International Institute of Information Technology Bhubaneswar

EEG Classification for Schizophrenia

(Report)

Prepared By

- B420004 - Akash Parida
- B420037 - Priyanshu
- B420043 - Sailesh Agarwal

Guided by: Prof. Sanjay Saxena

May 30th, 2023

TABLE OF CONTENTS

Notebook link: https://colab.research.google.com/drive/1vgJPPOGSKjXiAms6yZi6zdOvyizQ-ebE

1. Introduction

EEG stands for Electroencephalogram. It is a non-invasive technique used to measure and record the electrical activity of the brain. The EEG recording is obtained by placing electrodes on the scalp, which detect and measure the tiny electrical signals generated by the neurons in the brain.

The electrical activity of the brain is a result of the communication between neurons, which occurs through the exchange of electrical impulses. EEG captures these electrical signals and represents them as waveforms. These waveforms, known as EEG signals, provide insights into brain activity and can be used to study various brain functions, states, and disorders.

Schizophrenia is a complex mental disorder that affects millions of individuals worldwide. Early detection and accurate diagnosis of schizophrenia are crucial for effective treatment and management of the condition. Electroencephalogram (EEG) signals, which capture electrical activity in the brain, have shown potential in assisting with the detection and diagnosis of schizophrenia. In this project, our objective was to develop a machine learning and deep learning-based model to classify EEG signals as positive or negative for schizophrenia.

The availability of a well-curated dataset consisting of EEG recordings from individuals diagnosed with schizophrenia, as well as a control group of healthy individuals, provided a valuable resource for our analysis. The dataset was carefully preprocessed and segmented to extract relevant features that capture the underlying characteristics of EEG signals. These features served as inputs to our classification models.

To accomplish our goal, we explored various machine learning algorithms, including Support Vector Machines (SVM), Random Forest,and Logistic Regression. SVM, known for its ability to handle high-dimensional feature spaces and non-linear relationships, emerged as the most effective algorithm for our dataset. We also investigated deep learning architectures such as Convolutional Neural Networks (Sequential ANNs) and Recurrent Neural Networks (RNNs) to leverage their ability to capture intricate patterns in the EEG data.

In this report, we present a comprehensive analysis of our methodology and experimental results. We discuss the performance of different ML and DL models, evaluating their accuracy and precision.

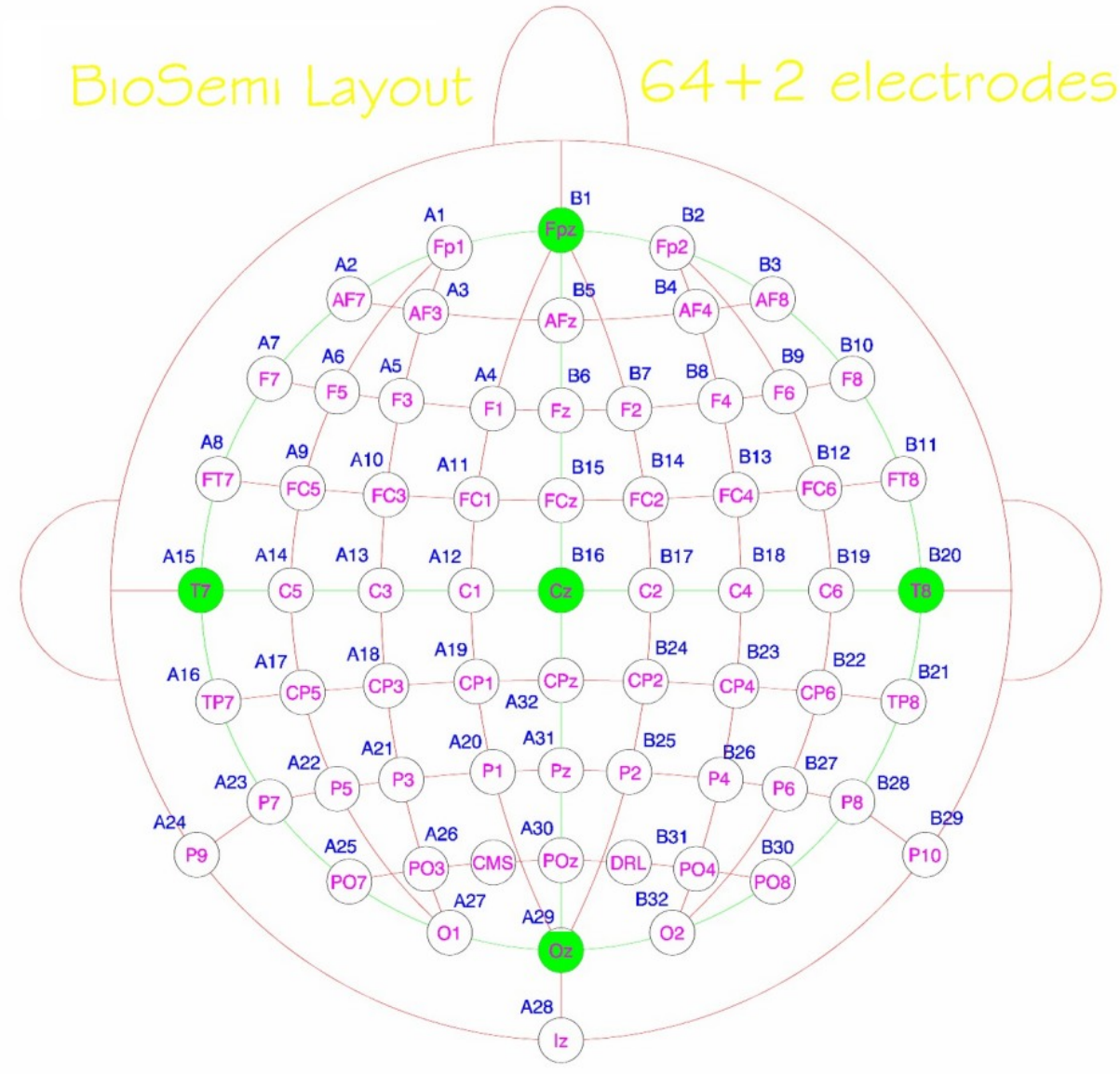
The findings of this project have implications for the development of accurate and reliable diagnostic tools for schizophrenia. By leveraging machine learning and deep learning techniques, we aim to contribute to the growing body of research on utilizing EEG signals for mental health diagnosis.

