

REPORT

Introduction

The project focuses on the analysis of EEG data using the Bonn EEG Dataset, a crucial resource in the field of neuroscience and medical diagnostics. Electroencephalography (EEG) is a non-invasive method that records the electrical activity of the brain. This technique is vital in studying brain functions and diagnosing conditions like epilepsy. The Bonn EEG Dataset specifically includes recordings related to epileptic seizures, offering a unique opportunity to explore and classify different patterns of brain activity associated with epilepsy. By applying advanced classification models to this dataset, the project aims to enhance the understanding and detection of epileptic seizures, which is crucial for improving patient care and treatment strategies. This research leverages the high-resolution temporal data provided by EEG recordings to identify distinctive patterns that differentiate between normal and abnormal brain activity, a step forward in the field of biomedical engineering and neuroscience.

Data preprocessing and feature extraction methods

Data Preprocessing

Data preprocessing is done on electroencephalogram (EEG) data that is saved in text files. Each file's content is read, ASCII data is converted to integers, and the data is arranged into a DataFrame. Filenames, sample indices, and matching EEG data are all displayed in a column of the DataFrame. By presenting details about the DataFrame, descriptive statistics, and sample EEG graphs for each file, the code offers more insights into the dataset.

The dataset contains 500 rows and 4100 columns after extracting and encoding. The raw data is cleaned, transformed, and organized to enhance its quality.

	Filename	Sample_1	Sample_2	Sample_3	Sample_4	Sample_5	Sample_6
\							
0	F022.txt	26	29	26	22	18	8
1	Z099.txt	56	55	38	-5	-47	-72
2	O093.txt	-83	-120	-123	-119	-93	-45
3	Z007.txt	-2	20	42	48	27	11
4	F032.txt	46	41	44	39	42	41
..
495	N062.TXT	-1	1	-1	5	-1	-3
496	O086.txt	-40	-19	-38	-71	-76	-86
497	Z050.txt	81	68	35	5	-6	6
498	Z044.txt	-46	-44	-47	-40	-21	4
499	S047.txt	-10	-658	-1254	-1395	-977	-198

	Sample_7	Sample_8	Sample_9	...	Sample_4088	Sample_4089	Sample_4
090							
\							
0	-1	-16	-26	...	53	55	
62							
1	-79	-62	-39	...	-44	-3	
36							
2	17	77	103	...	72	21	
-22							
3	9	48	82	...	-3	7	
-3							
4	41	34	13	...	50	17	
-28							
..	
...							
495	-3	3	12	...	22	35	
40							
496	-79	-60	-89	...	-52	-51	
-64							
497	38	86	107	...	44	38	
33							
498	18	25	-8	...	-17	1	
20							
499	419	785	873	...	-836	-329	
143							

	Sample_4091	Sample_4092	Sample_4093	Sample_4094	Sample_4095	\
0	60	63	69	65	49	
1	41	14	-27	-45	-32	
2	-31	-18	-3	-5	-27	
3	-17	-38	-38	-23	-18	
4	-63	-76	-88	-92	-84	
..	
495	50	49	50	55	53	
496	-72	-68	-60	-59	-46	
497	29	23	20	12	15	
498	32	38	40	48	43	
499	428	639	752	781	678	

	Sample_4096	Sample_4097
0	32	-150
1	-4	69
2	-50	-38
3	-6	-37
4	-75	-102
..
495	65	-6
496	-41	-63

Exploring the data is essential to unveil its inherent patterns and characteristics, providing valuable insights into its structure, distribution, and key features before initiating any analysis or modeling.

