series data for different brain regions. Each patient data consists of time series values for 375 time points for 294 functional brain regions.

2.3 Unique Observations:

The observations in the data set are collected from various patients, and since the data pertains to the metabolic profiles of each individual, all of them are unique.

2.4 Time Period:

The data set does not have a specified time period for the observations. However, the time series data inside each column stands for the signals received from the corresponding brain regions. The sampling rate is per every 3 seconds, a 3D brain image is created and time series data is created.

2.5 Data Cleaning:

First, we have loaded the CSV files for performing the analysis on them.

2.5.1 Checking for Duplicate values

We have checked null and duplicate values on all 36 TBI-Neg and TBI-Pos patients' data and removed if there are any.

For TBI-Neg patients:

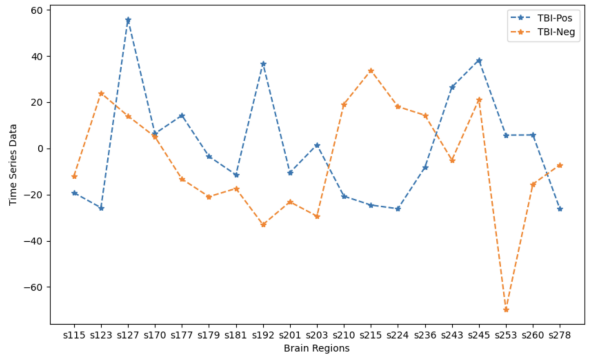
```
Checking for null values in TBI-Neg Data:
204-DOD-tts_all.csv has no null values
209-DOD-tts_all.csv has no null values
202-DOD-tts_all.csv has no null values
203-DOD-tts_all.csv has no null values
212-DOD-tts_all.csv has no null values
205-DOD-tts_all.csv has no null values
211-DOD-tts_all.csv has no null values
207-DOD-tts_all.csv has no null values
210-DOD-tts_all.csv has no null values
206-DOD-tts_all.csv has no null values
213-DOD-tts_all.csv has no null values
215-DOD-tts_all.csv has no null values
217-DOD-tts_all.csv has no null values
218-DOD-tts_all.csv has no null values
214-DOD-tts_all.csv has no null values
216-DOD-tts_all.csv has no null values
```

For TBI-Pos patients:

```
Checking for null values in TBI-Pos Data:
117-DOD-tts_all.csv has no null values
110-DOD-tts_all.csv has no null values
108-DOD-tts_all.csv has no null values
114-DOD-tts_all.csv has no null values
112-DOD-tts_all.csv has no null values
104-DOD-tts_all.csv has no null values
106-DOD-tts_all.csv has no null values
101-DOD-tts_all.csv has no null values
115-DOD-tts_all.csv has no null values
109-DOD-tts_all.csv has no null values
118-DOD-tts_all.csv has no null values
111-DOD-tts_all.csv has no null values
102-DOD-tts_all.csv has no null values
113-DOD-tts_all.csv has no null values
123-DOD-tts_all.csv has no null values
124-DOD-tts_all.csv has no null values
120-DOD-tts_all.csv has no null values
122-DOD-tts_all.csv has no null values
121-DOD-tts_all.csv has no null values
103-DOD-tts_all.csv has no null values
```

Observation: We don't have any null values for both the patients' data.

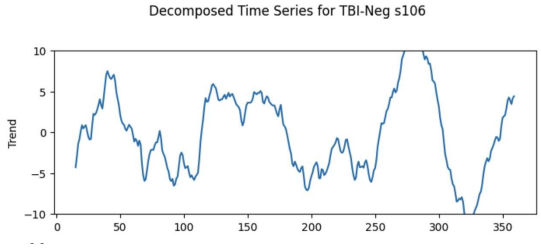
2.6 Visualization of Response

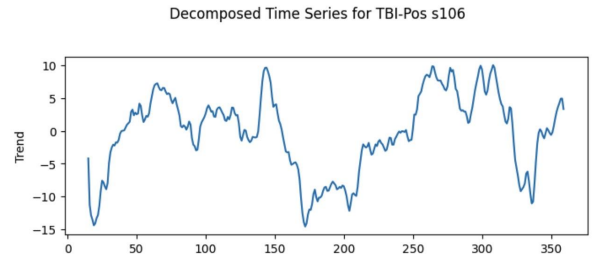


Observation: In the above plot we can see how the time series data of TBI-Pos and TBI-Neg have so many fluctuations. We can also infer the difference of strength of signals in the healthy and TBI-Pos brain.