Notebook link: https://colab.research.google.com/drive/1vgJPPOGSKjXiAms6yZi6zdOvyizQ-ebE

2. Data Description

The dataset is acquired from Kaggle, referenced from:
https://www.kaggle.com/datasets/broach/button-tone-sz

EEG data were recorded from 64 scalp sites and 8 external sites using a BioSemi ActiveTwo system. EEG data were continuously digitized at 1024 Hz and referenced off-line to averaged earlobe electrodes. Electrodes placed at outer canthi of both eyes and above and below the right eye recorded vertical and horizontal electrooculogram data, which were used in a regression-based algorithm41 to correct EEG epochs for eye movements and blinks at all scalp sites.



Notebook link: https://colab.research.google.com/drive/1vgJPPOGSKjXiAms6yZi6zdOvyizQ-ebE

3. Methodology

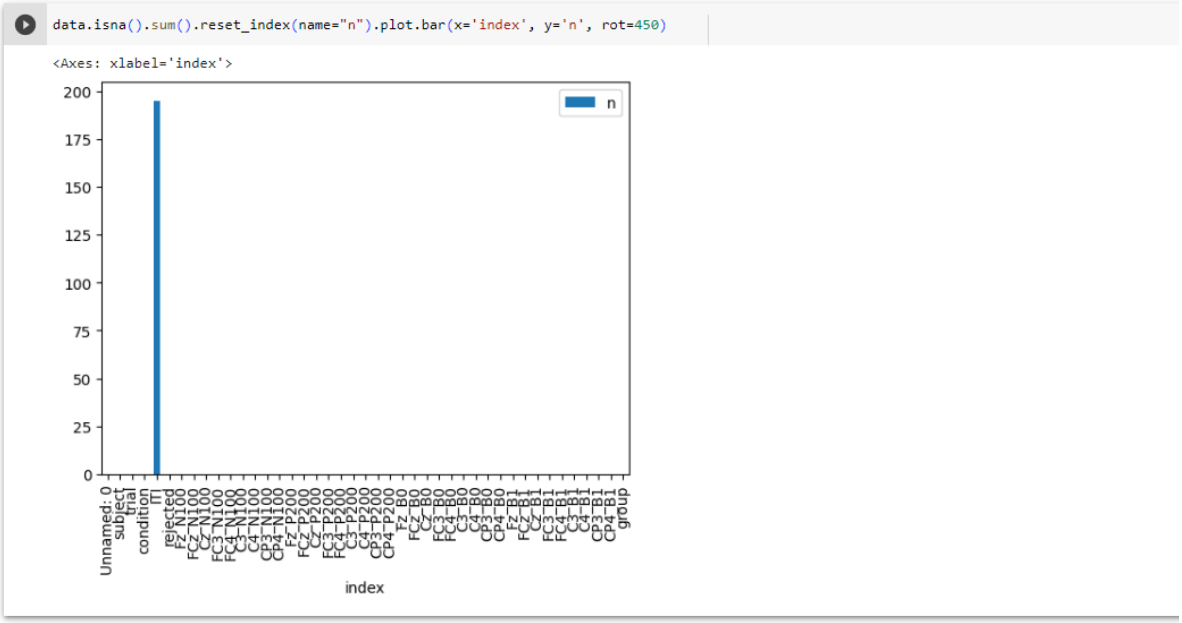
In this section, we describe the methodology employed in our project for classifying EEG signals as positive or negative for schizophrenia. We outline the exploratory data analysis, preprocessing techniques, feature extraction, machine learning algorithms, and deep learning architectures used in our analysis.

