Detection of Alzheimer's using Machine Learning Models

Thota Sahaja
CPSC 6300
Applied Data Science
Second semester
tsahaja@clemson.edu
Data Cleaning, Gaussian smoothing
Long Short-Term Memory

Kota Jagathi
CPSC 6300
Applied Data Science
Second semester
kjagath@clemson.edu
Data visualization Heatmap
Support Vector Machine

Sai Likhitha Pacha
CPSC 6300
Applied Data Science
Third semester
spacha@clemson.edu
Data visualization Correlation Network
Convolution Neural Networks

ABSTRACT

A dreadful brain condition that typically affects the elderly is Alzheimer's disease. It is a condition that is both undertreated and underrecognized and is quickly becoming to be a significant public health issue. Improved clinical diagnostic guidelines and the treatment of behavioral issues as well as cognitive impairment are recent accomplishments. The most typical reason for dementia is Alzheimer's disease (AD). It is a neurodegenerative illness having a pathogenic origin. In the field of biomedical science, from the delivery of medicines to medicinal visioning, machine learning algorithms, in particular analytical modeling and sample detection, have come to be as one of the key approaches that are assisting researchers in developing a deeper understanding of the overall problem and resolving difficult clinical issues. The most popular machine learning method for identifying and retrieving the features is deep learning. In this paper we are trying to detect whether a patient has Traumatic Brain Injury (TBI) or not based on their data using Support Vector Machine, Long Short-Term Memory (LSTM) and Convolutional Neural Network (CNN).

1. Introduction

Alzheimer's disease (AD) is a serious, developing, and irreversible brain disorder that shows many different signs and symptoms. It gradually kills brain cells, which has an impact on memory, cognition, and eventually the capacity to do even the most fundamental functions. Lastly, the effects of this illness on cognitive decline result in dementia. One in 85 persons have been diagnosed with Alzheimer's disease.

According to current theories, AD is brought on by an overproduction of proteins that obstruct brain transmission. The actual process of the cause, however, is unclear, and the accumulation may start long before symptoms do. Cognitively normal (CN), early mild cognitive impairment (EMCI), substantial memory concern (SMC), and late mild cognitive impairment (LMCI) are the distinct phases of this illness based on cognitive function. CN stands for healthy people who don't appear to have any symptoms. When EMCI symptoms like as forgetting events and missing things begin to appear, persons often seek medical attention. SMC, the moderate stage of AD, is distinguished by diminished cognitive function and more frequent forgetfulness. The disease's serious stage, known as LMCI, is marked by profound cognitive impairment and need care.

1.1 What is the main question your project seeks to answer?

The main purpose of this study is to classify whether a patient has Traumatic Brain Injury (TBI) or not based on their data using Long Short-Term Memory (LSTM) and Convolutional Neural Network (CNN). The patient data is loaded from a CSV file, and a Support Vector Machine model that has been trained on previous data is used to make predictions on the new patient data. The result of the model, which determines whether the patient has TBI or not, and the output is either "The patient has TBI" or "The patient does not have TBI.

1.2 Provide a brief motivation for your project question. Why is this question important? What can we learn from your project?

Traumatic brain injury (TBI) is one of the most erratic damages caused to a specific section of the brain or a larger area of the brain. The patient must be foreseen at an early stage for the diagnosis to control the wide area spread. By using machine learning models, we can predict TBI and underscore its potential to enhance neuroimaging study diagnostic results. we utilized the ADNI data set and applied Exploratory data analysis (EDA), which involves handling missing values, gaussian smoothing, correlation matrix, and finally, a data visualization. Later, we presented a comparative analysis of LSTM, SVM, and CNN models for predicting Traumatic Brain Injury (TBI). And concluded that LSTM works well with the time series data.

1.3 Briefly describe the data source(s) you have used in your project. Where is the data from? How big is the data in terms of data points and/or file size? If the data was not already available, how did you collect the data?

In the current study, we utilized the ADNI data set, which is made up of two subgroups: TBI POSITIVE, which contains 20 patient csv files, and TBI NEGATIVE, which includes 16 patient csv files. Each csv files data resembles individual patient files since they have 375 specific time points and 294 distinct brain regions. We even considered one 1162.csv file from HC-all and demographics-all.xlsx for EDA. Our project guide Luyi Li provided this ADNI data in the google drive link. In terms of data points, each .csv file has 294 brain region columns and 375 specific time point rows. We can calculate the total number of data points by multiplying the number of columns by the number of rows. Total data points = a number of brain region columns × Number of time point rows Total data points = 294 columns × 375 rows, Total data points = 110,250 data points. Each .csv file size is 1.4MB. The ADNI data set is collected five steps:

1. Using a magnetic resonance imaging (MRI) scanner, they acquire resting-state functional MRI data of the patient's brain every 3 seconds.
2. Generate a 3D image of the patient's brain for each time point to indicate the pattern of brain activity during rest.
3. Later, Preprocess the acquired data to correct for artifacts, motion, and physiological noise.
4. Segment the preprocessed data into 294 brain regions using a parcellation scheme.
5. In this ADNI dataset, they repeated the above steps for 375-specific time points, resulting in a time series.

2. Summary of your EDA

2.1 What is the unit of analysis?

The individual entity or object being analyzed in our data set can be considered the unit of analysis. In our project, we performed whether a patient csv file is TBI or not, so the individual patient's .csv files are considered as the unit of analysis. Each column represents a specific time point, and each row represents a particular brain region in the dataset for each patient. Understanding the unit of analysis is crucial because it influences the selection of appropriate procedures and the ability to draw reliable conclusions from the data.

2.2 How many observations in total are in the data set?

The ADNI data set has two folders: TBI POSITIVE, which includes 20 patient csv files, and TBI NEGATIVE, which includes 16 patient csv files. Each patient csv file has 375 rows and 294 columns. The 1162.csv file from HC-all and a demographic xl sheet are considered in the dataset.

2.2 How many unique observations are in the data set?

The distinguishing characteristic or observation that can be helpful in this case is usually the existence or absence of a TBI. This variable may be a two-sided pointer, with a value of 1 addressing patients who have experienced a traumatic brain injury and a value of 0 addressing patients who have not.

2.4 What time period is covered?

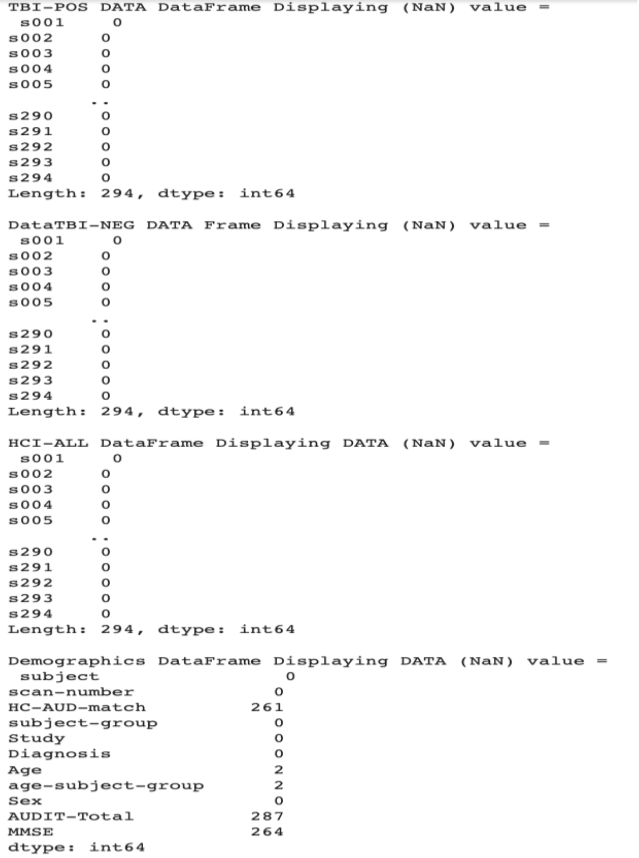
Dataset Information:

Time Period: 3 seconds per signal up to 375 specific points. Participants: 36 patients from TBI POSITIVE, TBI NEGATIVE.

2.5 Briefly summarize any data cleaning steps you have performed.

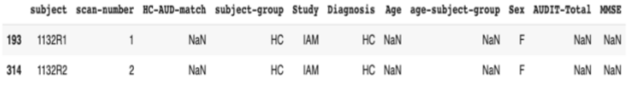
We used the Pandas module to import the following CSV and Excel files in order to do data cleaning: TBI-pos (103-DOD-tts all.csv), TBI-neg (213-DOD-tts all.csv), HC-all (1162 tts all.csv), and demographics-all.xlsx.

1. The first five rows were displayed for each dataset using the head() function.
2. In order to determine how many rows and columns were included in the datasets, we used the shape property to assess their shape.
3. The columns' individual data types were examined using the info() method.
4. The isnull() function was used to check for missing values in these datasets **[Figure1]**.
5. We only found that the demographics-all.xlsx was all. Specific columns in the xlsx dataset lack values.
6. The rows with a zero age and age-subject group were found and filtered **[Figure2]**.
7. Using the drop() method, we removed the rows with 0 values for age and age-subject-group.
8. All other NaN values have been changed to zero **[Figure3].**
9. We performed the necessary adjustments after checking to see if any data types needed to be modified. In particular, we changed the Age, HC-AUD-match, and AUDIT-TOTAL columns' data types to integer format. These important data cleaning activities were carried out to ensure the data was consistent, comprehensive, and prepared for exploratory data analysis.