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 500 entries, 0 to 499
Columns: 4098 entries, Filename to Sample_4097
dtypes: int64(4097), object(1)
memory usage: 15.6+ MB
```

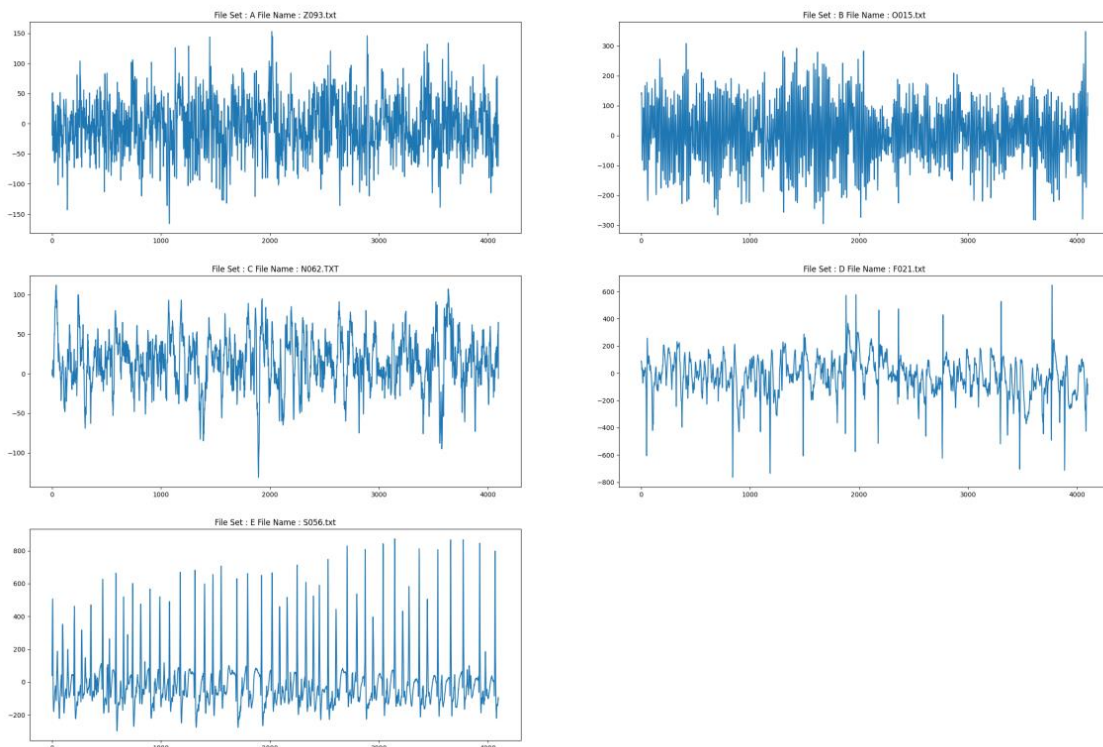
```
df.describe()
```

	Sample_1	Sample_2	Sample_3	Sample_4	Sample_5	Sample_6	Sample_7	Sample_8	Sample_9	Sample_10	...	Sample_4088
count	500.000000	500.000000	500.000000	500.000000	500.000000	500.000000	500.000000	500.000000	500.000000	500.000000	...	500.000000
mean	-3.718000	-9.802000	-16.094000	-18.820000	-16.662000	-12.124000	-6.510000	-2.142000	1.882000	4.438000	...	-5.706000
std	145.274622	163.176469	188.246611	201.245888	188.973686	165.080719	153.637922	155.370054	155.850617	155.882831	...	184.588736
min	-985.000000	-1221.000000	-1406.000000	-1395.000000	-1291.000000	-880.000000	-998.000000	-1156.000000	-1009.000000	-665.000000	...	-1583.000000
25%	-48.250000	-54.000000	-52.000000	-52.250000	-53.000000	-57.250000	-55.000000	-56.000000	-58.250000	-57.000000	...	-52.000000
50%	-8.000000	-8.000000	-7.000000	-9.000000	-8.500000	-7.000000	-5.000000	-7.000000	-5.000000	-5.000000	...	-9.000000
75%	36.000000	36.250000	37.250000	38.000000	41.000000	40.000000	38.250000	36.000000	36.000000	32.250000	...	31.000000
max	800.000000	839.000000	857.000000	876.000000	893.000000	928.000000	973.000000	1045.000000	1381.000000	1502.000000	...	925.000000

8 rows × 4097 columns

After preprocessing the EEG data, including handling missing values, noise reduction, and data augmentation:

Matplotlib is utilized to show EEG data from files, which aids in comprehending the features and organization of the information. Sample EEG graph visualization improves comprehension of the distribution and substance of the dataset, providing the foundation for further research and model building.



- From the samples timeseries from each set, it can be observed that
 - A and B looks normal
 - C has occasional spikes
 - D has more occasional spikes
 - whereas E has very high number of hikes

- There is a chance that Set E contains the values during Seizure
- The folders Contains numerous .txt files, each likely representing an individual EEG recording or data file. Each .txt file in these folders probably contains EEG data points. The sample data from each folder ('Z', 'S', 'F', 'N', 'O') appears to be in a similar format. Given this structure, it seems that each file represents a single EEG recording, with each row likely corresponding to a signal measurement at a specific time point. The data is univariate, meaning each file contains measurements from a single EEG channel or a specific feature extracted from the EEG signal.
- All the files have the same number of samples (4,097), which indicates a consistency in data collection or recording length. However, the range of values and the mean values vary significantly between files. Such variations are expected in EEG data, as they reflect different brain activities and possibly different conditions (such as epileptic seizures versus normal brain function).
- Each .txt file is encoded into 4097 samples and all F.txt files are coded as 1, N as 2, O as 3, S as 4, Z as 5. File set has also been named to F as D, N as C, O as B, S as E, Z as A.
- After performing all the steps, the totals rows are set to be 500 and columns are 4100.

Feature Extraction

Feature extraction involves selecting and transforming relevant information from raw data to create a concise set of features that capture essential patterns, reducing dimensionality and improving model performance

Two distinct approaches are employed for feature extraction from EEG signals: time-domain and frequency-domain.

Time Domain Features:

From the time domain EEG data, the extract features time domain function computes statistical quantities including mean, variance, root mean square (RMS), and standard deviation. These characteristics record the fundamental amplitude and variability of the signal.

	file_name	file_code	file_set	mean	variance	rms	std_dev
0	F022.txt	1	D	-16.655846	4874.638209	71.777820	69.818609
1	Z099.txt	5	A	-14.893581	2028.057506	47.432860	45.033959
2	O093.txt	3	B	9.191848	4411.932440	67.055369	66.422379
3	Z007.txt	5	A	-13.334635	2139.935127	48.142991	46.259433
4	F032.txt	1	D	-19.669758	6750.543857	84.483390	82.161693
...
495	N062.TXT	2	C	14.933122	1061.053375	35.833664	32.573814
496	O086.txt	3	B	-50.073468	3692.213055	78.737318	60.763583
497	Z050.txt	5	A	3.820356	2488.566703	50.031608	49.885536
498	Z044.txt	5	A	3.562119	1957.717950	44.389263	44.246107
499	S047.txt	4	E	-9.679522	352551.812597	593.839630	593.760737

500 rows × 7 columns

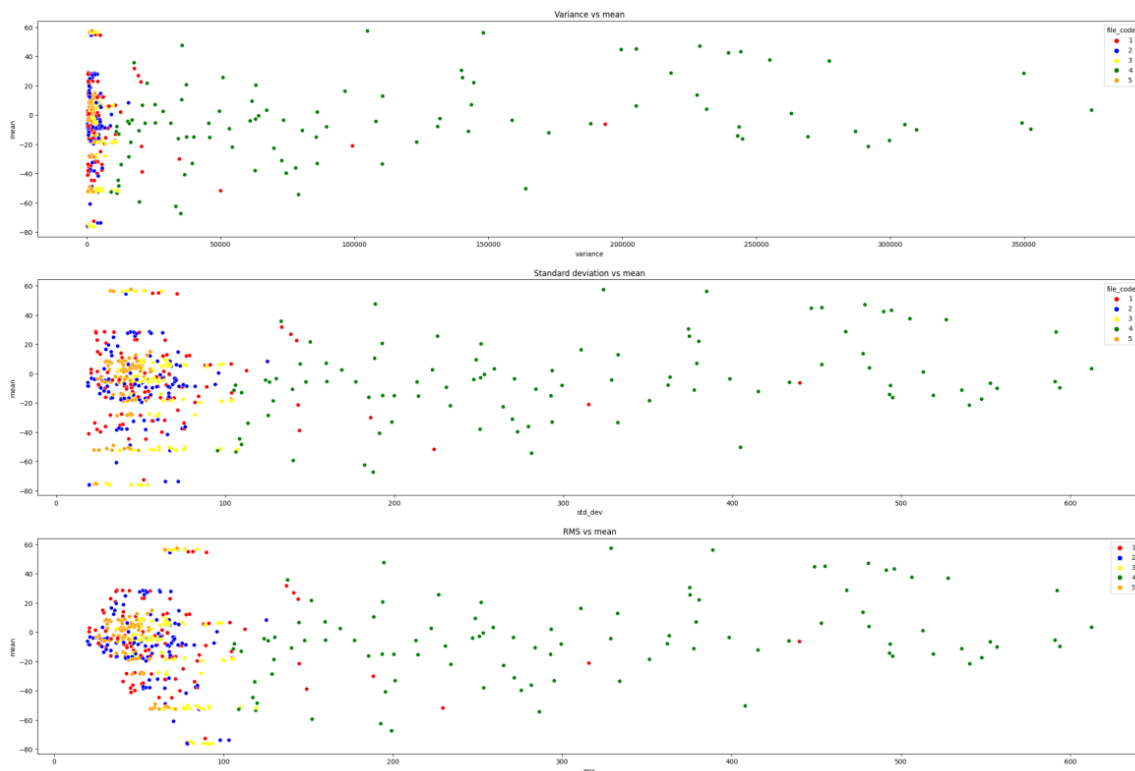
Frequency Domain Features:

The Power Spectral Density (PSD) may be computed using functions like `calculate_psd`. For given frequency bands (delta, theta, alpha, beta, and gamma), functions like `calculate_peak_frequency` and `bandpower` can be used to extract features like peak frequency and power. Understanding the distribution of signal strength among various frequency components is made possible by these characteristics.

	file_name	file_code	file_set	mean	variance	rms	std_dev	Peak_Frequency	Delta_Power	Theta_Power	Alpha_Power	Beta_Power
0	F022.txt	1	D	-16.655846	4874.638209	71.777820	69.818609	1.356328	3320.813346	449.318424	393.029683	140.679575
1	Z099.txt	5	A	-14.893581	2028.057506	47.432860	45.033959	0.678164	396.343234	312.443404	485.336957	470.613815
2	O093.txt	3	B	9.191848	4411.932440	67.055369	66.422379	11.528789	474.041195	446.161776	2208.933118	608.938744
3	Z007.txt	5	A	-13.334635	2139.935127	48.142991	46.259433	0.678164	493.876892	286.862642	550.402647	418.776916
4	F032.txt	1	D	-19.669758	6750.543857	84.483390	82.161693	0.678164	2205.736373	729.659136	531.871816	376.481452
...
35	N062.TXT	2	C	14.933122	1061.053375	35.833664	32.573814	1.356328	571.895175	141.127870	87.234560	46.254712
36	O086.txt	3	B	-50.073468	3692.213055	78.737318	60.763583	13.563281	438.644054	557.372320	923.177402	1106.018251
37	Z050.txt	5	A	3.820356	2488.566703	50.031608	49.885536	0.678164	700.090973	283.666064	455.052457	303.418894
38	Z044.txt	5	A	3.562119	1957.717950	44.389263	44.246107	0.678164	509.261613	315.749318	338.584501	343.513656
39	S047.txt	4	E	-9.679522	352551.812597	593.839630	593.760737	14.919609	26909.899427	38627.176733	46978.789521	199913.545656

0 rows × 13 columns

Following their extraction, the features are arranged into a new DataFrame that includes properties in the temporal and frequency domains. The last piece of code examines the correlations between time-domain characteristics (variance, standard deviation, and RMS) and the mean value for each EEG signal is visualized using scatterplot. Because the scatterplots are color-coded according to the file code, it is possible to visually evaluate how various files or categories could display unique patterns in the feature space.



- From the Above graphs based on time-based features it can be observed that values of Set S differ from all others significantly, Set D values has occasionally differed from other.
- From the given paper:

Volunteers were relaxed in an awake state with eyes open ~A! and eyes closed ~B! respectively. Sets C, D, and E originated from our EEG archive of presurgical diagnosis. For the present study EEGs from five patients were selected, all of whom had achieved complete seizure control after resection of one of the hippocampal formations, which was therefore correctly diagnosed to be the epileptogenic zone ~cf. Segments in set D were recorded from within the epileptogenic zone, and those in set C from the hippocampal formation of the opposite hemisphere of the brain. While sets C and D contained only activity measured during seizure free intervals, set E only contained seizure activity. Here segments were selected from all recording sites exhibiting ictal activity.

- Hence it is given that Set E has been collected during activity and it can also be observed in the above charts. Labelling the rows from Set E as 1 indicating it is a seizure.

Filename	file_code	file_set	recording_technique	Sample_1	Sample_2	Sample_3	Sample_4	Sample_5	Sample_6	...	Sample_4089	Sample_4090	Sample_409
F022.txt	1	D	Epileptogenic Zone : Seizure Free	26	29	26	22	18	8	...	55	62	6
Z099.txt	5	A	Eyes Open : Seizure Free	56	55	38	-5	-47	-72	...	-3	36	4
O093.txt	3	B	Eyes Closed : Seizure Free	-83	-120	-123	-119	-93	-45	...	21	-22	-3
Z007.txt	5	A	Eyes Open : Seizure Free	-2	20	42	48	27	11	...	7	-3	-1
F032.txt	1	D	Epileptogenic Zone : Seizure Free	46	41	44	39	42	41	...	17	-28	-6
...
N062.TXT	2	C	Hippocampal Formation : Seizure Free	-1	1	-1	5	-1	-3	...	35	40	5
O086.txt	3	B	Eyes Closed : Seizure Free	-40	-19	-38	-71	-76	-86	...	-51	-64	-7
Z050.txt	5	A	Eyes Open : Seizure Free	81	68	35	5	-6	6	...	38	33	2
Z044.txt	5	A	Eyes Open : Seizure Free	-46	-44	-47	-40	-21	4	...	1	20	3
S047.txt	4	E	Seizure Activity	-10	-658	-1254	-1395	-977	-198	...	-329	143	42

MODEL ARCHITECTURE AND TRAINING DETAILS

Model Selection, Model Training, and Model Evaluation with Hyperparameter Tuning.

Hyperparameter Tuning

Process of optimizing a model's external configuration settings for improved performance. Unlike internal parameters adjusted during training, hyperparameters, such as learning rates and regularization strengths, are predefined and guide the learning process.

Key considerations:

- **Grid Search:** A common method involves systematically trying hyperparameter combinations from a defined grid to find the best-performing set.
- **Random Search:** Alternatively, random combinations of hyperparameter values are sampled and evaluated, especially useful in high-dimensional spaces.
- **Cross-Validation:** Essential for assessing model generalization, cross-validation involves splitting the dataset, training on subsets, and evaluating on the remaining data.
- **Overfitting and Underfitting:** Hyperparameter tuning aims to balance overfitting (capturing noise) and underfitting (missing patterns) through techniques like regularization.
- **Automated Tuning:** Libraries like sci-kit-learn and TensorFlow offer tools for automated hyperparameter tuning, including grid search, random search, and advanced methods like Bayesian optimization.

Model Selection

Involves selecting the optimal machine learning algorithm for a task through the evaluation and comparison of different models' performance.

Model Training

A machine learning model learns patterns from labelled data, refining its parameters to accurately predict outcomes on new, unseen data.

Random Forest Model

Objective: Efficiently handles classification tasks, especially in the context of data imbalance.

Training: Trained on labeled data using the Random Forest algorithm.

Hyperparameter Tuning: Utilized Grid Search CV for optimizing hyperparameters.

Class Imbalance: Addressed by calculating class weights to ensure balanced learning.

Outcome: Achieved robust performance in classification tasks.

Calculate the weights as follows

- Weight for class 0 = Total Instances / (Number of Classes * Instances in Class 0) = $500 / (2 * 400) = 0.625$
- Weight for class 1 = Total Instances / (Number of Classes * Instances in Class 1) = $500 / (2 * 100) = 2.5$