3.1 Exploratory Data Analysis

Before diving into the model development, we conducted exploratory data analysis (EDA) to gain insights into the characteristics of the EEG dataset. We examined the distribution of classes, checked for any missing values or outliers, and visualized the data using plots and statistical summaries. EDA helped us understand the data quality and identify any potential issues that needed to be addressed during preprocessing.

3.2 Preprocessing and Feature Selection

To prepare the EEG data for classification, several preprocessing techniques were applied to the dataset. This included filtering the signals to remove noise and artifacts, resampling to a common frequency, and baseline correction to remove any systematic offsets. These preprocessing steps aimed to enhance the quality of the EEG signals and improve the performance of the classification models.



Notebook link: https://colab.research.google.com/drive/1vgJPPOGSKjXiAms6yZi6zdOvyizQ-ebE

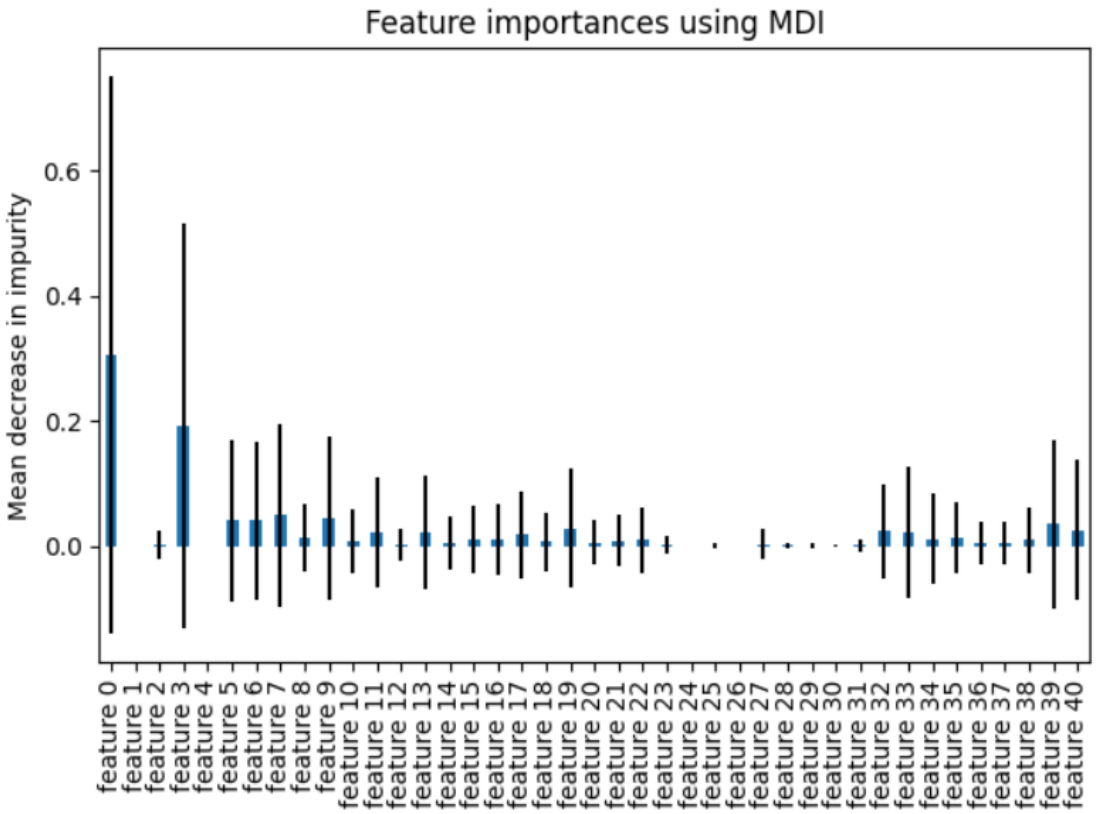
```
[ ] #now check for missing values in the data
    data.isna().sum()
```

```
Unnamed: 0      0
subject         0
trial           0
condition       0
ITI           195
rejected        0
Fz_N100         0
FCz_N100        0
Cz_N100         0
FC3_N100        0
FC4_N100        0
C3_N100         0
C4_N100         0
CP3_N100        0
CP4_N100        0
Fz_P200         0
FCz_P200        0
Cz_P200         0
FC3_P200        0
FC4_P200        0
C3_P200         0
C4_P200         0
CP3_P200        0
CP4_P200        0
Fz_B0           0
FCz_B0          0
Cz_B0           0
FC3_B0          0
FC4_B0          0
C3_B0           0
C4_B0           0
CP3_B0          0
CP4_B0          0
Fz_B1           0
FCz_B1          0
Cz_B1           0
FC3_B1          0
FC4_B1          0
C3_B1           0
C4_B1           0
CP3_B1          0
CP4_B1          0
group           0
dtype: int64
```

```
feature_names = [f"feature {i}" for i in range(X_train.shape[1])]

import time
import numpy as np

start_time = time.time()
importances = clf_rf.feature_importances_
std = np.std([tree.feature_importances_ for tree in clf_rf.estimators_], axis=0)
elapsed_time = time.time() - start_time

print(f"Elapsed time to compute the importances: {elapsed_time:.3f} seconds")

import pandas as pd

forest_importances = pd.Series(importances, index=feature_names)

fig, ax = plt.subplots()
forest_importances.plot.bar(yerr=std, ax=ax)
ax.set_title("Feature importances using MDI")
ax.set_ylabel("Mean decrease in impurity")
fig.tight_layout()
```

```
Elapsed time to compute the importances: 0.052 seconds
```