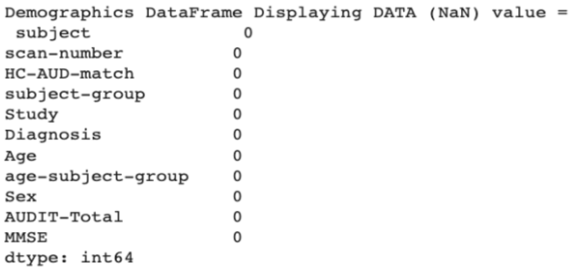
```
TBI-POS DATA DataFrame Displaying (NaN) value =
 s001     0
s002     0
s003     0
s004     0
s005     0
        ..
s290     0
s291     0
s292     0
s293     0
s294     0
Length: 294, dtype: int64

DataTBI-NEG DATA Frame Displaying (NaN) value =
 s001     0
s002     0
s003     0
s004     0
s005     0
        ..
s290     0
s291     0
s292     0
s293     0
s294     0
Length: 294, dtype: int64

HCI-ALL DataFrame Displaying DATA (NaN) value =
 s001     0
s002     0
s003     0
s004     0
s005     0
        ..
s290     0
s291     0
s292     0
s293     0
s294     0
Length: 294, dtype: int64

Demographics DataFrame Displaying DATA (NaN) value =
 subject                 0
scan-number             0
HC-AUD-match          261
subject-group           0
Study                   0
Diagnosis               0
Age                     2
age-subject-group       2
Sex                     0
AUDIT-Total           287
MMSE                  264
dtype: int64
```

Figure1: Checking Missing Values For TBI-POS, TBI-NEG, AND HC-ALL.



	subject	scan-number	HC-AUD-match	subject-group	Study	Diagnosis	Age	age-subject-group	Sex	AUDIT-Total	MMSE
193	1132R1	1	NaN	HC	IAM	HC	NaN	NaN	F	NaN	NaN
314	1132R2	2	NaN	HC	IAM	HC	NaN	NaN	F	NaN	NaN

Figure2: Age And Age-Subject Group Were Found To Be Zero.



```
Demographics DataFrame Displaying DATA (NaN) value =
 subject            0
scan-number         0
HC-AUD-match        0
subject-group       0
Study               0
Diagnosis           0
Age                 0
age-subject-group   0
Sex                 0
AUDIT-Total         0
MMSE                0
dtype: int64
```

Figure3: Replacing All Leftover NAN Values With Zero.

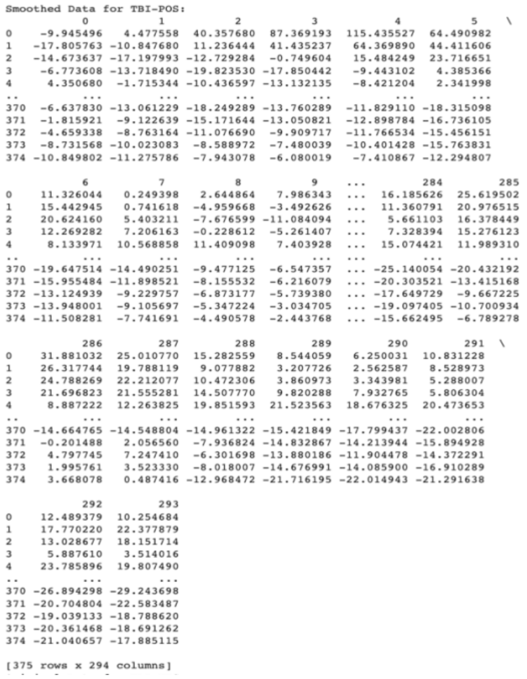
2.6 Visualization of the response with an appropriate technique

We applied Gaussian smoothing and correlation matrix for TBI-POS, TBI-NEG, and HC-ALL before visualizing in the dependent variable with the heatmap and correlation network.

1. **Gaussian Smoothing:** In time series data, gaussian smoothing reduces noise and brings out trends. A low-pass Gaussian filter with a standard deviation of 1.0 is applied to reduce noise and highlight patterns. Following applying Gaussian smoothing with a standard deviation of 1.0, the provided data displays the smoothed values for TBI-POS, TBI-NEG, and HC-ALL. The smoothed data represents the original time series data, which is a cleaner and more filtered version. The numbers in the smoothed data have been changed to emphasize long-term trends and reduce high-frequency noise. Each column in the given data correlates to a particular point in the time series, whereas each row in the data refers to a time step or observation. The values in the cells represent each data point's smoothed values. The values in the cells represent the smoothed data points for each time step and point within the time series. The data is anticipated to have less noise after applying Gaussian smoothing, making it simpler to spot long-term trends and patterns in the TBI-POS, TBI-NEG, and HC-ALL datasets **[Figure 4,5,6]**.
2. **Correlation Matrix:** A correlation matrix is a table that displays the correlation coefficients between variables in a dataset. It helps understand the relationships between variables and can help predict outcomes and select features. By calculating the correlation coefficients for the TBI-POS, TBI-NEG, and HC-ALL datasets, you can create correlation matrices that show the strength and direction of the relationships between variables.

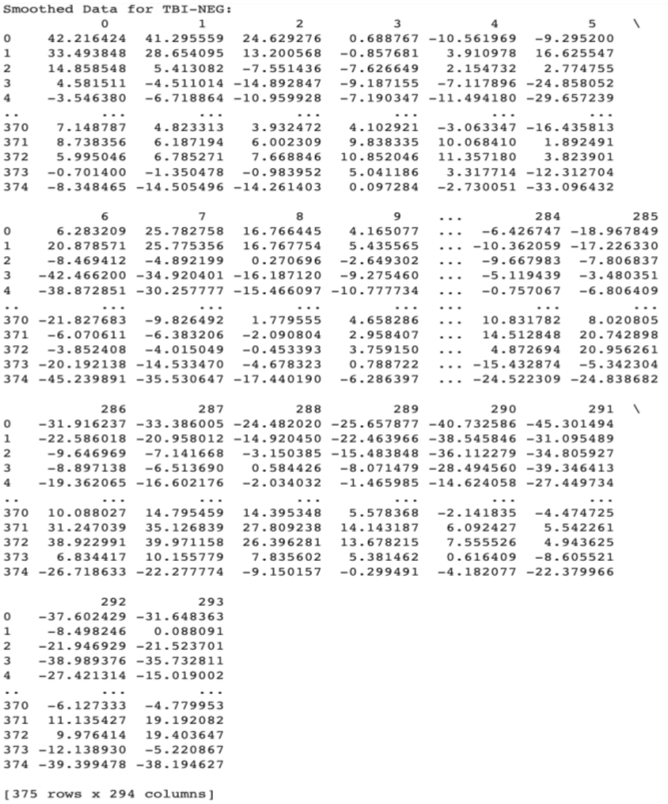
Visualization Techniques: In summary of EDA, we used heatmap, and correlation network.

1. **Heatmap:** A heatmap is a graphical representation of data that uses colors to represent different values in a matrix. In EDA, heatmaps can be used to visualize correlations between variables in a correlation matrix. Heatmaps are useful for identifying patterns and trends in large datasets, making it easier to spot areas of high correlation or low correlation between variables **[Figure 7,8,9]**.
2. **Correlation network:** A correlation network is a graphical representation of a correlation matrix that uses nodes (representing variables) and edges (representing correlations) to show the relationships between variables. Correlation networks are useful for identifying groups of variables that are highly correlated with each other, as well as identifying variables that are not strongly correlated with any other variable in the dataset. Correlation networks are often used in complex datasets where there are many variables and correlations to explore **[Figure 10,11,12]**.



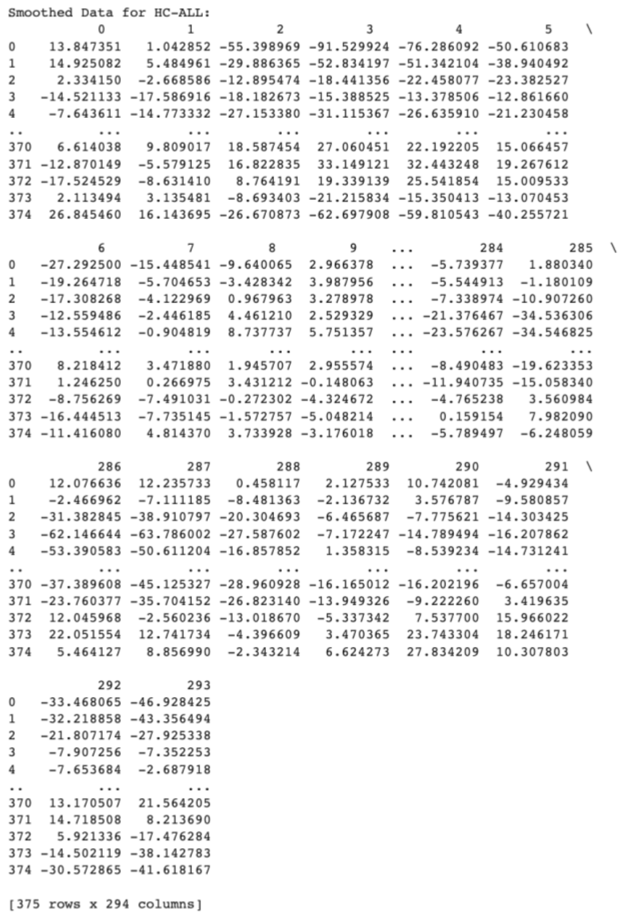
```
Smoothed Data for TBI-POS:
             0          1          2          3           4          5   \
0    -9.945496   4.477558  40.357680  87.369193  115.435527  64.490982
1   -17.805763 -10.847680  11.236444  41.435237   64.369890  44.411606
2   -14.673637 -17.197993 -12.729284  -0.749604   15.484249  23.716651
3    -6.773608 -13.718490 -19.823530 -17.850442   -9.443102   4.385366
4     4.350680  -1.715344 -10.436597 -13.132135   -8.421204   2.341998
..         ...        ...        ...        ...         ...        ...
370  -6.637830 -13.061229 -18.249289 -13.760289  -11.829110 -18.315098
371  -1.815921  -9.122639 -15.171644 -13.050821  -12.898784 -16.736105
372  -4.659338  -8.763164 -11.076690  -9.909717  -11.766534 -15.456151
373  -8.731568 -10.023083  -8.588972  -7.480039  -10.401428 -15.763831
374 -10.849802 -11.275786  -7.943078  -6.080019   -7.410867 -12.294807

             6          7          8          9  ...        284        285  \
0    11.326044   0.249398   2.644864   7.986343  ...  16.185626  25.619502
1    15.442945   0.741618  -4.959668  -3.492626  ...  11.360791  20.976515
2    20.624160   5.403211  -7.676599 -11.084094  ...   5.661103  16.378449
3    12.269282   7.206163  -0.228612  -5.261407  ...   7.328394  15.276123
4     8.133971  10.568858  11.409098   7.403928  ...  15.074421  11.989310
..         ...        ...        ...        ...  ...        ...        ...
370 -19.647514 -14.490251  -9.477125  -6.547357  ... -25.140054 -20.432192
371 -15.955484 -11.898521  -8.155532  -6.216079  ... -20.303521 -13.415168
372 -13.124939  -9.229757  -6.873177  -5.739380  ... -17.649729  -9.667225
373 -13.948001  -9.105697  -5.347224  -3.034705  ... -19.097405 -10.700934
374 -11.508281  -7.741691  -4.490578  -2.443768  ... -15.662495  -6.789278

           286        287        288        289        290        291  \
0    31.881032  25.010770  15.282559   8.544059   6.250031  10.831228
1    26.317744  19.788119   9.077882   3.207726   2.562587   8.528973
2    24.788269  22.212077  10.472306   3.860973   3.343981   5.288007
3    21.696823  21.555281  14.507770   9.820288   7.932765   5.806304
4     8.887222  12.263825  19.851593  21.523563  18.676325  20.473653
..         ...        ...        ...        ...        ...        ...
370 -14.664765 -14.548804 -14.961322 -15.421849 -17.799437 -22.002806
371  -0.201488   2.056560  -7.936824 -14.832867 -14.213944 -15.894928
372   4.797745   7.247410  -6.301698 -13.880186 -11.904478 -14.372291
373   1.995761   3.523330  -8.018007 -14.676991 -14.085900 -16.910289
374   3.668078   0.487416 -12.968472 -21.716195 -22.014943 -21.291638

           292        293
0    12.489379  10.254684
1    17.770220  22.377879
2    13.028677  18.151714
3     5.887610   3.514016
4    23.785896  19.807490
..         ...        ...
370 -26.894298 -29.243698
371 -20.704804 -22.583487
372 -19.039133 -18.788620
373 -20.361468 -18.691262
374 -21.040657 -17.885115

[375 rows x 294 columns]
```