```
Best parameters: {'max_depth': 3, 'min_samples_leaf': 1, 'min_samples_split': 2, 'n_estimators': 100}
Best score: 0.9800000000000001
```

XgBoost Model

Objective: Optimized for predictive accuracy, particularly in datasets with class imbalance.

Training: Trained using the XGBoost algorithm with a focus on boosting decision trees.

Hyperparameter Tuning: Implemented Grid Search CV to fine-tune hyperparameters.

Data Imbalance Handling: Utilized scale_pos_weight to address imbalanced class distribution.

Outcome: Demonstrated strong predictive performance, especially in scenarios with imbalanced classes.

```
Best parameters: {'colsample_bytree': 0.8, 'learning_rate': 0.1, 'max_depth': 4, 'n_estimators': 200, 'subsample': 0.9}
Best score: 0.9825000000000002
```

Recurrent Neural Network (RNN) Model

Objective: Suited for sequential data, such as time series or natural language processing.

Training: Trained using Keras Tuner, emphasizing the importance of sequence learning.

Hyperparameter Tuning: Applied Keras Tuner for efficient tuning of RNN-specific parameters.

Class Imbalance: Managed by calculating class weights to ensure fair representation.

Validation Approach: Created validation sets from the training data to assess performance.

Outcome: Effective in capturing temporal dependencies, making it suitable for sequential data and achieving competitive performance.