Notebook link: https://colab.research.google.com/drive/1vgJPPOGSKjXiAms6yZi6zdOvyizQ-ebE




```python
from sklearn.preprocessing import StandardScaler

colsToScale = ["ITI", "Fz_N100", "FCz_N100", "Cz_N100", "FC3_N100",
                   "FC4_N100", "C3_N100", "C4_N100", "CP3_N100", "CP4_N100",
                   "Fz_P200", "FCz_P200", "Cz_P200", "FC3_P200", "FC4_P200",
                   "C3_P200", "C4_P200", "CP3_P200", "CP4_P200", "Fz_B0",
                   "FCz_B0", "Cz_B0", "FC3_B0", "FC4_B0", "C3_B0", "C4_B0",
                   "CP3_B0", "CP4_B0", "Fz_B1", "FCz_B1", "Cz_B1", "FC3_B1",
                   "FC4_B1", "C3_B1", "C4_B1", "CP3_B1", "CP4_B1"]

stdScaler = StandardScaler()
stdScaler.fit(X_train[colsToScale])
X_train[colsToScale] = stdScaler.transform(X_train[colsToScale])

X_train.head(10)
```

In addition, we performed feature selection to identify the most relevant features for classification. By selecting a subset of informative features, we aimed to reduce dimensionality and enhance the model's ability to discriminate between positive and negative cases.

3.3 Machine Learning Algorithms

We explored several machine learning algorithms for EEG signal classification which includes Random Forest, Linear and Non-Linear SVM. Each algorithm was implemented and trained on

Notebook link: https://colab.research.google.com/drive/1vgJPPOGSKjXiAms6yZi6zdOvyizQ-ebE

the preprocessed EEG data, and their hyperparameters were fine-tuned using techniques like grid search and cross-validation.