2.7 Visualization of key predictors

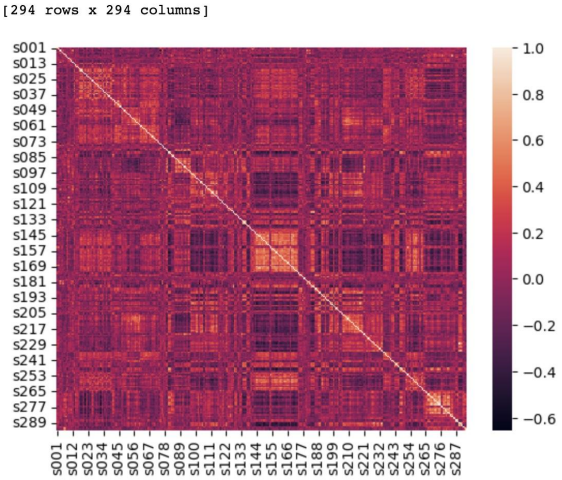
1. We checked the patterns in the TBI-pos and TBI-Neg data for a single brain region. In the below plot we can observe the trend in the data. For the below brain region, we can observe that the x-axis has more than 350 time series points and the y-axis represents the value of the trend component of the time series.





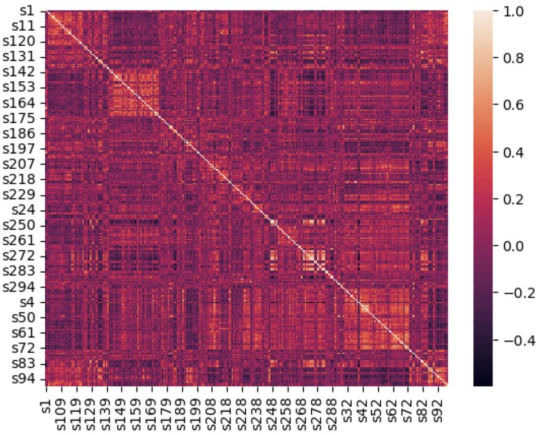
Observation: The patterns of the brain region for a TBI-Pos patient and TBI-Neg patient is varying, where the lines are moving in opposite directions at some points.

2. Correlation Matrix can help in identifying the relationships between different brain regions of TBI-positive patients. So, we created a heat map to visualize the correlation matrix.



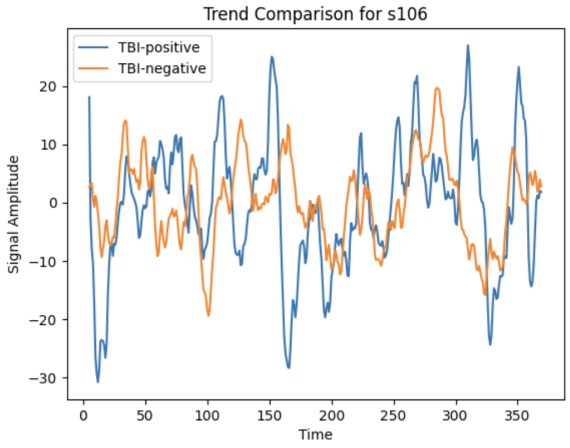
Observation: From the above heat map, if we observe the brain regions s157 and s155 we can see they both are having high correlation. This implies that these two brain regions are highly correlated which in turn indicates that they are part of the same functional neural network, and then it's likely that both are together affected by TBI. Furthermore, researchers can discover which brain regions show changed correlations in TBI by comparing the correlation matrix of TBI positive patients to that of healthy controls. These areas may be especially sensitive to TBI-induced injury.

3. We created a heat map to visualize the correlation matrix of TBI-Neg patients.



Observation: From the above heat map, if we observe the brain regions s248 and s250 we can see they both are having high correlation. This implies that these two brain regions are highly correlated which in turn indicates that they are part of the same functional neural network. Furthermore, researchers can discover these kinds of brain regions which have high and low correlations and compare it with TBI affected patients. These kinds of areas may be highly helpful in analyzing the TBI negative cases with affected patients.