Figur4: Smoothed Data For TBI-POS.



```
Smoothed Data for TBI-NEG:
             0          1          2          3          4          5   \
0    42.216424  41.295559  24.629276   0.688767 -10.561969  -9.295200
1    33.493848  28.654095  13.200568  -0.857681   3.910978  16.625547
2    14.858548   5.413082  -7.551436  -7.626649   2.154732   2.774755
3     4.581511  -4.511014 -14.892847  -9.187155  -7.117896 -24.858052
4    -3.546380  -6.718864 -10.959928  -7.190347 -11.494180 -29.657239
..         ...        ...        ...        ...        ...        ...
370   7.148787   4.823313   3.932472   4.102921  -3.063347 -16.435813
371   8.738356   6.187194   6.002309   9.838335  10.068410   1.892491
372   5.995046   6.785271   7.668846  10.852046  11.357180   3.823901
373  -0.701400  -1.350478  -0.983952   5.041186   3.317714 -12.312704
374  -8.348465 -14.505496 -14.261403   0.097284  -2.730051 -33.096432

             6          7          8          9  ...        284        285
0     6.283209  25.782758  16.766445   4.165077  ...  -6.426747 -18.967849
1    20.878571  25.775356  16.767754   5.435565  ... -10.362059 -17.226330
2    -8.469412  -4.892199   0.270696  -2.649302  ...  -9.667983  -7.806837
3   -42.466200 -34.920401 -16.187120  -9.275460  ...  -5.119439  -3.480351
4   -38.872851 -30.257777 -15.466097 -10.777734  ...  -0.757067  -6.806409
..         ...        ...        ...        ...  ...        ...        ...
370 -21.827683  -9.826492   1.779555   4.658286  ...  10.831782   8.020805
371  -6.070611  -6.383206  -2.090804   2.958407  ...  14.512848  20.742898
372  -3.852408  -4.015049  -0.453393   3.759150  ...   4.872694  20.956261
373 -20.192138 -14.533470  -4.678323   0.788722  ... -15.432874  -5.342304
374 -45.239891 -35.530647 -17.440190  -6.286397  ... -24.522309 -24.838682

           286        287        288        289        290        291  \
0   -31.916237 -33.386005 -24.482020 -25.657877 -40.732586 -45.301494
1   -22.586018 -20.958012 -14.920450 -22.463966 -38.545846 -31.095489
2    -9.646969  -7.141668  -3.150385 -15.483848 -36.112279 -34.805927
3    -8.897138  -6.513690   0.584426  -8.071479 -28.494560 -39.346413
4   -19.362065 -16.602176  -2.034032  -1.465985 -14.624058 -27.449734
..         ...        ...        ...        ...        ...        ...
370  10.088027  14.795459  14.395348   5.578368  -2.141835  -4.474725
371  31.247039  35.126839  27.809238  14.143187   6.092427   5.542261
372  38.922991  39.971158  26.396281  13.678215   7.555526   4.943625
373   6.834417  10.155779   7.835602   5.381462   0.616409  -8.605521
374 -26.718633 -22.277774  -9.150157  -0.299491  -4.182077 -22.379966

           292        293
0   -37.602429 -31.648363
1    -8.498246   0.088091
2   -21.946929 -21.523701
3   -38.989376 -35.732811
4   -27.421314 -15.019002
..         ...        ...
370  -6.127333  -4.779953
371  11.135427  19.192082
372   9.976414  19.403647
373 -12.138930  -5.220867
374 -39.399478 -38.194627

[375 rows x 294 columns]
```

Figure5: Smoothed Data For TBI-NEG.



```
Smoothed Data for HC-ALL:
             0          1          2          3          4          5  \
0    13.847351   1.042852 -55.398969 -91.529924 -76.286092 -50.610683
1    14.925082   5.484961 -29.886365 -52.834197 -51.342104 -38.940492
2     2.334150  -2.668586 -12.895474 -18.441356 -22.458077 -23.382527
3   -14.521133 -17.586916 -18.182673 -15.388525 -13.378506 -12.861660
4    -7.643611 -14.773332 -27.153380 -31.115367 -26.635910 -21.230458
..         ...        ...        ...        ...        ...        ...
370   6.614038   9.809017  18.587454  27.060451  22.192205  15.066457
371 -12.870149  -5.579125  16.822835  33.149121  32.443248  19.267612
372 -17.524529  -8.631410   8.764191  19.339139  25.541854  15.009533
373   2.113494   3.135481  -8.693403 -21.215834 -15.350413 -13.070453
374  26.845460  16.143695 -26.670873 -62.697908 -59.810543 -40.255721

             6          7         8         9  ...        284        285  \
0   -27.292500 -15.448541 -9.640065  2.966378  ...  -5.739377   1.880340
1   -19.264718  -5.704653 -3.428342  3.987956  ...  -5.544913  -1.180109
2   -17.308268  -4.122969  0.967963  3.278978  ...  -7.338974 -10.907260
3   -12.559486  -2.446185  4.461210  2.529329  ... -21.376467 -34.536306
4   -13.554612  -0.904819  8.737737  5.751357  ... -23.576267 -34.546825
..         ...        ...       ...       ...  ...        ...        ...
370   8.218412   3.471880  1.945707  2.955574  ...  -8.490483 -19.623353
371   1.246250   0.266975  3.431212 -0.148063  ... -11.940735 -15.058340
372  -8.756269  -7.491031 -0.272302 -4.324672  ...  -4.765238   3.560984
373 -16.444513  -7.735145 -1.572757 -5.048214  ...   0.159154   7.982090
374 -11.416080   4.814370  3.733928 -3.176018  ...  -5.789497  -6.248059

           286        287        288        289        290        291  \
0    12.076636  12.235733   0.458117   2.127533  10.742081  -4.929434
1    -2.466962  -7.111185  -8.481363  -2.136732   3.576787  -9.580857
2   -31.382845 -38.910797 -20.304693  -6.465687  -7.775621 -14.303425
3   -62.146644 -63.786002 -27.587602  -7.172247 -14.789494 -16.207862
4   -53.390583 -50.611204 -16.857852   1.358315  -8.539234 -14.731241
..         ...        ...        ...        ...        ...        ...
370 -37.389608 -45.125327 -28.960928 -16.165012 -16.202196  -6.657004
371 -23.760377 -35.704152 -26.823140 -13.949326  -9.222260   3.419635
372  12.045968  -2.560236 -13.018670  -5.337342   7.537700  15.966022
373  22.051554  12.741734  -4.396609   3.470365  23.743304  18.246171
374   5.464127   8.856990  -2.343214   6.624273  27.834209  10.307803

           292        293
0   -33.468065 -46.928425
1   -32.218858 -43.356494
2   -21.807174 -27.925338
3    -7.907256  -7.352253
4    -7.653684  -2.687918
..         ...        ...
370  13.170507  21.564205
371  14.718508   8.213690
372   5.921336 -17.476284
373 -14.502119 -38.142783
374 -30.572865 -41.618167

[375 rows x 294 columns]
```

Figure6: Smoothed Data For HC-ALL.