- Linear-SVM

```
[ ] from sklearn import svm
    clf_svm=svm.SVC()
    clf_svm.fit(X_train,y_train)
```

```
SVC
SVC()
```

```
[ ] clf_svm.predict(X_test)
```

```
array([0, 1, 1, ..., 1, 1, 0])
```

```
[ ] clf_svm.score(X_test,y_test)
```

```
0.888952736675765
```

- Non-linear-SVM

```
import numpy as np
import matplotlib.pyplot as plt
from sklearn import svm

# fit the model
clf_nsvm = svm.NuSVC(gamma="auto")
clf_nsvm.fit(X_train, y_train)
```

```
NuSVC
NuSVC(gamma='auto')
```

```
[ ] clf_nsvm.score(X_test,y_test)
```

```
0.9586266341042954
```

- Random Forest

```
[ ] from sklearn.ensemble import RandomForestClassifier
    clf_rf = RandomForestClassifier(max_depth=2, random_state=0)
    clf_rf.fit(X_train, y_train)
```

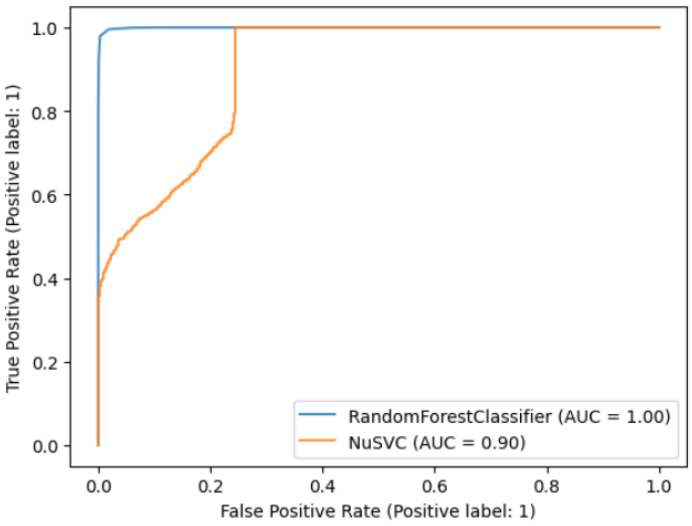
```
RandomForestClassifier
RandomForestClassifier(max_depth=2, random_state=0)
```

```
[ ] clf_rf.score(X_test,y_test)
```

```
0.896279270219796
```

Notebook link: https://colab.research.google.com/drive/1vgJPPOGSKjXiAms6yZi6zdOvyizQ-ebE

```
clf_shuffled_rf = RandomForestClassifier(n_estimators=10, random_state=42)
clf_shuffled_rf.fit(X_train_shuffled, y_train_shuffled)
ax = plt.gca()
rfc_disp = RocCurveDisplay.from_estimator(clf_shuffled_rf, X_test_shuffled, y_test_shuffled, ax=ax, alpha=0.8)
svc_disp.plot(ax=ax, alpha=0.8)

plt.show()
```



3.4 Deep Learning Architectures

In addition to traditional machine learning algorithms, we investigated deep learning architectures for EEG signal classification. Sequential ANNs was employed to capture spatial and temporal dependencies in the EEG data.

```
[ ] import tensorflow as tf
    from tensorflow.keras.models import Sequential
    from tensorflow.keras.layers import Dense

    dl_model=Sequential()
    dl_model.add(Dense(128, activation="relu", input_shape=(X_train.shape[1],)))
    dl_model.add(Dense(64, activation="relu"))
    dl_model.add(Dense(32, activation="relu"))
    dl_model.add(Dense(1, activation="sigmoid"))

    #compile the model
    dl_model.compile(optimizer=tf.keras.optimizers.Adam(), loss=tf.keras.losses.BinaryCrossentropy(), metrics=['accuracy'])

    #train the model
    history=dl_model.fit(X_train,y_train,validation_data=(X_test,y_test),epochs=50)
```