4. For our understanding, we took a few common brain regions of both TBI-Pos and TBI-Neg patients and checked how the time series data is varying between them at a few time steps.



Observation: In the above plot we can see how the time series data for a brain region is varying at different time points for TBI-Pos and TBI-Neg patients. These fluctuations indicated a difference in the strength of signal between TBI-Pos and TBI-Neg case.

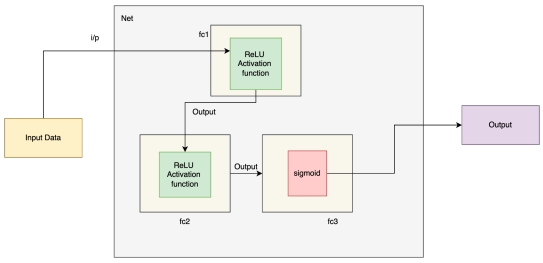
3 Summary of Machine Learning Models:

While performing EDA for TBI-Pos and TBI-Neg patients data, we understood that models which we build should be able to classify complex patterns in time series data from different brain regions. So we thought a fully connected neural network model, LSTM Model and matrix profiling with random forest modeling can act as classifiers to better assist us in predicting TBI pos or TBI neg data.

3.1 Fully Connected Neural Network Model:

We choose a fully connected neural network because it can learn the complex non-linear correlations between the output and input variables and is utilized in classification tasks. These can understand the raw input data and are also highly adaptable to new circumstances and data sets.

In the below figure, we have explained the workflow of our neural network model. We have trained the model using the entropy function and optimizer.



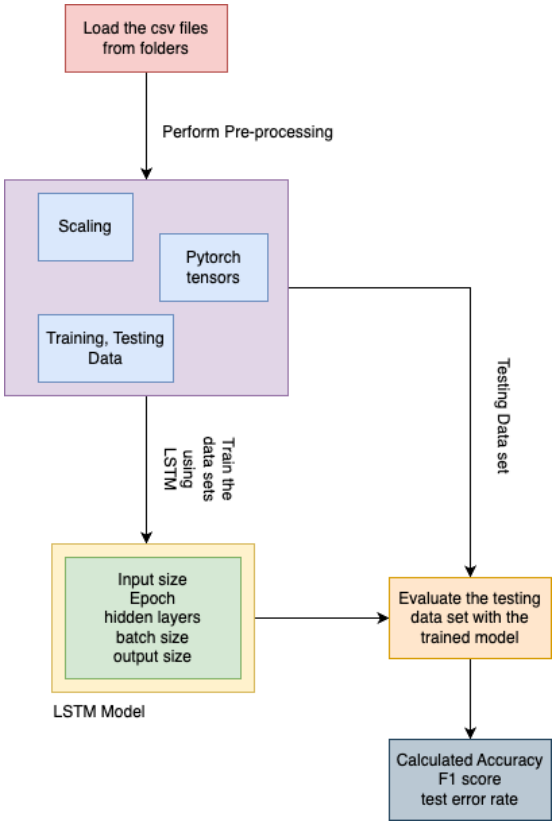
In the above figure: fc1, fc2, and fc3 are three fully connected layers in our model. ReLU and sigmoid are the activation functions we have used for our output and hidden layers. Our input tensor is passed through the first fc1 layer, and the ReLU activation function is applied. Next, the output of this is passed to the fc2 layer, and ReLU has applied again, and then the output of fc2 is passed to the fc3 layer, and then the sigmoid function is applied to it. This returns the output tensor.

3.2 LSTM Model for Classification of TBI-Pos and TBI-Neg

LSTM models help in identifying the complex patterns and trends in the data. They are efficient when compared with other algorithms and can work with different kinds of input and output data. They can work with incomplete or noisy data as well. So, we thought this LSTM can be used for classifying the time series data between the TBI-Positive and TBI-Negative patients.