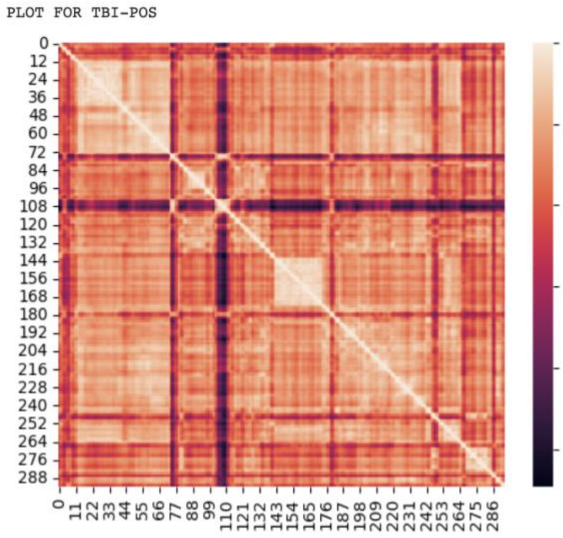


Figure7: Heatmap For TBI-POS

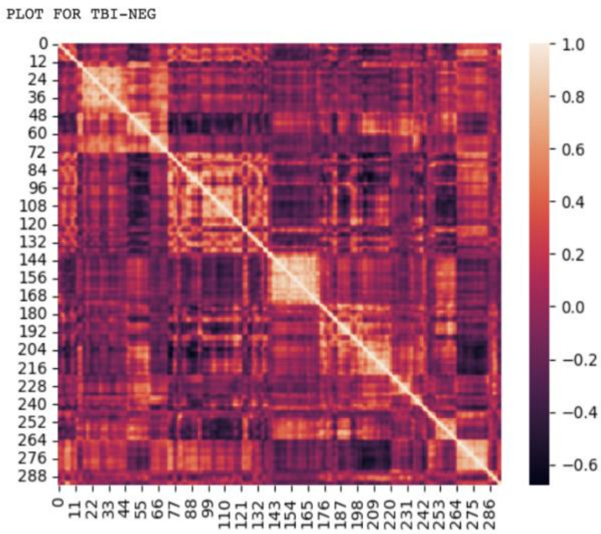


Figure8: Heatmap For TBI-NEG

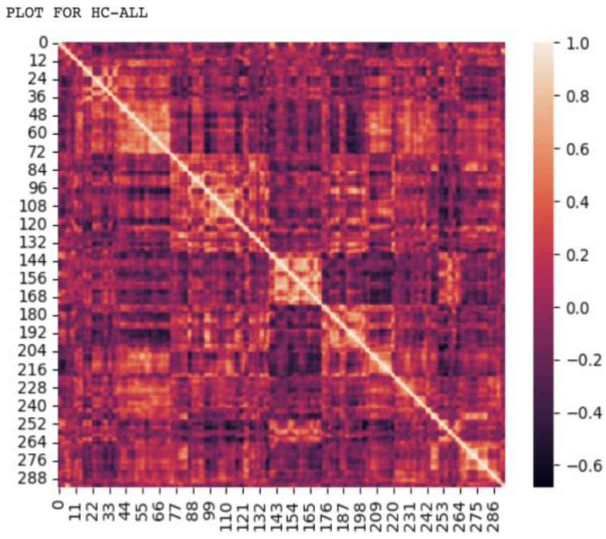


Figure9: Heatmap For HC-ALL

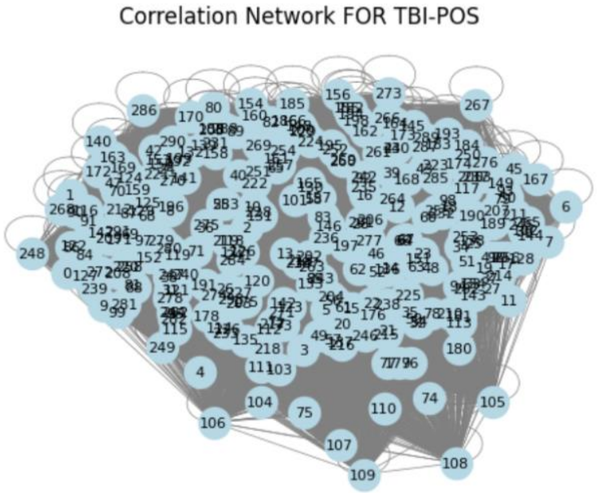


Figure10: Correlation Network For TBI-POSITIVE

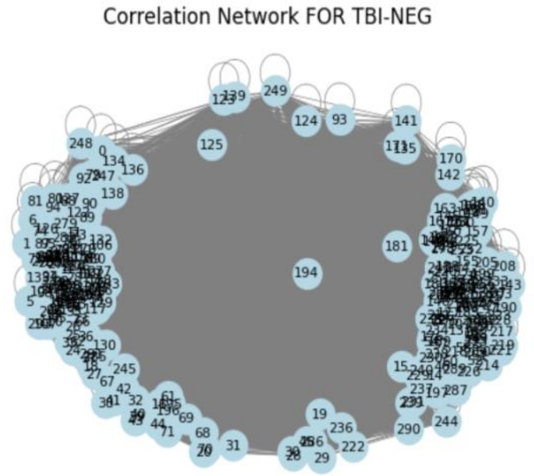


Figure11: Correlation Network For TBI-NEGATIVE

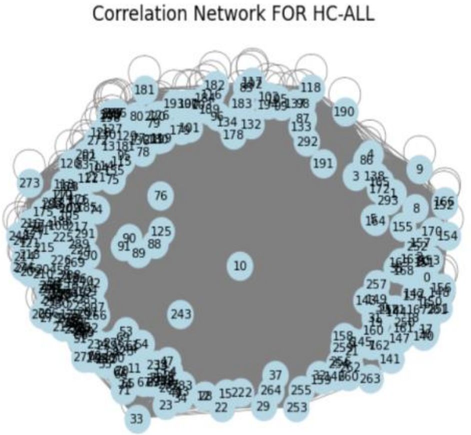


Figure12: Correlation Network For HC-ALL

2.7 Visualization of key predictors against the response. Pick one or two predictors that you think are going to be most important in explaining the response. Your selection of predictors can either be guided by your domain knowledge or be the result of your EDA on all predictors.

Using a heatmap, we represent the relationship the response variable and other variables in a tabular form using colors to indicate the intensity of the relationship. current heatmap displays different colors representing different levels of correlation. According to the heat maps created for TBI-positive and TBI-negative subjects, the TBI-positive heatmap shows a stronger positive correlation between TBI-positive and the likelihood of TBI disease. In contrast, the TBI-negative heatmap does not show the same level of intensity in terms of correlation. The brighter spots are key predictors because they demonstrate that they are heavily correlated to each other even when they are the same. The diagonal line is white because they are the same value, but everything around the white diagonal line is a key predictor. Similarly, a correlation network could be generated to visualize the relationships between variables in the TBI-positive, HC-ALL and TBI-negative datasets. The nodes in the network would represent variables, and the edges would represent the strength and direction of the correlations between them. For the TBI-positive dataset, the nodes could include brain regions. The edges between the nodes would be weighted based on the strength of the correlations between them, with brighter and thicker edges indicating stronger correlations. The resulting correlation network would help identify key predictors of TBI disease in the TBI-positive dataset, as well as the strength and direction of their relationships with other variables in the dataset. It would also help identify any groups of variables that are highly correlated with each other, which could be useful in developing predictive models for TBI disease. Upon performing these two data visualization techniques we can see that this is a large data pool to work with. I took the approach of finding the correlation matrix and finding the highest absolute correlation to be the most viable. In other words, the feature with the highest absolute correlation is my key predictor. In this case the feature with 0.939 correlation value is my key feature for TBI-POS similarly for TBI-NEG 0.932 is my key feature and 0.935 for HC-ALL.

3. Summary of Machine Learning Models

The main purpose of this study is to classify whether a patient has Traumatic Brain Injury (TBI) or not based on their data. We have successfully implemented one machine learning model and two deep learning techniques. We have used Support vector machine, Convolution neural networks and Long short-term memory (LSTM).

3.1 Justify your model choices based on how your response is measured and any observations you have made in your EDA.

Support Vector Machine:

A Support Vector Machine model (SVM) is a supervised machine learning approach for classification and regression analysis. It is a robust and adaptable model frequently used in many applications. It is an efficient and reliable algorithm for noisy, high-dimensional, massive datasets. Additionally, it is relatively simple to read and performs well in terms of generalization. Therefore, we consider the Support Vector Machine (SVM) model, which can be a good choice for detecting Alzheimer's disease. In addition, Support Vector Machine can handle both linear and nonlinear data. SVM improves generalization performance and prevents overfitting by maximizing the margin. SVM can deal with datasets with many characteristics, including datasets with many dimensions. For these reasons, we have considered SVM to be a good fit for our problem.

Convolution Neural Networks:

Convolutional Neural Networks intend to learn spatial feature hierarchies automatically and adaptively from unprocessed input data, eliminating the need for manual feature engineering. They accomplish this by first applying filters to the input data using convolutional layers, then reducing the spatial size of the data and boosting its robustness to changes with pooling layers. They perform incredibly well when the input data has a grid-like pattern, and the task involves image or signal processing. CNNs have gained popularity for medical image analysis tasks, including tumor detection and disease marker identification, due to their capacity to learn pertinent features from big and complicated datasets. For the above reasons, we have opted CNN to solve our problem.

LSTM:

A Long short- Term Memory (LSTM) Networks constitute a form type recurrent neural network architecture created to address the issues that standard RNNs may experience with vanishing and expanding gradients during backpropagation. According to the input and previous state, LSTM networks employ a memory cell that can sustain information over time as well as deliberately forget or remember information. The neural network design known as LSTM, which has been effectively used in several domains, including the analysis of medical datasets, may capture long-term dependencies. It is a common option for time-series analysis at medical research due to its efficacy in modeling sequential data. Therefore, we consider the Long short- Term Memory model, which can be a good choice for detecting Alzheimer's disease. For these reasons, we selected LSTM for our problem.

3.2 Report the results from at least two different models: For each model, report the model's test error. Justify your choice. For each model, discuss how well the model fits the data.

In order to justify which model is working well, we have considered several metrics. They are:

F1-Score: - It sums up the predictive performance of a model by combining two otherwise competing metrics — precision and recall **[Figure13,18,24]**.