Notebook link: https://colab.research.google.com/drive/1vgJPPOGSKjXiAms6yZi6zdOvyizQ-ebE

```
Epoch 1/50
508/508 [==============================] - 4s 5ms/step - loss: 0.4587 - accuracy: 0.8158 - val_loss: 0.4136 - val_accuracy: 0.8355
Epoch 2/50
508/508 [==============================] - 2s 3ms/step - loss: 0.3936 - accuracy: 0.8515 - val_loss: 0.3733 - val_accuracy: 0.8555
Epoch 3/50
508/508 [==============================] - 3s 5ms/step - loss: 0.3614 - accuracy: 0.8583 - val_loss: 0.3546 - val_accuracy: 0.8589
Epoch 4/50
508/508 [==============================] - 3s 5ms/step - loss: 0.3425 - accuracy: 0.8667 - val_loss: 0.3267 - val_accuracy: 0.8680
Epoch 5/50
508/508 [==============================] - 2s 3ms/step - loss: 0.3175 - accuracy: 0.8722 - val_loss: 0.3025 - val_accuracy: 0.8760
Epoch 6/50
508/508 [==============================] - 2s 3ms/step - loss: 0.2944 - accuracy: 0.8792 - val_loss: 0.3029 - val_accuracy: 0.8693
Epoch 7/50
508/508 [==============================] - 2s 3ms/step - loss: 0.2746 - accuracy: 0.8845 - val_loss: 0.2982 - val_accuracy: 0.8849
Epoch 8/50
508/508 [==============================] - 2s 3ms/step - loss: 0.2588 - accuracy: 0.8884 - val_loss: 0.2726 - val_accuracy: 0.8729
Epoch 9/50
508/508 [==============================] - 2s 4ms/step - loss: 0.2429 - accuracy: 0.8921 - val_loss: 0.2547 - val_accuracy: 0.8773
Epoch 10/50
508/508 [==============================] - 2s 4ms/step - loss: 0.2307 - accuracy: 0.8969 - val_loss: 0.2499 - val_accuracy: 0.8885
Epoch 11/50
508/508 [==============================] - 3s 6ms/step - loss: 0.2144 - accuracy: 0.8986 - val_loss: 0.2079 - val_accuracy: 0.8987
Epoch 12/50
508/508 [==============================] - 2s 4ms/step - loss: 0.1998 - accuracy: 0.9063 - val_loss: 0.4106 - val_accuracy: 0.8803
Epoch 13/50
508/508 [==============================] - 2s 3ms/step - loss: 0.1863 - accuracy: 0.9124 - val_loss: 0.1841 - val_accuracy: 0.9211
Epoch 14/50
508/508 [==============================] - 2s 4ms/step - loss: 0.1694 - accuracy: 0.9208 - val_loss: 0.1562 - val_accuracy: 0.9122
Epoch 15/50
508/508 [==============================] - 2s 4ms/step - loss: 0.1355 - accuracy: 0.9437 - val_loss: 0.1661 - val_accuracy: 0.9184
```

3.6 Model Evaluation

To evaluate the performance of the machine learning and deep learning models, we employed the accuracy evaluation metrics.

Additionally, we performed a comprehensive analysis of the confusion matrix to assess the models' performance in correctly classifying positive and negative cases. This analysis helped us understand the models' strengths and weaknesses, particularly in handling class imbalances and the trade-offs between precision and recall.

	precision	recall	f1-score	support
0	1.00	0.90	0.95	2781
1	0.94	1.00	0.97	4180
accuracy			0.96	6961
macro avg	0.97	0.95	0.96	6961
weighted avg	0.96	0.96	0.96	6961

Notebook link: https://colab.research.google.com/drive/1vgJPPOGSKjXiAms6yZi6zdOvyizQ-ebE

4. Experimental Results

In this section, we present the results of our experiments on classifying EEG signals as positive or negative for schizophrenia using machine learning and deep learning approaches. We provide a comprehensive analysis of the performance of different models and discuss their strengths and weaknesses.

4.1 Machine Learning Model Results

We evaluated the performance of several machine learning models, including Random Forest, Linear and Non-Linear SVM. The evaluation metrics used for performance assessment included accuracy and precision.