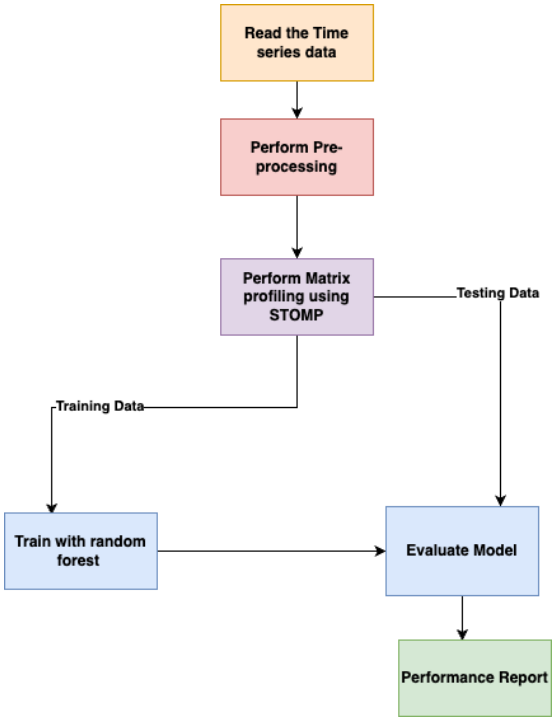
Below is the picture of a simple flow diagram of how our LSTM model works. After loading the patients data we performed pre-processing where scaling of data and conversion into PyTorch tensors for training data takes place. This trained data is fed into our LSTM and trained model is evaluated with

the testing data for calculating the performance of the model and checking whether the model is accurate for predicting the time series between the TBI-Pos and TBI- Neg patients.



3.3 Matrix profiling and Random Forest Modelling for Classification of TBI-Pos and TBI-Neg

For the next model we have chosen to perform the matrix profiling and use random forest for classification. Matrix profiling can be used to identify patterns and abnormalities in time series data from various brain areas. This can aid in the identification of brain activity correlations and patterns, which can be useful in the study of brain function and neurological illnesses. However, it may not provide a complete answer to a problem. Random forests, on the other hand, is a machine learning technique that can handle complex data sets and non-linear connections between variables. So, the patterns and abnormalities detected by matrix profiling can be used as helpful features in the random forest model, boosting its performance. Using these techniques together can be a strong approach for time series analysis, revealing both specific characteristics and overall trends in the data.



So, we have applied matrix profiling on the time series data of both TB-Pos and TBI-Neg patients and then used a random forest model for the classification of TBI-Pos and TBI-Neg patients. Above is the pictorial representation of our model.

4 Results

TBI-Pos folder has 20 person's data and TBI-Neg folder has 16 person's data. We have a total 36 patients data combining TBI-Pos and TBI-Neg. On dividing them to 80 and 20, we have 28 patients data for training and 8 patients data for testing purpose. Below are the results of our training and testing data set after pre-processing, training and evaluating them on the model.

On dividing them into training and testing below are the results for the same:

```
training data set: 28
testing data set: 8
```

4.1 Training and Evaluation of LSTM Model

For the LSTM model, we have divided our data into training and testing sets on a ratio of 80 and 20. Our LSTM model takes input as a sequence of data with input size as dimensions. We are sending each csv file which is each person's data as a single input to our model. Output of LSTM is generated through passing the output of LSTM cells at each time step through a fully connected layer which classifies the data as 0 or 1 for TBI-Neg and TBI-Pos data respectively.

(a) On training the LSTM model on 10 epochs, evaluating the model with testing data set below are the results of our model performance.

```
Epoch [1/10], Train Loss: 0.6948
Epoch [2/10], Train Loss: 0.5603
Epoch [3/10], Train Loss: 0.2289
Epoch [4/10], Train Loss: 0.0746
Epoch [5/10], Train Loss: 0.0207
Epoch [6/10], Train Loss: 0.0111
Epoch [7/10], Train Loss: 0.0072
Epoch [8/10], Train Loss: 0.0053
Epoch [9/10], Train Loss: 0.0041
Epoch [10/10], Train Loss: 0.0033
```

On evaluating the trained LSTM Model with testing data set:

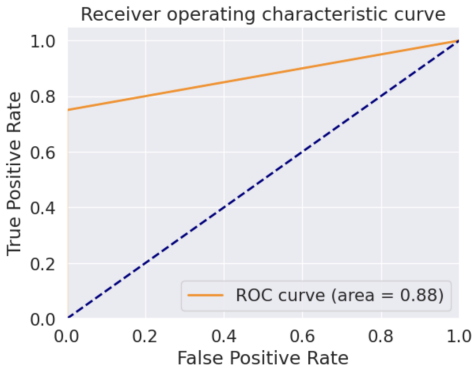
```
Test Loss: 0.4985, Test Acc: 87.50%
F1 Score: 0.8571
Confusion Matrix:
[[4 0]
 [1 3]]
```

(b) We have calculated test error rate, F1 score for our model, which helps us in identifying the performance of the model on time series data. Test error rate metric provides the information on how well the model is able to predict the data and F1 score helps in performing the evaluation of binary classification model.

Test error rate for our LSTM model is : 0.125

```
test error rate is 0.125
```

(c) We plot the ROC curve to evaluate the performance of binary classification done by the LSTM model: In the below plot, the X-axis represents the false positive rate which indicates the number of patients who are TBI-Negative incorrectly classified as TBI-Positive. Y-axis represents the True positive rate which indicates the number of positive patients correctly classified as TBI-Positive.



By comparing the true positive rate (TPR) and false positive rate (FPR) at each threshold value, the curve is drawn. The area under the curve (AUC) is a measure of the model's overall performance. Higher AUC as well indicates better performance.

A classification model with low test error rate and high F1 score indicates that it is able to classify data accurately. For our LSTM we can see low test error rate and high F1 score with which we assumed that our model is fitting the data.

4.2 Training and Evaluation of Fully Connected Neural Network Model

We have loaded the data set and implemented the neural network for our classification using PyTorch. We split the data into test and training data on 80, 20 ratio. We converted the data into PyTorch data loader objects with batch sizes.

(a) We trained the model, which iterated over our training data set multiple times and after every epoch our model parameters were updated, and it was evaluated with testing data for performance checking. We obtained an accuracy of 60 percent for the model performed:

```
Epoch 1, loss: 0.686
Epoch 2, loss: 0.671
Epoch 3, loss: 0.667
Epoch 4, loss: 0.665
Epoch 5, loss: 0.665
Epoch 6, loss: 0.662
Epoch 7, loss: 0.666
Epoch 8, loss: 0.660
Epoch 9, loss: 0.661
Epoch 10, loss: 0.659
Epoch 11, loss: 0.660
Epoch 12, loss: 0.662
Epoch 13, loss: 0.661
Epoch 14, loss: 0.660
Epoch 15, loss: 0.660
Epoch 16, loss: 0.660
Epoch 17, loss: 0.662
Epoch 18, loss: 0.659
Epoch 19, loss: 0.659
Epoch 20, loss: 0.659
Epoch 21, loss: 0.659
Epoch 22, loss: 0.658
Epoch 23, loss: 0.659
Epoch 24, loss: 0.658
Epoch 25, loss: 0.661
Epoch 26, loss: 0.658
Epoch 27, loss: 0.661
Epoch 28, loss: 0.658
Epoch 29, loss: 0.659
Epoch 30, loss: 0.660
Epoch 31, loss: 0.659
Epoch 32, loss: 0.660
Epoch 33, loss: 0.659
Epoch 34, loss: 0.660
Epoch 35, loss: 0.657
```

```
Epoch 92, loss: 0.657
Epoch 93, loss: 0.656
Epoch 94, loss: 0.658
Epoch 95, loss: 0.657
Epoch 96, loss: 0.658
Epoch 97, loss: 0.657
Epoch 98, loss: 0.656
Epoch 99, loss: 0.657
Epoch 100, loss: 0.658
Accuracy on the test set: 60.00%
```

(b) From the techniques discussed in the lecture: We calculated the precision, recall, and F1 score as well for our model performance evaluation.

```
Precision: 0.4555
Recall: 0.4935
F1 score: 0.4737
```