Precision: - Precision is one indicator of the quality of a positive prediction made by the model. Precision refers to the number of true positives divided by the total number of positive predictions **[Figure 13,18,24]**.

Recall: - Recall is a metric that quantifies the number of correct positive predictions made from all positive predictions that could have been made. Unlike precision that only comments on the correct positive predictions out of all positive predictions, recall provides an indication of missed positive predictions **[Figure 13,18,24]**.

Confusion matrix: - It helps in identifying which classes are easy to predict and which are hard to predict. It provides how many examples for each class are correctly classified and how many are confused with other classes. Each row of the matrix represents the instances in an actual class while each column represents the instances in a predicted class.

Accuracy: - It is a way to measure how often the algorithm classifies a data points correctly. Accuracy is the number of correctly predicted data points among all the data points **[Figure 13,18,24]**.

Mean Absolute Error: Between the actual and anticipated values, it calculates the typical absolute difference. The average of the absolute discrepancies between the projected and actual values is used to compute it. MAE is less susceptible to outliers than other measures since it is represented in the same units as the objective variable **[Figure 13,18,24]**.

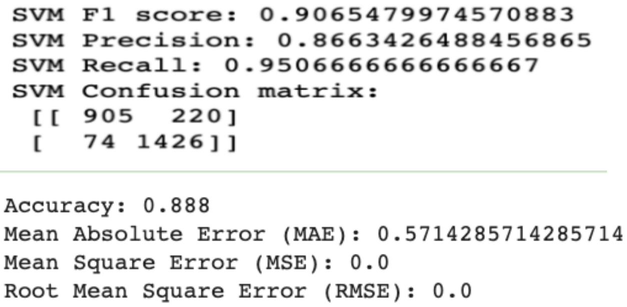
Mean Squared Error: It calculates the arithmetical mean of the squared deviation between the actual and expected values. It is determined by taking the square root of the average of the disparities between the expected and actual values **[Figure 13,18,24]**.

Root Mean Squared Error: The average discrepancy between a variable's anticipated and actual values is measured using the performance metric known as root mean squared error in regression analysis. The mean of the squared discrepancies between the expected and actual values is what this term is calculated as **[Figure 13,18,24]**.

Precision-Recall Curve: The precision-recall curve allows to visualize the tradeoff between precision and recall at different thresholds for the predicted probabilities **[Figure 15,20,26]**.

Receiver Operating Characteristic Curve: A classification model's performance at various classification thresholds is depicted graphically by a ROC curve. The true positive rate (TPR) and false positive rate (FPR) based on different threshold values are plotted here. A good model will produce a curve nearer to the plot's top left corner by having a high TPR and a low FPR. that the model may not be able to correctly identify all TBI cases, resulting in lower recall **[Figure 16,21,27]**.

3.2.1 Support Vector Machine Results



```
SVM F1 score: 0.9065479974570883
SVM Precision: 0.8663426488456865
SVM Recall: 0.9506666666666667
SVM Confusion matrix:
 [[ 905  220]
 [  74 1426]]
```

```
Accuracy: 0.888
Mean Absolute Error (MAE): 0.5714285714285714
Mean Square Error (MSE): 0.0
Root Mean Square Error (RMSE): 0.0
```

Figure13: Evaluation metrics on SVM.

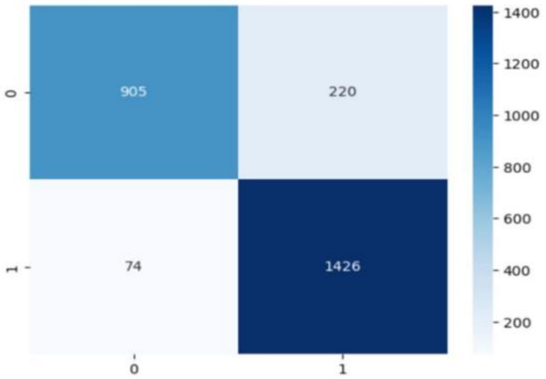


Figure14: SVM Confusion matrix plot.

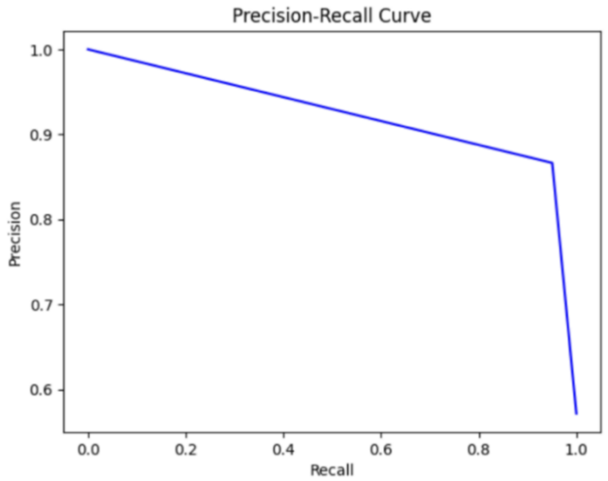


Figure15: SVM Precision-recall curve.

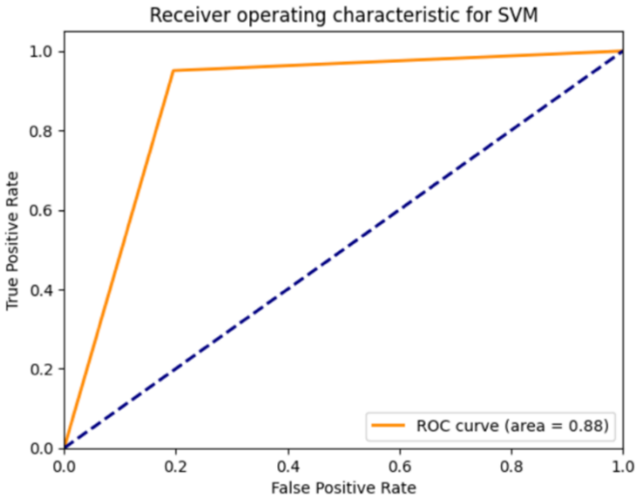


Figure16: SVM ROC curve.

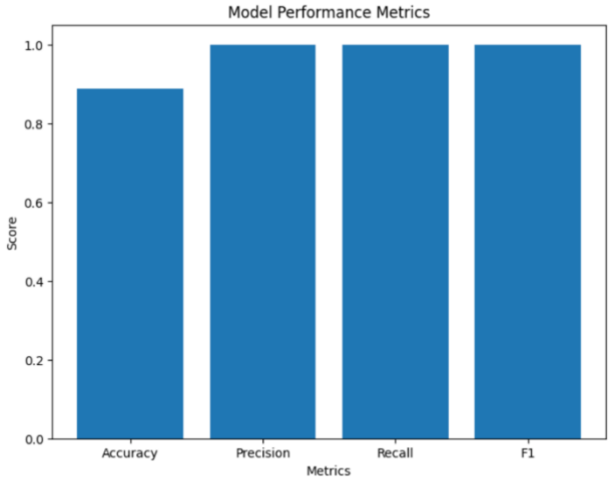
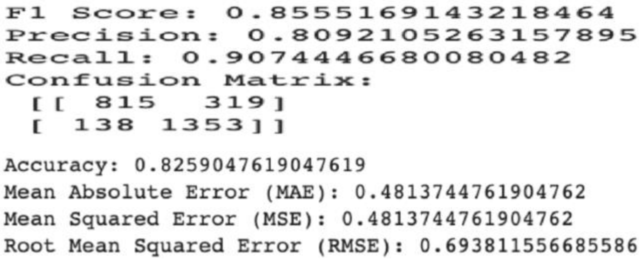


Figure17: SVM model performance metrics plot.

3.2.2 Convolution Neural Networks Results



```
F1 Score: 0.8555169143218464
Precision: 0.8092105263157895
Recall: 0.9074446680080482
Confusion Matrix:
 [[ 815  319]
 [ 138 1353]]

Accuracy: 0.8259047619047619
Mean Absolute Error (MAE): 0.4813744761904762
Mean Squared Error (MSE): 0.4813744761904762
Root Mean Squared Error (RMSE): 0.693811556685586
```

Figure18: Evaluation metrics on CNN.

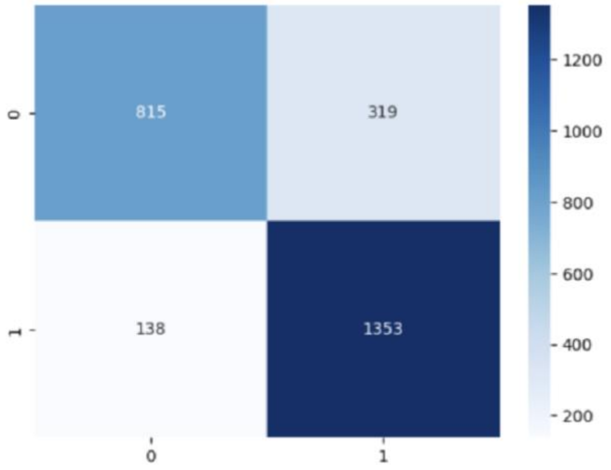


Figure19: CNN Confusion matrix plot.

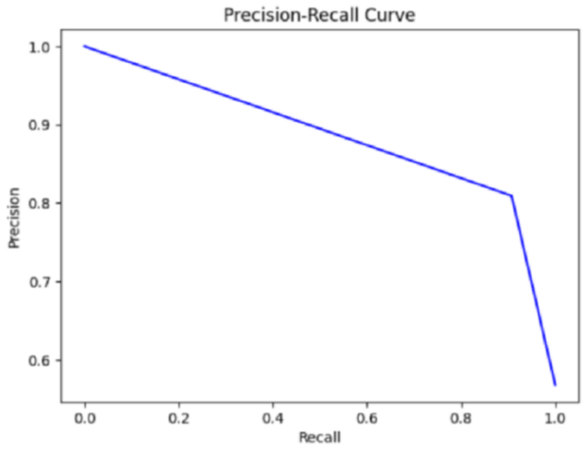


Figure20: CNN Precision-recall curve.