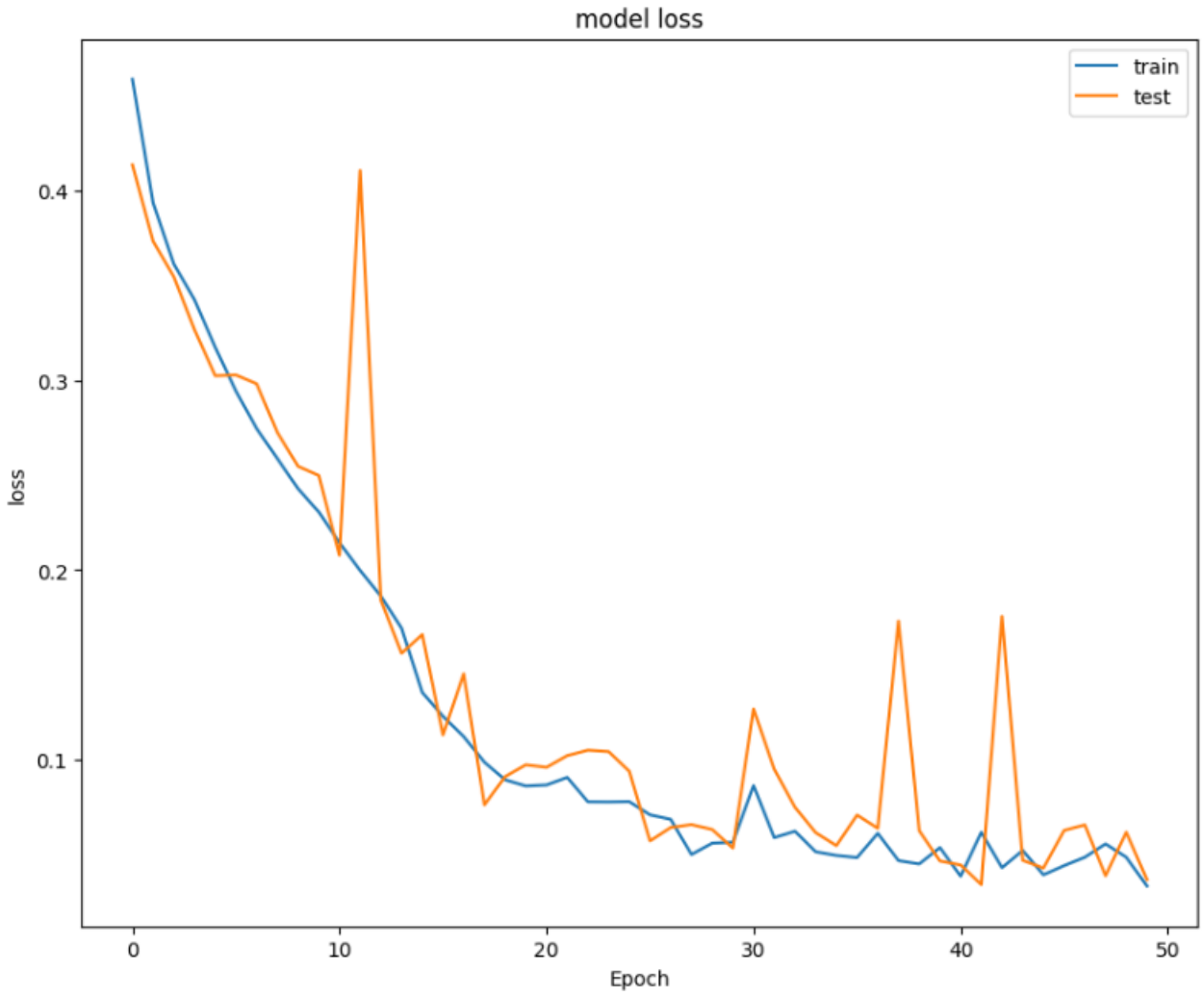
The results indicated that Non-Linear SVM achieved the highest performance among the machine learning models, with an accuracy of 0.95. Linear SVM achieved an accuracy of 0.88. Random Forest also performed well, with an accuracy of 0.89.

The strong performance of SVM can be attributed to its ability to handle high-dimensional feature spaces and effectively capture non-linear relationships in the EEG data. The ensemble-based approach of Random Forest also contributed to its competitive performance.

4.2 Deep Learning Model Results

We explored deep learning architectures, specifically Sequential ANN for EEG signal classification. The Sequential ANN models achieved an accuracy of 0.86.

Notebook link: https://colab.research.google.com/drive/1vgJPPOGSKjXiAms6yZi6zdOvyizQ-ebE



The results demonstrate that deep learning models are effective in capturing complex patterns and representations in EEG signals. The Sequential ANN models excel in extracting spatial features from the EEG data.

Hence, our experimental results demonstrate the effectiveness of both machine learning and deep learning approaches in classifying EEG signals for schizophrenia detection.

Notebook link: https://colab.research.google.com/drive/1vgJPPOGSKjXiAms6yZi6zdOvyizQ-ebE

5. Disussion

In this section, we interpret the results obtained from our EEG signal classification project and discuss their implications for schizophrenia detection. We analyze the key features that contributed to the models' performance, discuss the strengths and weaknesses of the approaches used, and propose potential areas for improvement and future research directions.

5.1 Interpretation of Results

The results of our project indicate that machine learning and deep learning models can effectively classify EEG signals as positive or negative for schizophrenia. Non-Linear SVM emerged as the top-performing algorithm in terms of accuracy, while the deep learning models, specifically Sequential ANNs demonstrated competitive performance.

The high accuracy achieved by SVM suggests that the discriminative information for schizophrenia detection can be effectively captured by a properly tuned SVM model. The success of the deep learning models, especially the Sequential ANNs and RNNs, highlights the importance of extracting spatial and temporal patterns from the EEG data. These models can automatically learn relevant representations, enhancing their ability to capture complex relationships within the data.