(c) We choose to use the Mean Squared Error rate technique as well to calculate the model's test error rate. The MSE (Mean Squared Error) is a measure that will quantify how well the model will perform on the test data set. We first used our model to make predictions on the testing data, and then we compared the actual values with our predicted values. Then we calculated the squared difference between the real values and predicted values and obtained the MSE value by taking the average of those squared differences. We calculated the Mean Squared Error and, absolute error, root mean squared error for our model to report the test error rate:

```
Test loss: 0.275
Test error rate: 0.460
Test mean squared error: 0.268
Test mean absolute error: 0.500
Test root mean squared error: 0.518
```

If there is a lower MSE, it indicates how well our model is suitable for working on new data. If there is a higher MSE, it may indicate that the model which we choose is not well-suitable for predicting data accurately. The test error rate we obtained from the above model indicates is moderately not high or not low. Based on these values, our model might fit the data but not too accurately. So we built a third model for our analysis.

4.3 Training and Evaluation of Matrix profiling and Random forest model

We have divided our data into training and testing data on 80, 20 scale ratio after doing the matrix profiling and then used this train and test data sets for predicting using random forest model.

(a) Below is the result of our F1 score, testing accuracy of our random forest model.

```
Accuracy: 0.6250
Recall: 0.6667
F1 Score: 0.7273
Confusion Matrix:
[[1 1]
 [2 4]]
Mean Squared Error: 0.3750
Test Error Rate: 0.3750
```

We got accuracy of 62 percent on performing matrix profiling, training the matrix profiling data with random forest and evaluating the trained model with the testing data set.

(b) We have calculated the F1 score and test error rate as well, which helps in identifying how well the model is able to predict and perform the binary classification on our data set.

```
Recall: 0.6667
F1 Score: 0.7273
Mean Squared Error: 0.3750
Test Error Rate: 0.3750
```

Observation: Our test error rate is 0.37 and F1 score is 0.72, our F1 score is high but the test error rate is not relatively very low.

(c) To Check how our model has predicted, we have compared the actual and predicted values for few of the testing data.

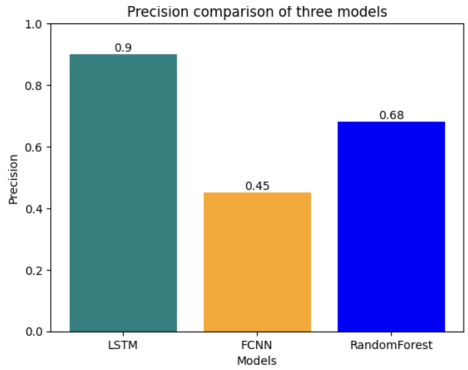
	Actual	Predicted
0	0.0	0.0
1	0.0	1.0
2	1.0	0.0
3	1.0	0.0
4	1.0	1.0
5	1.0	1.0

We have low test error rate of 0.37 and good F1 score with accuracy of 62 percent which indicates that this model moderately fits the data.

5 Comparison of Machine learning Models

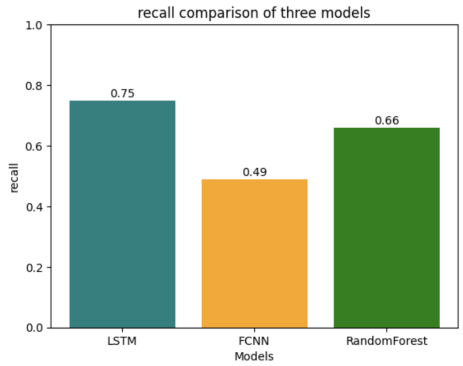
To check which model fits our data, we created the bar plots to compare F1 score, recall, test error rate and precision score of all the three models. Below are the plots for f1 Score, recall, precision, test error rate and accuracy of the three models.

(a)



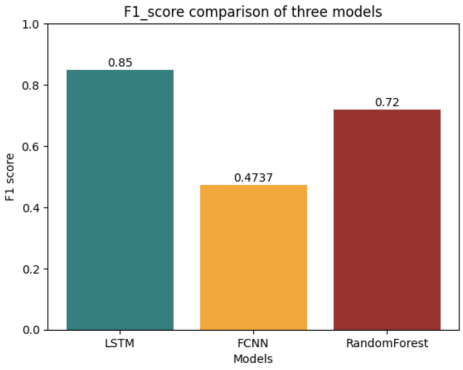
Observation: In the above plot it has models on x-axis and the model precision values which we got from classification report of the three models on y-axis. We can see that precision value for LSTM model is high among all the three models.