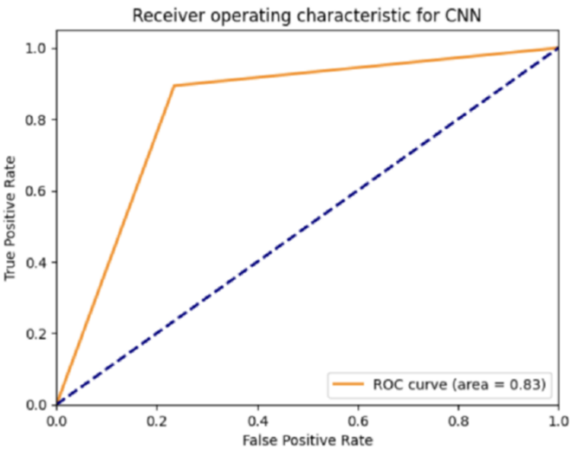


Figure21: CNN ROC curve.

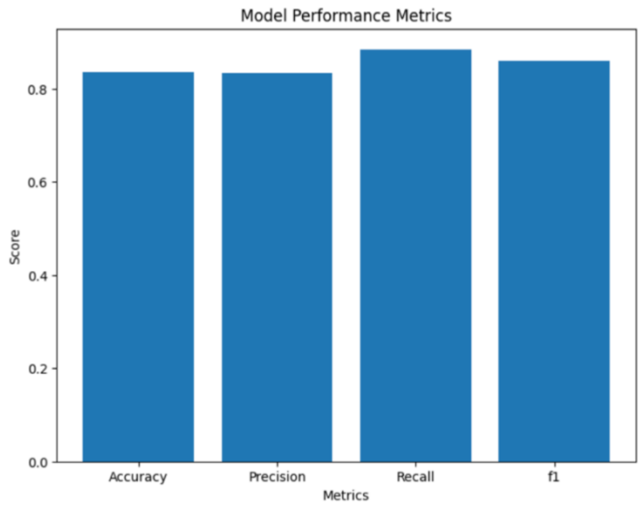
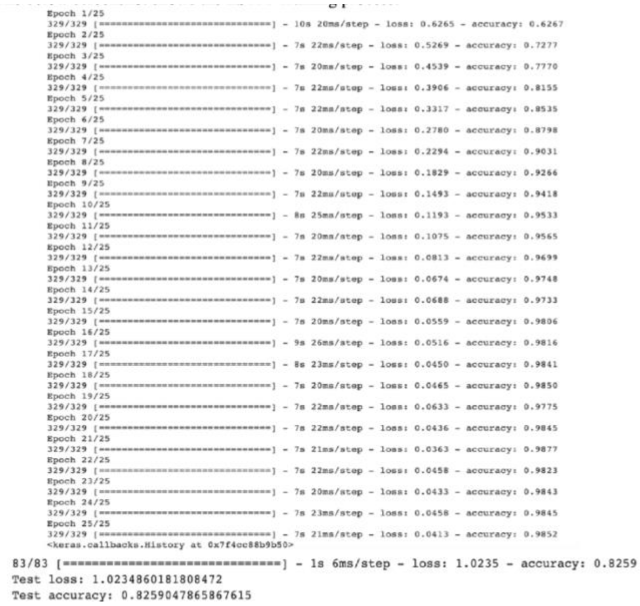


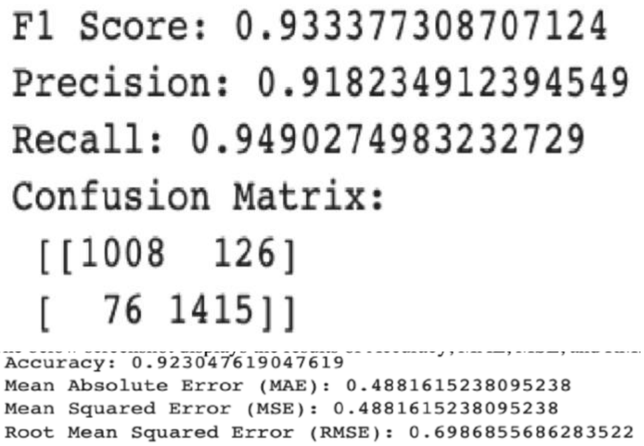
Figure22: CNN model performance metrics plot.



```
Epoch 1/25
329/329 [==============================] - 10s 20ms/step - loss: 0.6265 - accuracy: 0.6267
Epoch 2/25
329/329 [==============================] - 7s 22ms/step - loss: 0.5269 - accuracy: 0.7277
Epoch 3/25
329/329 [==============================] - 7s 20ms/step - loss: 0.4539 - accuracy: 0.7770
Epoch 4/25
329/329 [==============================] - 7s 22ms/step - loss: 0.3906 - accuracy: 0.8155
Epoch 5/25
329/329 [==============================] - 7s 22ms/step - loss: 0.3317 - accuracy: 0.8535
Epoch 6/25
329/329 [==============================] - 7s 20ms/step - loss: 0.2780 - accuracy: 0.8798
Epoch 7/25
329/329 [==============================] - 7s 22ms/step - loss: 0.2294 - accuracy: 0.9031
Epoch 8/25
329/329 [==============================] - 7s 20ms/step - loss: 0.1829 - accuracy: 0.9266
Epoch 9/25
329/329 [==============================] - 7s 22ms/step - loss: 0.1493 - accuracy: 0.9418
Epoch 10/25
329/329 [==============================] - 8s 25ms/step - loss: 0.1193 - accuracy: 0.9533
Epoch 11/25
329/329 [==============================] - 7s 20ms/step - loss: 0.1075 - accuracy: 0.9565
Epoch 12/25
329/329 [==============================] - 7s 22ms/step - loss: 0.0813 - accuracy: 0.9699
Epoch 13/25
329/329 [==============================] - 7s 20ms/step - loss: 0.0674 - accuracy: 0.9748
Epoch 14/25
329/329 [==============================] - 7s 22ms/step - loss: 0.0688 - accuracy: 0.9733
Epoch 15/25
329/329 [==============================] - 7s 20ms/step - loss: 0.0559 - accuracy: 0.9806
Epoch 16/25
329/329 [==============================] - 9s 26ms/step - loss: 0.0516 - accuracy: 0.9816
Epoch 17/25
329/329 [==============================] - 8s 23ms/step - loss: 0.0450 - accuracy: 0.9841
Epoch 18/25
329/329 [==============================] - 7s 20ms/step - loss: 0.0465 - accuracy: 0.9850
Epoch 19/25
329/329 [==============================] - 7s 22ms/step - loss: 0.0633 - accuracy: 0.9775
Epoch 20/25
329/329 [==============================] - 7s 22ms/step - loss: 0.0436 - accuracy: 0.9845
Epoch 21/25
329/329 [==============================] - 7s 21ms/step - loss: 0.0363 - accuracy: 0.9877
Epoch 22/25
329/329 [==============================] - 7s 22ms/step - loss: 0.0458 - accuracy: 0.9823
Epoch 23/25
329/329 [==============================] - 7s 20ms/step - loss: 0.0433 - accuracy: 0.9843
Epoch 24/25
329/329 [==============================] - 7s 23ms/step - loss: 0.0458 - accuracy: 0.9845
Epoch 25/25
329/329 [==============================] - 7s 21ms/step - loss: 0.0413 - accuracy: 0.9852
<keras.callbacks.History at 0x7f4cc88b9b50>
83/83 [==============================] - 1s 6ms/step - loss: 1.0235 - accuracy: 0.8259
Test loss: 1.0234860181808472
Test accuracy: 0.8259047865867615
```

Figure23: The above image shows epochs, accuracy, test loss.

3.2.3 Long Short-Term Memory Results



```
F1 Score: 0.933377308707124
Precision: 0.918234912394549
Recall: 0.9490274983232729
Confusion Matrix:
 [[1008  126]
 [  76 1415]]
```

```
Accuracy: 0.923047619047619
Mean Absolute Error (MAE): 0.4881615238095238
Mean Squared Error (MSE): 0.4881615238095238
Root Mean Squared Error (RMSE): 0.6986855686283522
```

Figure24: Evaluation metrics on LSTM.

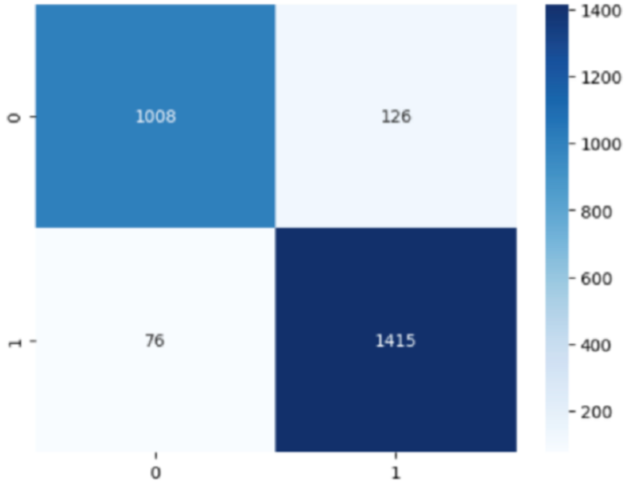


Figure25: LSTM Confusion matrix plot.

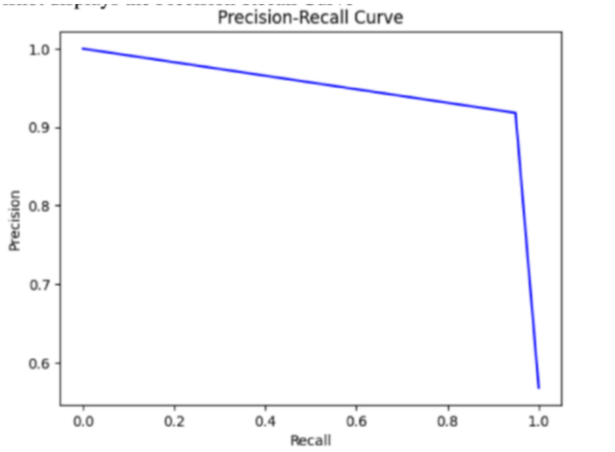


Figure26: LSTM Precision-recall curve.