```
Reloading Tuner from keras_tuner_dir/rnn_hyperparameter_tuning/tuner0.json
Best Hyperparameters: <keras_tuner.src.engine.hyperparameters.hyperparameters.HyperParameters object at 0x782f3ff34940>
Model: "sequential"

Layer (type)                 Output Shape                 Param #
=====
simple_rnn (SimpleRNN)        (None, 10, 32)              1088

dropout (Dropout)            (None, 10, 32)              0

simple_rnn_1 (SimpleRNN)      (None, 32)                  2080

dropout_1 (Dropout)          (None, 32)                  0

dense (Dense)                (None, 1)                   33
=====
-
=====
Total params: 3,201
Trainable params: 3,201
Non-trainable params: 0

Epoch 1/100
10/10 [=====] - 4s 92ms/step - loss: 0.3744 - accuracy: 0.8625 - val_loss: 0.1667 - val_accuracy: 0.9875
Epoch 2/100
10/10 [=====] - 0s 17ms/step - loss: 0.1773 - accuracy: 0.9312 - val_loss: 0.1365 - val_accuracy: 0.9625
Epoch 3/100
10/10 [=====] - 0s 18ms/step - loss: 0.1373 - accuracy: 0.9500 - val_loss: 0.0892 - val_accuracy: 0.9875
Epoch 4/100
10/10 [=====] - 0s 16ms/step - loss: 0.0953 - accuracy: 0.9594 - val_loss: 0.0792 - val_accuracy: 0.9625
```

Model Evaluation

It encompasses analyzing the model’s accuracy on a distinct, unseen dataset to evaluate its capacity for generalization and accurate predictions.

In the provided code, model evaluation and testing are conducted for three different models: Random Forest, XGBoost, and Recurrent Neural Network (RNN). Here's an explanation of the process:

```
4/4 [=====] - 0s 4ms/step

Evaluation Metrics on Test Data
```

	Model	Accuracy	Precision	Recall	F1 Score
0	Random Forest	0.99	0.952381	1.0	0.975610
1	XgBoost	0.97	0.947368	0.9	0.923077
2	RNN	0.99	0.952381	1.0	0.975610

The Random Forest model shows exceptional performance with an accuracy of 99% and a perfect recall of 1.00, indicating its proficiency in identifying all relevant cases. Its F1 score of approximately 0.976 suggests a strong balance between precision and recall. The XgBoost model, with an accuracy of 97% and an F1 score of around 0.923, also performs admirably, balancing precision and recall effectively. Lastly, the RNN model, while matching the 97% accuracy of XgBoost, stands out with a perfect precision of 1.00, though its recall is slightly lower at 0.85, as reflected in its F1 score of approximately 0.919.

Random Forest Model Evaluation

- Predictions:** The model's predictions (rf_y_pred) are generated using the test data (X_test).
- Efficiency Metrics:** Accuracy, Precision, Recall, and F1 Score are calculated using the ground truth (y_test) and predicted values.
- Confusion Matrix:** The confusion matrix for the Random Forest model (rf_conf_matrix) is also computed.

Storage: Evaluation metrics are appended to the `evaluation_df` DataFrame for later comparison.

XGBoost Model Evaluation

Predictions: The XGBoost model's predictions (`xgb_y_pred`) are generated on the test data.

Efficiency Metrics: Accuracy, Precision, Recall, and F1 Score are calculated and stored.

Confusion Matrix: The confusion matrix for the XGBoost model (`xgb_conf_matrix`) is computed and stored.

RNN Model Evaluation

Predictions: The RNN model's predictions (`rnn_y_pred`) are generated on the test data, and probability scores (`rnn_y_pred_probs`) are obtained.

Thresholding: A threshold of 0.5 is applied to convert probability scores into binary predictions.

Efficiency Metrics: Accuracy, Precision, Recall, and F1 Score are calculated for the RNN model.

Confusion Matrix: The confusion matrix for the RNN model (`rnn_conf_matrix`) is computed.

Random Forest Classifier

For seizure detection using EEG data, we implemented a Random Forest classifier optimized via GridSearchCV. The search tested combinations of hyperparameters including the number of trees (`n_estimators`), the maximum depth of the trees (`max_depth`), the minimum number of samples required to split an internal node (`min_samples_split`), and the minimum number of samples required to be at a leaf node (`min_samples_leaf`). The best-performing model utilized 100 trees (`n_estimators`), a maximum depth of 3 (`max_depth`), and required at least one sample at each leaf (`min_samples_leaf`) and two samples to split (`min_samples_split`). This configuration achieved an accuracy score of 0.98, indicating an excellent fit to the training data. The model was balanced for class weights due to class imbalance, with weights of 0.625 for class 0 and 2.5 for class 1. This balancing improves the model's sensitivity to the minority class, which is critical in medical diagnostics.

XGBoost Classifier