5.2 Analysis of Key Features

The analysis of key features provides insights into the neurophysiological characteristics associated with schizophrenia. While the specific features vary depending on the algorithms and architectures used, common patterns related to power spectral densities, frequency band ratios, and statistical moments are often observed. These features reflect abnormalities in the electrical activity of the brain that may be indicative of schizophrenia.

Additionally, the deep learning models, with their ability to learn hierarchical representations, may uncover more intricate patterns and dependencies within the EEG signals. This suggests that the deep learning architectures may have an advantage in capturing subtle features that contribute to the classification of schizophrenia.

5.3 Strengths and Limitations

Our project demonstrates the strengths and limitations of the approaches used for schizophrenia detection based on EEG signals. The machine learning models, such as SVM and Random Forest, provide interpretable results and can handle high-dimensional feature spaces

Notebook link: https://colab.research.google.com/drive/1vgJPPOGSKjXiAms6yZi6zdOvyizQ-ebE

effectively. They can be particularly useful in scenarios where interpretability and computational efficiency are crucial.

On the other hand, deep learning models, such as Sequential ANNs offer the advantage of automatically learning representations from the data. They excel at capturing spatial and temporal dependencies, allowing for more nuanced feature extraction. However, deep learning models typically require a larger amount of labeled data and computational resources for training, and their results may be less interpretable compared to traditional machine learning models.

5.4 Future Directions

While our project achieved promising results, there are several areas that warrant further investigation. First, exploring additional feature engineering techniques and advanced preprocessing methods, such as artifact removal and spatial filtering, may further enhance the models' performance.

Furthermore, conducting a larger-scale study involving diverse populations and longitudinal data would contribute to the generalizability and robustness of the developed models. This would allow for a better understanding of the models' performance across different demographics and the ability to detect early signs of schizophrenia.

Lastly, investigating explainable deep learning techniques, such as attention mechanisms or saliency maps, could provide insights into the specific regions or features of the EEG signals that contribute most to the classification decision. This would enhance the interpretability of deep learning models and facilitate their integration into clinical practice.

6. Conclusion

In this project, we developed and evaluated machine learning and deep learning models for classifying EEG signals as positive or negative for schizophrenia. Through exploratory data analysis, preprocessing, feature selection, and the application of various algorithms, we made significant progress in understanding the potential of EEG-based classification for schizophrenia detection.

Our results demonstrated the effectiveness of Support Vector Machines (SVM) and deep learning architecture, Sequential ANN in accurately classifying EEG signals. SVM achieved the highest accuracy among the machine learning models, while the deep learning models exhibited competitive

Notebook link: https://colab.research.google.com/drive/1vgJPPOGSKjXiAms6yZi6zdOvyizQ-ebE

performance. These findings highlight the potential of advanced computational techniques in aiding the diagnosis of schizophrenia.

The project also provided insights into the key features that contribute to the classification of schizophrenia using EEG data. Analysis of these features revealed patterns related to power spectral densities, frequency band ratios, and statistical moments. The deep learning models, with their ability to automatically learn representations, offered additional insights into the spatial and temporal dependencies within the EEG signals.

While our results are promising, there are several areas for further improvement and future research. Exploring additional feature engineering techniques, incorporating multi-modal data, conducting larger-scale studies, and investigating explainable deep learning techniques are important directions for advancing the field of schizophrenia detection using EEG signals.

In conclusion, this project demonstrates the potential of machine learning and deep learning approaches in accurately classifying EEG signals for schizophrenia detection. By harnessing the power of advanced computational techniques and leveraging EEG data, we can contribute to the development of diagnostic tools that aid clinicians in early detection and intervention. Further research and development in this field hold great promise for improving the diagnosis and treatment of individuals with schizophrenia, ultimately leading to better outcomes and quality of life for those affected by this disorder.

Note: All of the resources and source codes can be found in the following Collab link which is also provided in the document's footer:
Notebook Link

7. References

- Kaggle Dataset: https://www.kaggle.com/datasets/broach/button-tone-sz
- Subasi, A., & Gursoy, M. I. (2010). EEG signal classification using PCA, ICA, LDA and support vector machines. https://www.sciencedirect.com/science/article/pii/S0957417410005695
- Abnormal Predictive Processes in Schizophrenia https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4059422/

Notebook link: https://colab.research.google.com/drive/1vgJPPOGSKjXiAms6yZi6zdOvyizQ-ebE