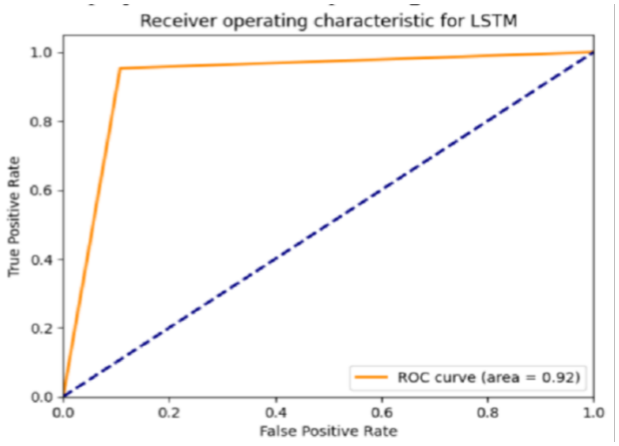


Figure27: LSTM ROC curve.

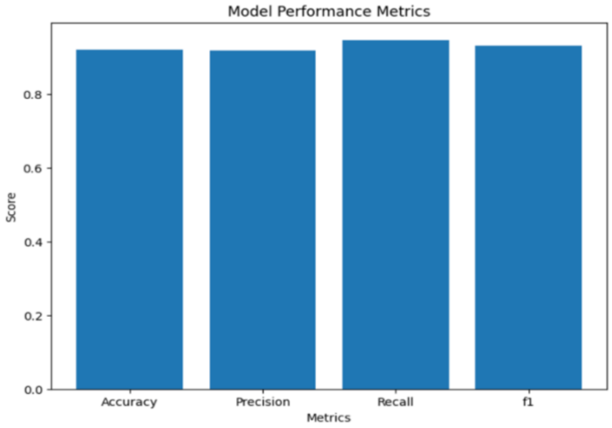
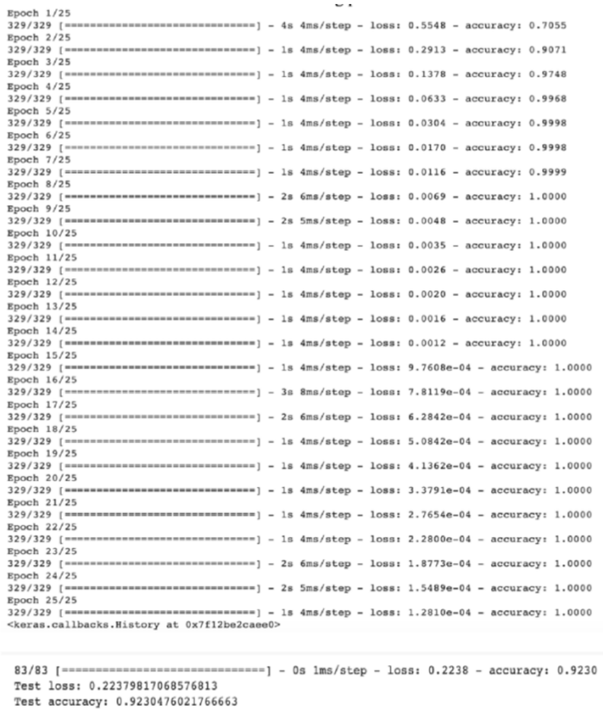


Figure28: LSTM model performance metrics plot.



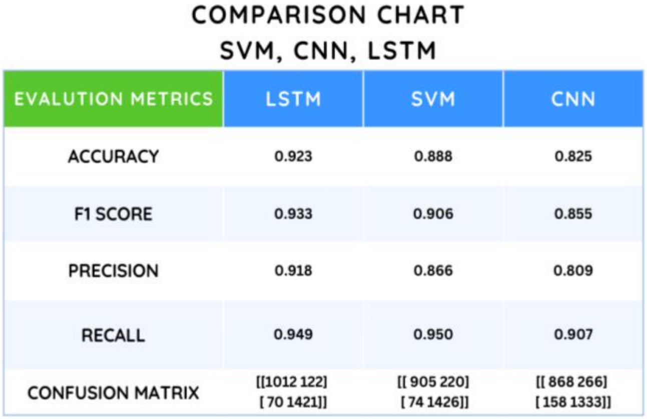
```
Epoch 1/25
329/329 [==============================] - 4s 4ms/step - loss: 0.5548 - accuracy: 0.7055
Epoch 2/25
329/329 [==============================] - 1s 4ms/step - loss: 0.2913 - accuracy: 0.9071
Epoch 3/25
329/329 [==============================] - 1s 4ms/step - loss: 0.1378 - accuracy: 0.9748
Epoch 4/25
329/329 [==============================] - 1s 4ms/step - loss: 0.0633 - accuracy: 0.9968
Epoch 5/25
329/329 [==============================] - 1s 4ms/step - loss: 0.0304 - accuracy: 0.9998
Epoch 6/25
329/329 [==============================] - 1s 4ms/step - loss: 0.0170 - accuracy: 0.9998
Epoch 7/25
329/329 [==============================] - 1s 4ms/step - loss: 0.0116 - accuracy: 0.9999
Epoch 8/25
329/329 [==============================] - 2s 6ms/step - loss: 0.0069 - accuracy: 1.0000
Epoch 9/25
329/329 [==============================] - 2s 5ms/step - loss: 0.0048 - accuracy: 1.0000
Epoch 10/25
329/329 [==============================] - 1s 4ms/step - loss: 0.0035 - accuracy: 1.0000
Epoch 11/25
329/329 [==============================] - 1s 4ms/step - loss: 0.0026 - accuracy: 1.0000
Epoch 12/25
329/329 [==============================] - 1s 4ms/step - loss: 0.0020 - accuracy: 1.0000
Epoch 13/25
329/329 [==============================] - 1s 4ms/step - loss: 0.0016 - accuracy: 1.0000
Epoch 14/25
329/329 [==============================] - 1s 4ms/step - loss: 0.0012 - accuracy: 1.0000
Epoch 15/25
329/329 [==============================] - 1s 4ms/step - loss: 9.7608e-04 - accuracy: 1.0000
Epoch 16/25
329/329 [==============================] - 3s 8ms/step - loss: 7.8119e-04 - accuracy: 1.0000
Epoch 17/25
329/329 [==============================] - 2s 6ms/step - loss: 6.2842e-04 - accuracy: 1.0000
Epoch 18/25
329/329 [==============================] - 1s 4ms/step - loss: 5.0842e-04 - accuracy: 1.0000
Epoch 19/25
329/329 [==============================] - 1s 4ms/step - loss: 4.1362e-04 - accuracy: 1.0000
Epoch 20/25
329/329 [==============================] - 1s 4ms/step - loss: 3.3791e-04 - accuracy: 1.0000
Epoch 21/25
329/329 [==============================] - 1s 4ms/step - loss: 2.7654e-04 - accuracy: 1.0000
Epoch 22/25
329/329 [==============================] - 1s 4ms/step - loss: 2.2800e-04 - accuracy: 1.0000
Epoch 23/25
329/329 [==============================] - 2s 6ms/step - loss: 1.8773e-04 - accuracy: 1.0000
Epoch 24/25
329/329 [==============================] - 2s 5ms/step - loss: 1.5489e-04 - accuracy: 1.0000
Epoch 25/25
329/329 [==============================] - 1s 4ms/step - loss: 1.2810e-04 - accuracy: 1.0000
<keras.callbacks.History at 0x7f12be2caee0>
```

```
83/83 [==============================] - 0s 1ms/step - loss: 0.2238 - accuracy: 0.9230
Test loss: 0.22379817068576813
Test accuracy: 0.9230476021766663
```

Figure29: The above image shows epochs, accuracy, test loss.

3.3 Briefly discuss which model fits the data better.

In order to predict whether the patient csv files are TBI or not we have used LSTM, SVM, and CNN on Alzheimer's dataset. The SVM model achieved an accuracy of 88 percent, correctly predicting 34 out of 36 files. The F1 score of the SVM model was 0.906, precision was 0.866, and recall was 0.950. Later two models were used, namely the LSTM and CNN. The LSTM model achieved the highest accuracy of 92 percent, correctly predicting 35 out of 36 files. The F1 score of the LSTM model was 0.933, precision was 0.918, and recall was 0.949. The CNN model achieved an accuracy of 81 percent, correctly predicting 30 out of 36 files. The F1 score of the CNN model was 0.855, precision was 0.809, and recall was 0.907. Overall, it seems that the LSTM model performed the best in terms of accuracy, correctly predicting the most number of files. The CNN model had the lowest accuracy but still achieved a reasonable F1 score. The SVM model had a lower accuracy than LSTM, but still achieved a decent F1 score. It is also important to note that other evaluation metrics such as precision, recall, and F1 score should be considered when evaluating model performance, not just accuracy. Overall, based on these metrics, the LSTM model seems to be the most effective model for predicting Whether a patient is TBI or not. The below table show the evaluation metrics comparison between CNN, SVM, and LSTM **[Figure 30]**. We could come to the conclusion that Long Short-Term Memory has the best accuracy and the lowest mean absolute error based on the observations we made from the estimated values we incurred from the approach we utilized. As a consequence, it operates most effectively and produces the most accurate results.