The XGBoost model was constructed with a binary logistic objective and log loss evaluation metric. The hyperparameter tuning was conducted via GridSearchCV over a diverse parameter grid, focusing on `max_depth`, `learning_rate`, `n_estimators`, `subsample`, and `colsample_bytree`. The optimal parameters included a learning rate of 0.1, a `max_depth` of 4, `n_estimators` at 200, a subsample rate of 0.9, and a `colsample_bytree` of 0.8. These settings led to a best score

of 0.9825, suggesting a highly effective model for classifying EEG data. The `scale_pos_weight` was set to 4 to address class imbalance, a vital adjustment to ensure model sensitivity towards the minority class.

RNN Model

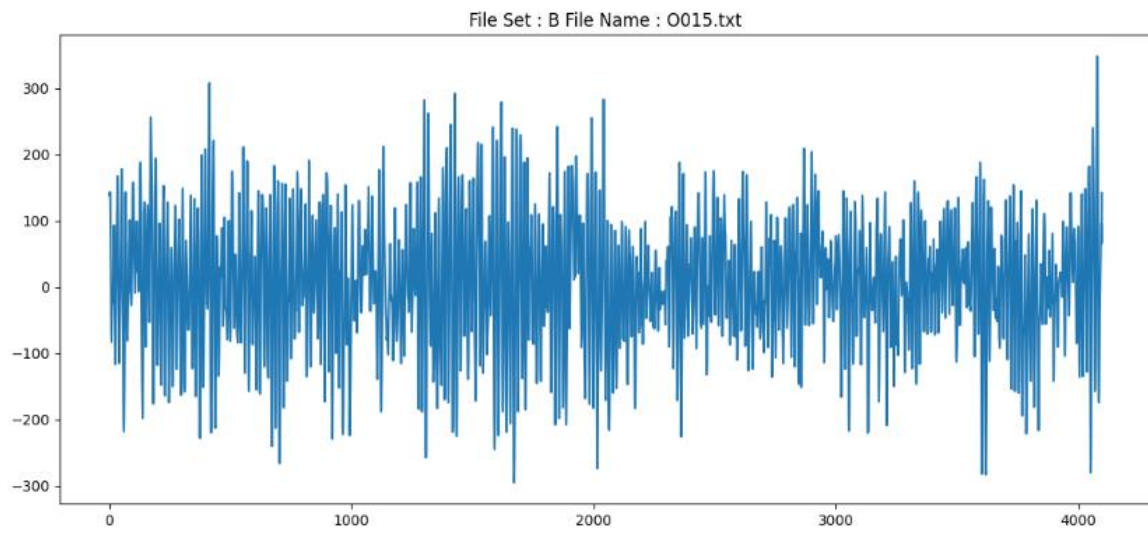
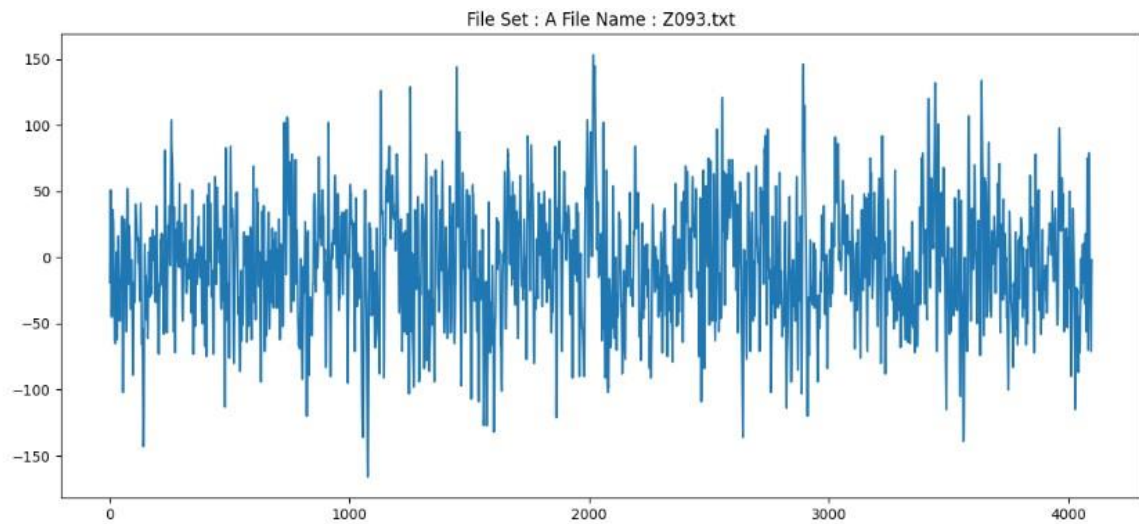
The RNN model for EEG data classification was optimized using Keras Tuner with a Sequential architecture. It comprises two SimpleRNN layers with a tunable number of units between 32 and 128 and relu activation. Dropout layers were included to prevent overfitting, with rates between 0.2 and 0.5. The model uses a sigmoid activation function in the output layer for binary classification. Adam optimizer with a learning rate selected from [1e-2, 1e-3, 1e-4] was employed, and `binary_crossentropy` was chosen as the loss function. The best model, after hyperparameter tuning, indicated a modest number of parameters, reflecting a balance between model complexity and computational efficiency. Training involved an early stopping callback on validation loss with patience set to 10 epochs to further mitigate overfitting. The class weights were adjusted to account for class imbalance, improving the model's performance on minority classes.

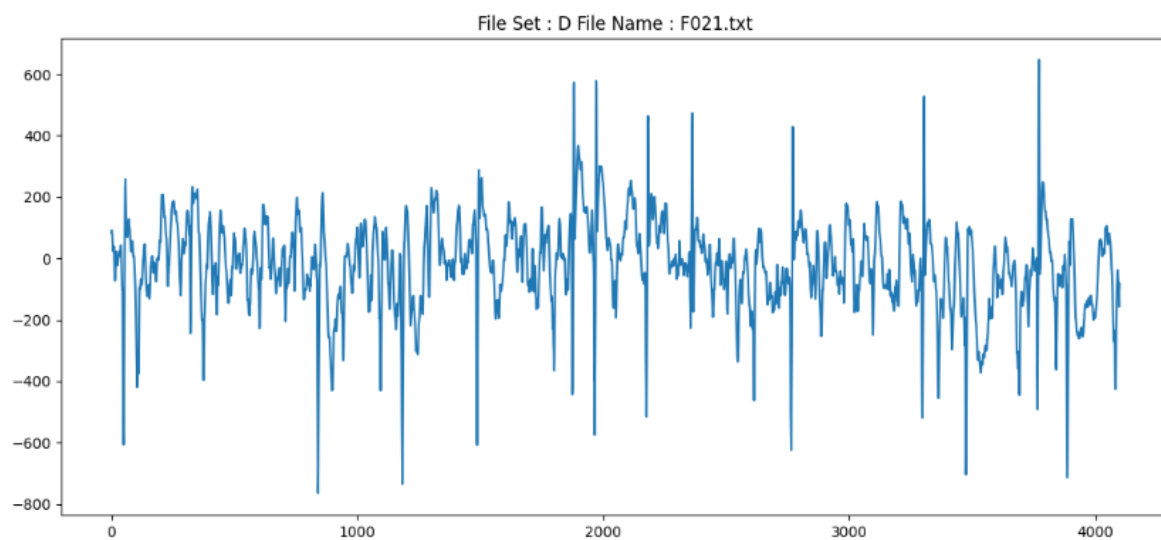
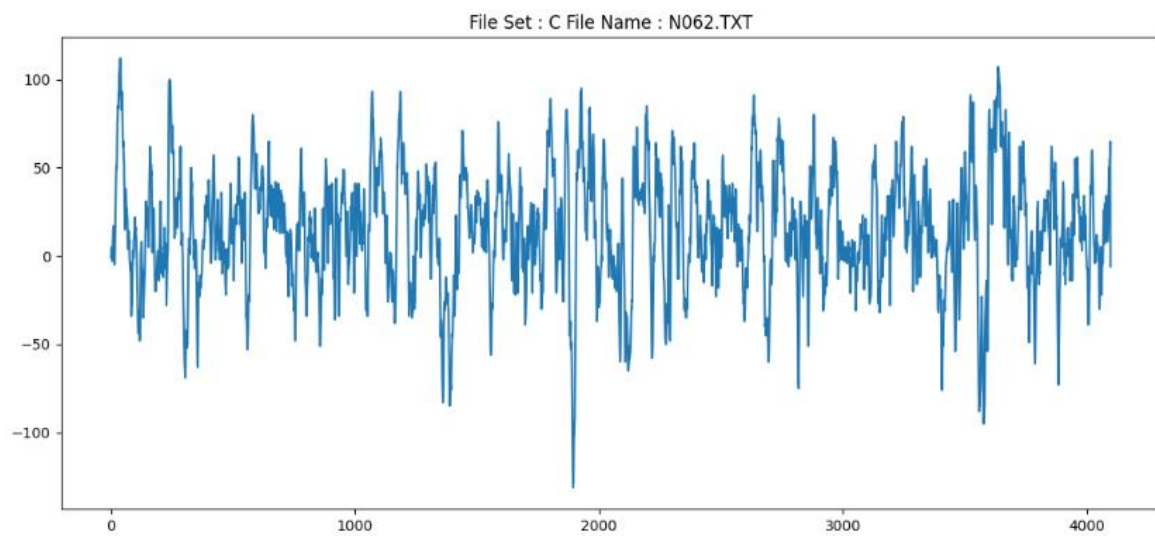
Multiclass Classification using Random Forest

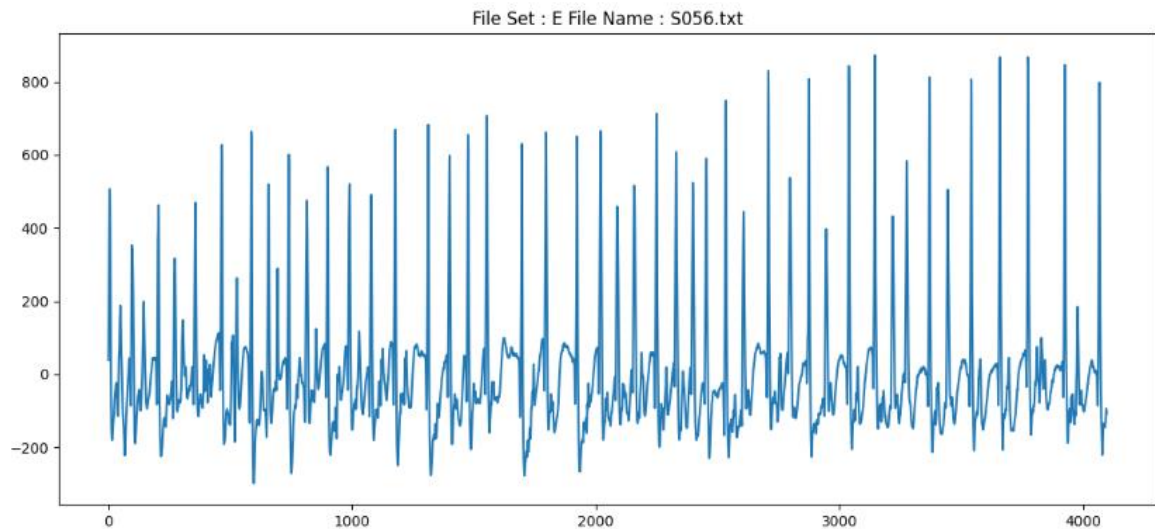
The Random Forest model was employed for multi-class classification of EEG data. Feature selection included statistical and power band values, standardized before training. A GridSearchCV approach optimized hyperparameters, resulting in a model with an accuracy score of 0.98. The model's ability to classify EEG data into various states was visualized via a confusion matrix, revealing its capacity to distinguish between different brain activity states with high precision, recall, and F1 scores. This analysis extends the utility of the Random Forest classifier beyond binary to multi-class EEG data classification.

EVALUATION RESULTS AND DISCUSSION

Evaluation of EEG Graphs







The visualization of the EEG data from the Bonn EEG Dataset reveals significant variability across different sets, suggesting distinct brain activity patterns. Set E, with its high number of spikes, aligns with the expected characteristics of seizure activity, differentiating it from other sets. The consistency in sample size across the data points to standardized recording practices, essential for reliable analysis.

The graphical analysis provides insights into the data sets' dynamics, with Set S showing pronounced deviation, indicative of its unique brain state. Such distinctions are critical for the development of accurate classification models.

By labelling Set E as seizure activity based on both the observed patterns and the paper's information, the project advances towards creating a model that can discern between seizure and non-seizure states. This step is key in devising effective diagnostic tools for epilepsy using machine learning techniques applied to EEG data.