(b)



Observation: In the above plot it has models on x-axis and the model Recall values for, which we got from classification report of the three models on y-axis. We can see that the recall value for LSTM model is high among all the three models.

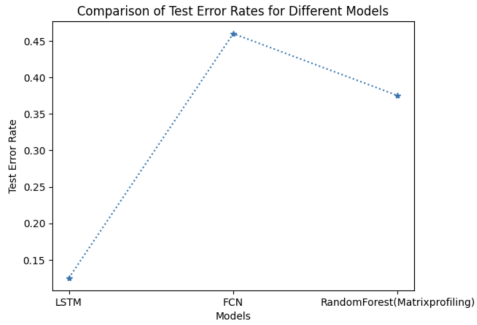
(c)



Observation: In the above plot it has models on x-axis and the model F1 Score on y-axis which we got from a classification report of the three models. We can see that the F1 score

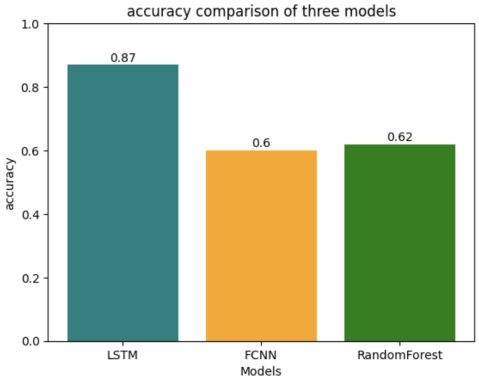
for LSTM model is high among all the three models.

(d)



Observation: In the above plot it has model labels on x-axis and the test error rate of the three models on y-axis. We can see that the test error rate value for the LSTM model is very low among all the three models.

(e)



Observation: In the above plot it has model labels on x-axis and the accuracies of the three models on y-axis. We can see that the accuracy rate value for the LSTM model is very high among all the three models.

Out of all the models which we implemented for classifying the TBI status of the patients, we have observed that the LSTM model gave us a high F1 score of 0.85 and low-test error rate of 0.125 with 87 percent accuracy. Hence, it performs the best in all the cases and gives the most accurate result for our data.

6 Predictions

As this is the case of medical diagnosis, accuracy cannot be the only metric to evaluate. In medical diagnosis, it is essential to predict the actual values correctly. We cannot incorrectly diagnose a patient as TBI-Neg after the accurate diagnosis report shows that patient has Traumatic Brain Injury. So along with higher accuracy, we need to check other performance metrics as well like higher F1 score, recall, test error rate. Comparatively, the LSTM model gives 87 percent accuracy and a high F1 score of 0.857 on test and train data. Hence, for our specific data set LSTM model appears to be a better match.

Below are the three predictions we made for LSTM model:

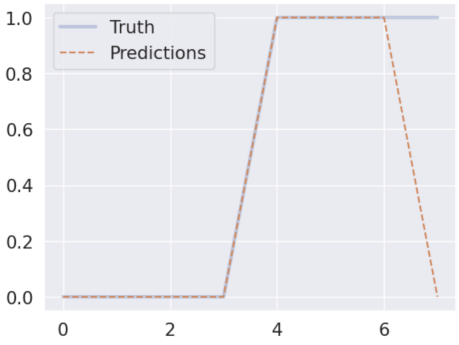
(a) We have calculated the Testing accuracy for our model. Our model acquired 75 percent accuracy which indicated us that the model can be used for prediction of TBI status of a patient. We calculated the other performance metrics as well like F1 score, recall, precision as well. Below pictures show the results of these metric values for our model.

```
Test Loss: 0.4985, Test Acc: 87.50%

              precision    recall  f1-score   support

         0.0       0.80      1.00      0.89         4
         1.0       1.00      0.75      0.86         4

    accuracy                           0.88         8
   macro avg       0.90      0.88      0.87         8
weighted avg       0.90      0.88      0.87         8
```

(b) To Check how our model has predicted, we have compared the actual and predicted values: And observed most of the actual and predicted values are the same. We also plotted the actual and predicted variables for visualization of testing data.