COMPARISON CHART
SVM, CNN, LSTM

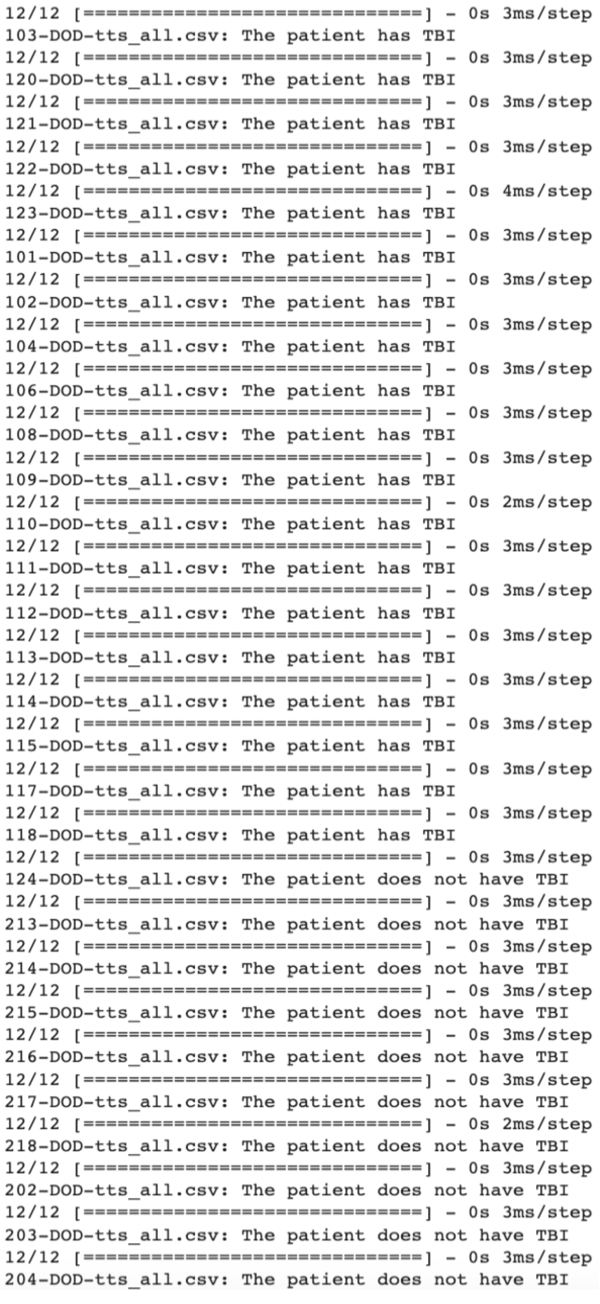
EVALUTION METRICS	LSTM	SVM	CNN
ACCURACY	0.923	0.888	0.825
F1 SCORE	0.933	0.906	0.855
PRECISION	0.918	0.866	0.809
RECALL	0.949	0.950	0.907
CONFUSION MATRIX	[[1012 122] [70 1421]]	[[905 220] [74 1426]]	[[868 266] [158 1333]]

Figure30: Comparison chart on SVM, CNN, LSTM.

3.4 For the model that fits the data best, make predictions for at least three cases of interest. One option is to show changes in predicted outcomes for changes in one of the predictors, holding all other predictors constant. Another option is to calculate predicted outcomes for particular cases of interest from the data set, or for hypothetical cases that are of interest.

Accuracy is not the only metric considered since this is a medical diagnosis. Forecasting the actual values while making a medical diagnosis accurately is crucial. After an accurate diagnosis report reveals whether a patient has a traumatic brain injury, we cannot misdiagnose them as a regular occurrence. So along with higher accuracy, we need higher recall. Comparatively, LSTM gives 92% accuracy, and the recall score is 0.949 on test and train data. Hence, for our specific dataset, long-term memory appears to be a better match. The second-best fit is SVM which has a recall score of 0.950, followed by CNN, with a recall score of 0.90.

The patient's data are predicted using the LSTM model to determine whether the patient has a TBI. We can see from the screenshot below that the model correctly predicted 35 patients .csv files out of 36 patients' .csv files. 92% of the time, the Long short-term Memory model can correctly predict whether a patient has suffered a traumatic brain injury or not **[Figure 31]**.



```
12/12 [==============================] - 0s 3ms/step
103-DOD-tts_all.csv: The patient has TBI
12/12 [==============================] - 0s 3ms/step
120-DOD-tts_all.csv: The patient has TBI
12/12 [==============================] - 0s 3ms/step
121-DOD-tts_all.csv: The patient has TBI
12/12 [==============================] - 0s 3ms/step
122-DOD-tts_all.csv: The patient has TBI
12/12 [==============================] - 0s 4ms/step
123-DOD-tts_all.csv: The patient has TBI
12/12 [==============================] - 0s 3ms/step
101-DOD-tts_all.csv: The patient has TBI
12/12 [==============================] - 0s 3ms/step
102-DOD-tts_all.csv: The patient has TBI
12/12 [==============================] - 0s 3ms/step
104-DOD-tts_all.csv: The patient has TBI
12/12 [==============================] - 0s 3ms/step
106-DOD-tts_all.csv: The patient has TBI
12/12 [==============================] - 0s 3ms/step
108-DOD-tts_all.csv: The patient has TBI
12/12 [==============================] - 0s 3ms/step
109-DOD-tts_all.csv: The patient has TBI
12/12 [==============================] - 0s 2ms/step
110-DOD-tts_all.csv: The patient has TBI
12/12 [==============================] - 0s 3ms/step
111-DOD-tts_all.csv: The patient has TBI
12/12 [==============================] - 0s 3ms/step
112-DOD-tts_all.csv: The patient has TBI
12/12 [==============================] - 0s 3ms/step
113-DOD-tts_all.csv: The patient has TBI
12/12 [==============================] - 0s 3ms/step
114-DOD-tts_all.csv: The patient has TBI
12/12 [==============================] - 0s 3ms/step
115-DOD-tts_all.csv: The patient has TBI
12/12 [==============================] - 0s 3ms/step
117-DOD-tts_all.csv: The patient has TBI
12/12 [==============================] - 0s 3ms/step
118-DOD-tts_all.csv: The patient has TBI
12/12 [==============================] - 0s 3ms/step
124-DOD-tts_all.csv: The patient does not have TBI
12/12 [==============================] - 0s 3ms/step
213-DOD-tts_all.csv: The patient does not have TBI
12/12 [==============================] - 0s 3ms/step
214-DOD-tts_all.csv: The patient does not have TBI
12/12 [==============================] - 0s 3ms/step
215-DOD-tts_all.csv: The patient does not have TBI
12/12 [==============================] - 0s 3ms/step
216-DOD-tts_all.csv: The patient does not have TBI
12/12 [==============================] - 0s 3ms/step
217-DOD-tts_all.csv: The patient does not have TBI
12/12 [==============================] - 0s 2ms/step
218-DOD-tts_all.csv: The patient does not have TBI
12/12 [==============================] - 0s 3ms/step
202-DOD-tts_all.csv: The patient does not have TBI
12/12 [==============================] - 0s 3ms/step
203-DOD-tts_all.csv: The patient does not have TBI
12/12 [==============================] - 0s 3ms/step
204-DOD-tts_all.csv: The patient does not have TBI
```

Figure31: The output of our LSTM prediction is shown above.

4. Summary and Conclusion

4.1 Briefly summarize what you have learned from your project.

In this project, we got to know that majority of the people who have Traumatic Brain injury will have more for getting Alzheimer’s, the most prevalent type of dementia, Alzheimer's disease is a neurological disorder with uncertain mechanism and causes that primarily affects elderly people and discovered that in order to predict the disease, a suitable machine learning model is needed. In order to do it, we got the data from our instructor which is ADNI datasets. For this project, we used TBI positive and TBI negative. TBI positive consists of 20 patients who are having Traumatic Brain injury and TBI negative consist of 16 patients’ data who are not having traumatic Brain injury. In each patient’s data, there are different regions of the brain which are attributes. After that, we have cleaned the data in order to train the models. we have implemented Convolution Neural Networks and Long Short-Term Memory (LSTM), which are deep learning models, using the key predictor. Prior to implementing the model, we looked at Support Vector Machine, a machine learning model, to understand which model performs the best in which specific situations.

Finally, thanks to this project, we have learned how to plot graphs for a given dataset, how to conduct exploratory data analysis, how to use machine learning models to predict outcomes, and how to determine the accuracy, recall, Area Under the Curve (AUC), Mean Absolute Error, Mean Squared Error, and Root Mean Squared Error and their significance. We also discovered the significance of deep learning methods.

4.2 Going back to the question that has motivated your project, how would you answer that question given the results of your analysis?

We demonstrate that we have a got 92.3 percent accuracy rate applying the LSTM model. We claim that the LSTM is the best-performing model by considering the other accuracy levels. Additionally, we were better than other models at predicting the patients' future outcomes.

4.3 Think about domain experts in the field you have analyzed. What can they learn from your project? How could the results of your analysis inform their work?

The medical professionals who specialize in treating Alzheimer's disease would be the subject matter experts for this investigation. Through this project, the doctors will be able to forecast the disease for a specific patient and determine whether or not he will become ill in the future. The doctors can learn about the patients who might be affected in the future from my findings because we were able to forecast the future outcomes of the patients using the LSTM model. More work can be done in this area by using more concepts in Deep learning.

4.4 Identify one way that your project could be improved if you had more time and resources to work on this project. For example, what additional data would you gather? What alternative data cleaning decisions would you make? What additional models would you estimate?

If we have additional time, we would like to incorporate the MRI brain scan data from the Alzheimer's dataset and attempt to apply deep learning models to the dataset. For instance, a sparse model could yield greater accuracy. Since the data provider only provided a small number of null values, the data was cleaned.

5. Comments

Checkpoint1: Feedback from dr. Hubig: do not include code in report, more explanation about results in writing, good job with plots, add more plots of different kind. Feedback from Nushrat: Explain in detail about project objective and how your EDA helps you with the objective of Alzheimer’s.

As from the feedback requirements we have included correlation network that plots the correlation matrix of TBI-POS, TBINEG, and HC-ALL. And explained clearly about the data cleaning steps, key predictors and project purpose.