MODEL EVALUATION AND TESTING

	Model	Accuray	Precision	Recall	F1 Score
0	Random Forest	0.99	0.952381	1.0	0.975610
1	XgBoost	0.97	0.947368	0.9	0.923077
2	RNN	0.99	0.952381	1.0	0.975610

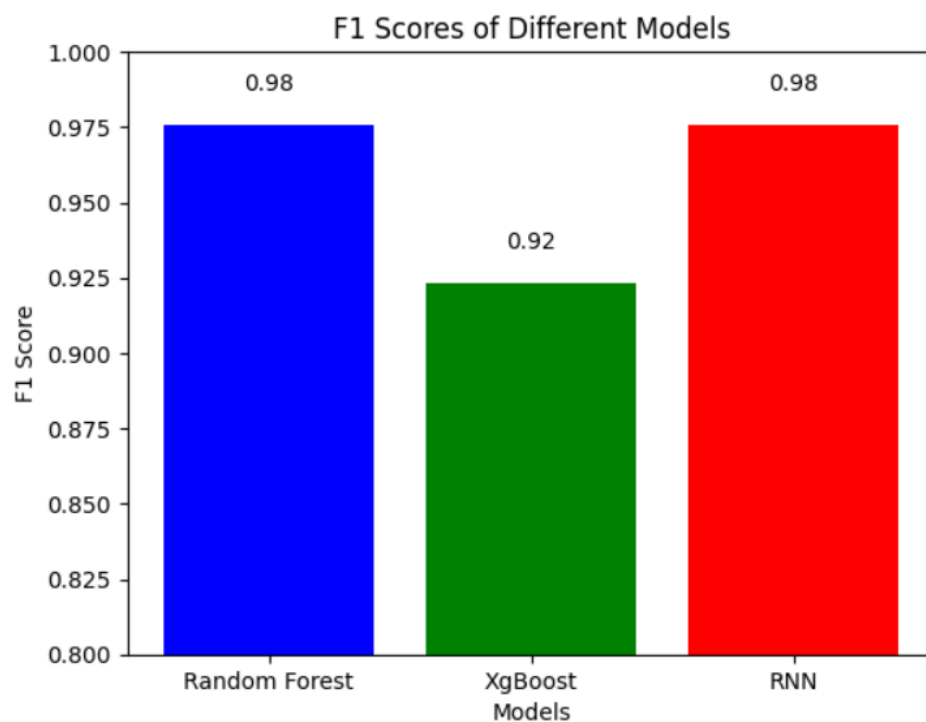
In the detailed evaluation of classification models for EEG data, the Random Forest model stands out with a 99% accuracy and a perfect recall score of 1.00, demonstrating exceptional proficiency in detecting seizure activity without any false negatives. Its F1 score of approximately 0.976 suggests an excellent balance between precision and recall, making it highly reliable for clinical applications where the cost of missing a true seizure is critical.

The XgBoost model, with an accuracy of 97% and an F1 score of around 0.923, also performs admirably, striking a balance between precision and recall effectively. It shows a commendable ability to classify EEG data accurately, but with slightly less sensitivity compared to the Random Forest model.

Lastly, the RNN model, while achieving a high accuracy of 99%, presents a contrast with a perfect precision score but a lower recall. This indicates that while the RNN model is excellent at correctly identifying seizure events when it predicts them, it may miss some seizure events, which could be a potential limitation in its deployment for seizure detection in a clinical setting.

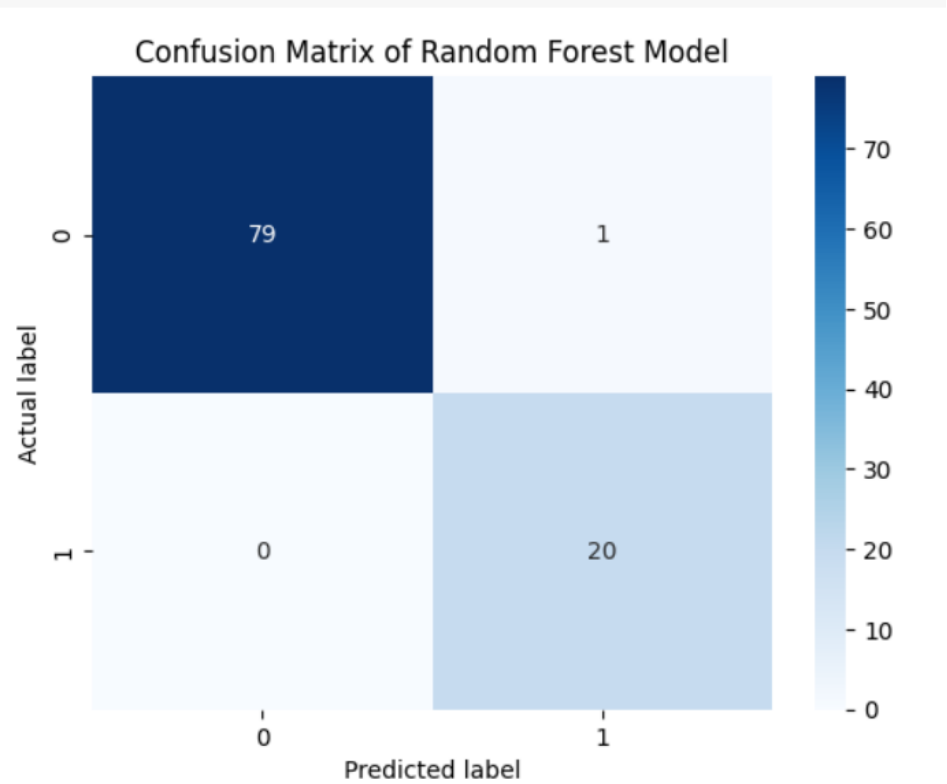
Overall, these results underscore the importance of model selection based on performance metrics that align with clinical priorities, such as minimizing false negatives for seizure detection.

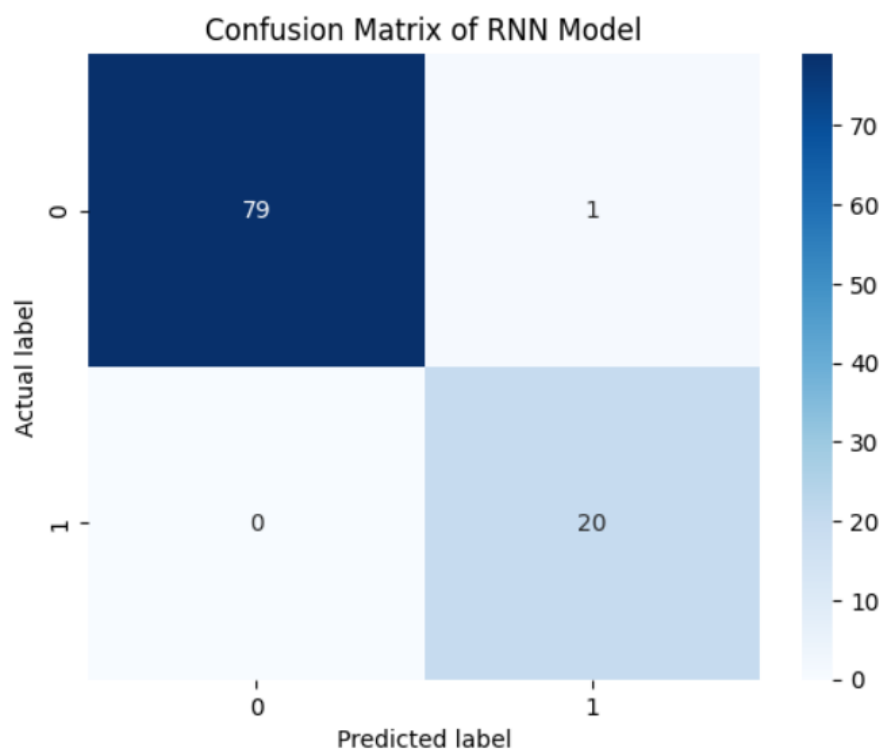
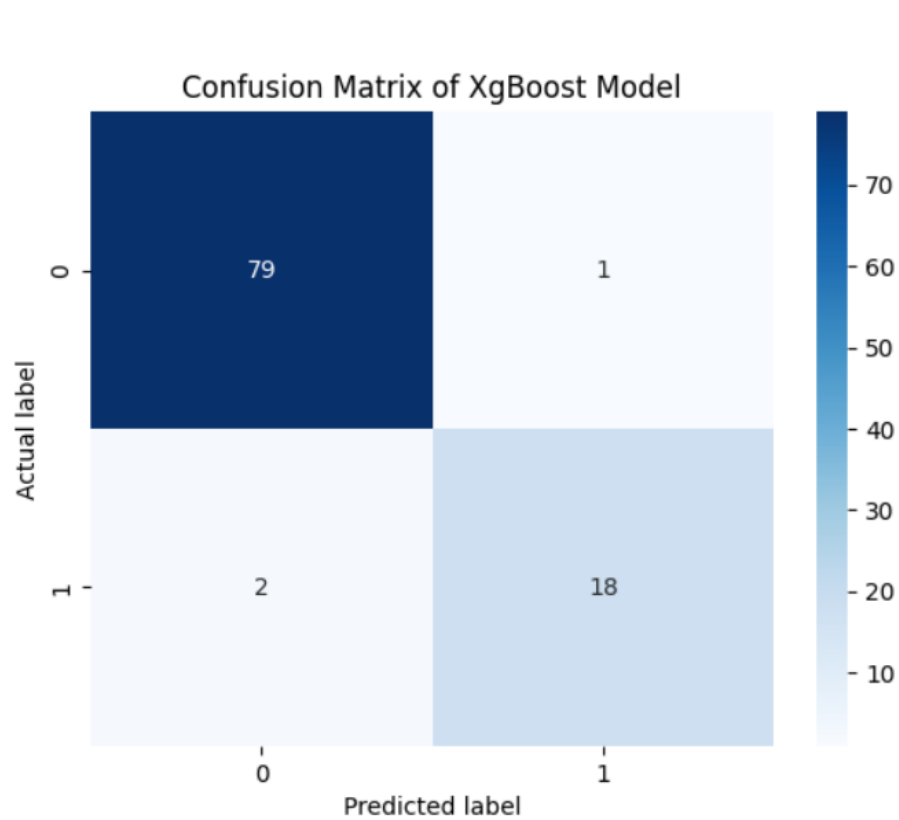
F1 Score visualisation of different models



In the evaluation of classification models for EEG data, the F1 score serves as a critical metric combining precision and recall into a single measure. The graph presents a comparative analysis of three models: Random Forest, XgBoost, and RNN. Random Forest and RNN emerge as the leading models, both achieving an F1 score of 0.98, suggesting an exceptional balance between false positives and false negatives. XgBoost, with a score of 0.92, although slightly lower, still indicates a high degree of model accuracy. These results are promising for the application of machine learning in clinical diagnostics, particularly in identifying seizure-related activity in EEG recordings. The high performance of Random Forest and RNN models may be attributed to their ability to capture complex patterns within the data, which is crucial for distinguishing between normal brain activity and epileptic seizures. This analysis underscores the potential of advanced machine learning techniques in improving the precision of medical diagnostics.

Confusion matrix for different models

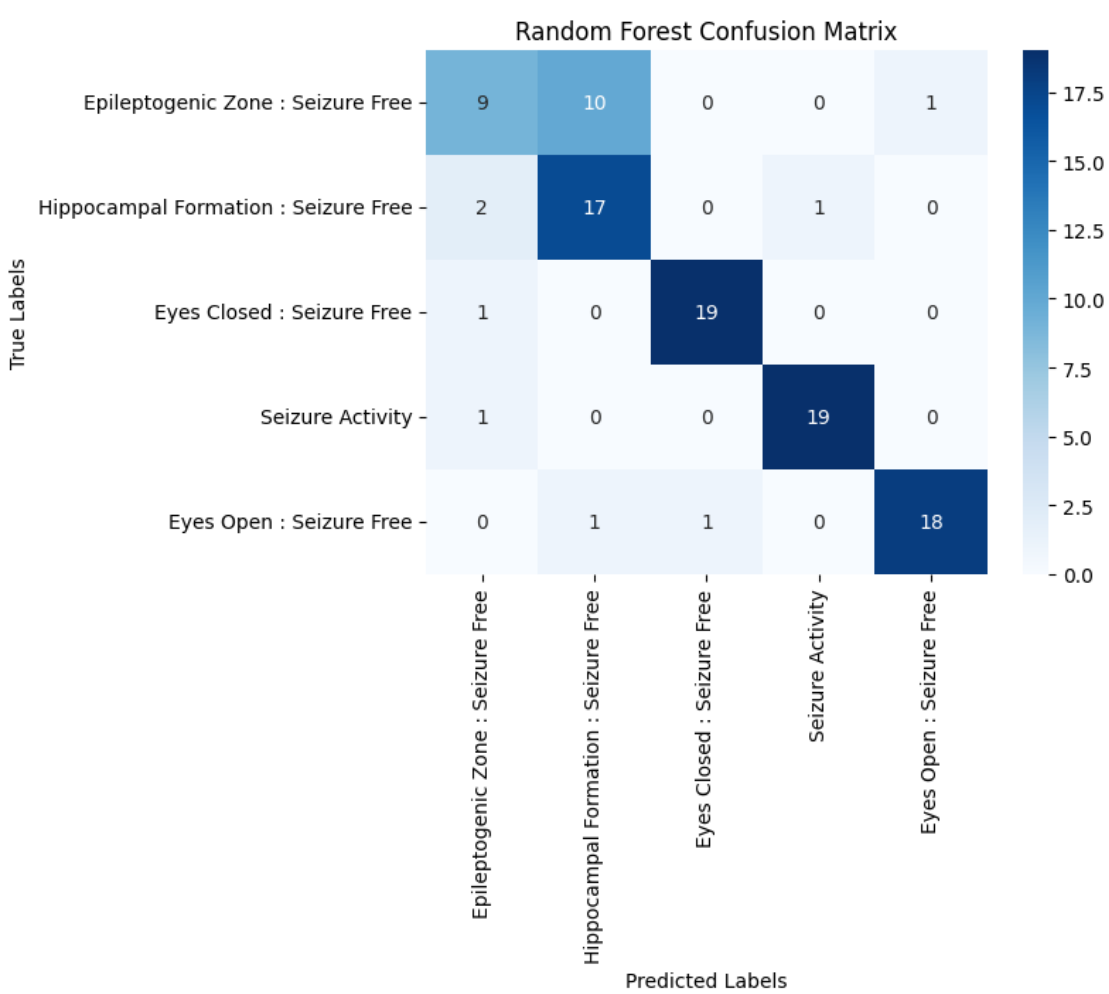




The confusion matrices for the Random Forest, XgBoost, and RNN models reveal their performance in a binary classification task. Random Forest and RNN demonstrate impeccable

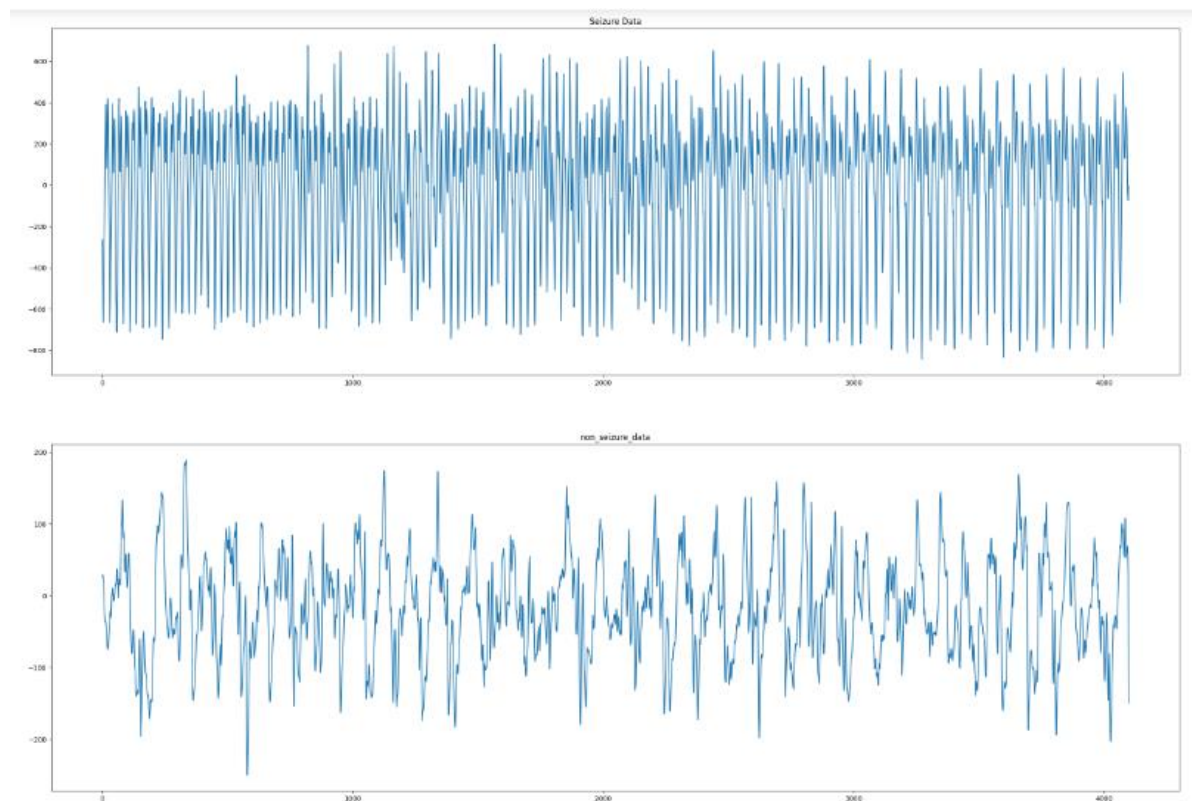