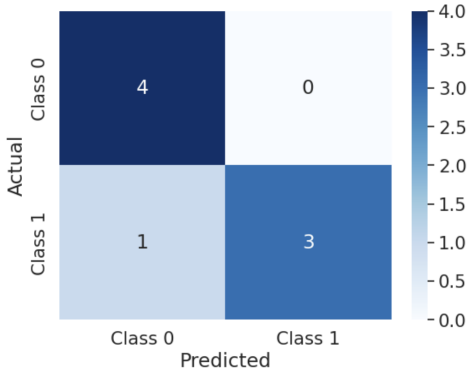
	Actual	Predicted
0	0.0	[0.0]
1	0.0	[0.0]
2	0.0	[0.0]
3	0.0	[0.0]
4	1.0	[1.0]
5	1.0	[1.0]
6	1.0	[1.0]
7	1.0	[0.0]

We have pictorial representation of these data below:



Observation: The comparison between the true and anticipated variables is shown in the plot. Orange dotted line represents the predicted values and the blue line represents the truth values. X- axis represents the index of each data point and y-axis represents the actual value of data.

(c) We created a confusion matrix to see the visual display of the performance of the model. This matrix helped us to know how the model classifies the data and if there are any steps we have to perform to improvise the model. Our confusion matrix showed that values that are being predicted correctly and are high for both TBI Pos and TBI Neg.



Observation: From the confusion matrix, we can infer that our model has correctly predicted four samples as TBI Neg, whereas the actual class is also TBI Neg. And for TBI Pos as well, three samples are correctly classified as TBI Pos, where the actual class is also TBI Pos. Since our True Positive and True Negative rate is high, we thought by this analysis that our model fits the data.

(d) For our input data, it should predict whether its TBI pos or TBI neg based on the input values: We gave a TBI-neg patients (205-DOD file) data and checked to see which category the input values fall and we got TBI-Neg.

Observation: From this we infer that our model is predicting correctly.

7 Summary and Conclusion

7.1 Conclusion Based on Results of our Analysis:

We Worked on building a model that would help in classifying a TBI-Pos person from a TBI-Neg person which would highly benefit the doctors, researchers in implementing the medicines, treatments and understanding patients characteristics. Throughout the project, we have worked on understanding different ways of performing preprocessing for a data set, multiple ways to visualize data, plotting ROC curve, heat maps, confusion matrix, time series plots as well. And also by working on various deep learning models and techniques that would reduce the noise, anomalies, we understood how to work with complex patterns in time series data. From all the models we have implemented we have achieved accuracy of 87 percent for the LSTM model which is relatively high. This can be used as a classifier in prediction analysis of TBI-Status as it has high results in other performance metrics as well like F1 score, precision, recall and test error rate as well. Through this model we can identify if a person has TBI pos or not and can also use this as analysis in prediction of Alzheimer's.

7.2 Domain Experts Inference from the Results:

Prediction of patients' TBI status helps in identifying the underlying patterns and detect the risk of injury severity over time. Building a model that classifies the TBI status of patients would help in the implementation of medicine for the cure of TBI, which leads to a decrease in the impact of TBI on a patient and its long-term effects. From our project, domain experts, namely researchers, doctors, and clinicians, can evaluate the TBI from a Non-TBI person's characteristics and differentiate between the TBI-Pos and TBI-Neg patients based on their time series data of different functional brain regions. This in turn leads them to improvise the

treatments, perform advance research resulting in improvement of patient outcomes.

7.3 Future Work and Conclusion:

If we had more time, we would have gathered more data to predict the TBI status of a patient and worked on more neural network models to analyze the models that help classify TBI patients from TBI-Neg patients. We will have gathered a data set with more relevant features like age and physical symptoms if the patient has a headache or dizziness problem. As per our current data, we have 36 patients' results. We aim to collect more patient data of both TBI-Pos and TBI-Neg, which would give us more training and testing samples that would help in precise accuracy and performance metrics.

And also, we would have worked on other factors that lead to Alzheimer's like mild cognitive impairment patients data, alcohol use disorder patients data, subjective cognitive decline patients data. Since people with any of these factors might be at higher risk of acquiring Alzheimer's along with TBI.