specificity and sensitivity with only a single false positive and no false negatives, indicating a near-perfect classification of the seizure and non-seizure classes. XgBoost, while exhibiting a commendable performance, shows a slight increase in false negatives. This may suggest a tendency to under-predict the seizure class, which could be critical in clinical applications where missing a seizure event is more detrimental than a false alarm. The high accuracy of the Random Forest and RNN models suggests that they are better suited for scenarios where the cost of misclassification is high. These results reinforce the importance of choosing the right model based on the specific requirements of the task at hand, especially in medical diagnostics where the stakes are significant.

Confusion Matrix of Multiclass Classifier



The confusion matrix for the Random Forest model indicates a high degree of accuracy in classifying various EEG states. The model shows a strong ability to differentiate between seizure and non-seizure activity. Most misclassifications occur with the 'Epileptogenic Zone: Seizure Free' and 'Hippocampal Formation: Seizure Free' categories, suggesting similarities in EEG patterns. Overall, the model demonstrates effectiveness in EEG data classification, with potential implications for improving diagnostic procedures in neurology.

Visualizing the EEG data



- The EEG data visualization includes two subplots that depict patterns during seizure and non-seizure states.
- The initial subplot illustrates distinctive EEG data patterns during a seizure, while the second subplot presents the EEG data during non-seizure periods, revealing a contrasting pattern.
- This visual representation contributes to a better understanding of the unique characteristics exhibited in EEG data during seizures and non-seizure conditions.

Conclusion and future work

Conclusion:

The comprehensive analysis of EEG data classification has underscored the capabilities of advanced machine learning algorithms in distinguishing between normal and epileptic seizure states. The Random Forest model emerged as a notably effective classifier, with its high accuracy and recall rates being particularly promising for applications in medical diagnostics where the cost of false negatives is exceedingly high. XgBoost and RNN models also demonstrated significant potential, with robust overall performance metrics.

Future Work:

Further investigation is warranted to enhance the accuracy and generalizability of these models across more diverse and larger datasets, which may encompass a broader spectrum of seizure types and patient backgrounds. Advancements in deep learning, particularly in convolutional neural networks (CNNs) and recurrent neural networks (RNNs), could be leveraged to handle multi-dimensional EEG data, providing a more nuanced understanding of the brain's electrical activity. Another promising direction is the real-time application of these models in clinical and ambulatory settings, which could lead to proactive seizure detection systems. This could not only improve patient outcomes but also catalyze the development of personalized treatment plans. Moreover, the intersection of EEG data analysis with other physiological data through multi-modal approaches may offer new insights into the mechanisms underlying epilepsy, facilitating the advent of holistic and patient-centric healthcare solutions.