IE 6400 – FOUNDATIONS OF DATA ANALYTICS
SUSHMITHA SUDHARSAN (+1 857 565 8800)
HARINI PRASAD VASISHT (+1 984 374 4836)
TANMAYI SHURPALI (+1 848 205 5511)

Project 3 Report
EEG Signal Analysis and Classification

1. Introduction

This project focuses on the analysis and classification of EEG signals to distinguish between different brain states. Using advanced preprocessing techniques, signal visualizations, and machine learning models, the goal is to identify patterns and anomalies in the EEG data for effective classification and diagnosis.

---

2. Project Workflow

2.1 Data Acquisition and Inspection

- **Data Source:** EEG signal datasets from the Bonn University repository.
- **Steps Taken:**
    - Loaded EEG data from multiple .txt files stored in directory hierarchies.
    - Used Python libraries like numpy and pandas to handle the data efficiently.
    - Checked for missing values and anomalies in the dataset.

2.2 Data Preprocessing

- Steps included:
    - Conversion of EEG signals into structured arrays for analysis.
    - Plotted signals to observe patterns and ensure data integrity.
    - Applied noise reduction techniques if necessary.

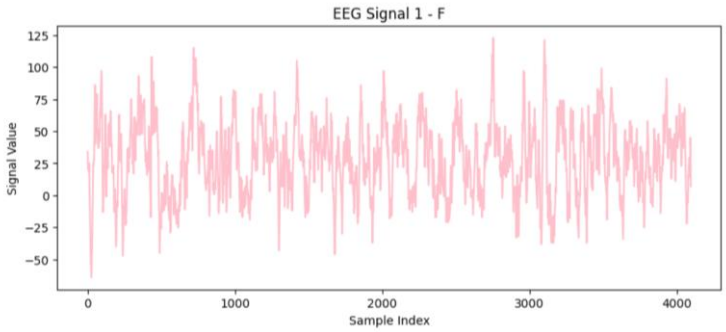
2.3 Exploratory Data Analysis (EDA)

- Visualized EEG signals using line plots to understand variations.
- Observed characteristics of signals across different brain states.

2.4 Analysing the Dataset:

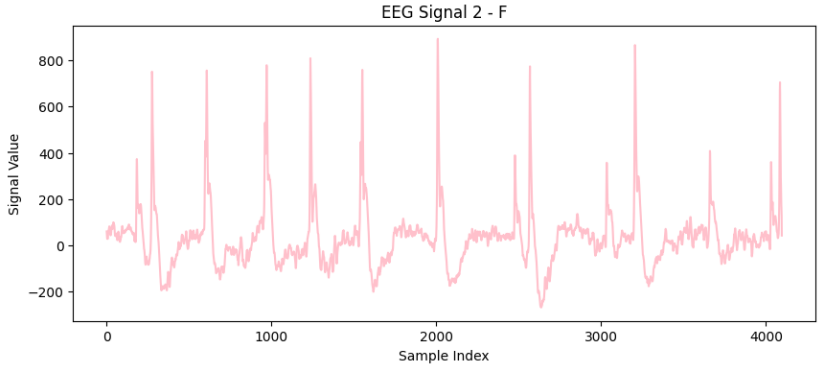
Interpretation of the EEG Signal Graphs

1. **Signal 1:**



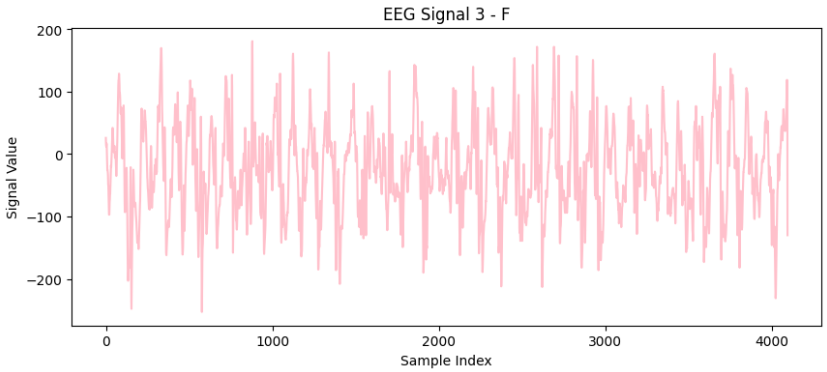
   - The signal exhibits moderate variability, with spikes ranging between approximately -50 to +100.
   - This suggests activity with no significant anomalies or consistent patterns of oscillations.
   - Interpretation: Likely normal brain activity, potentially from a resting or baseline state.

2. **Signal 2:**



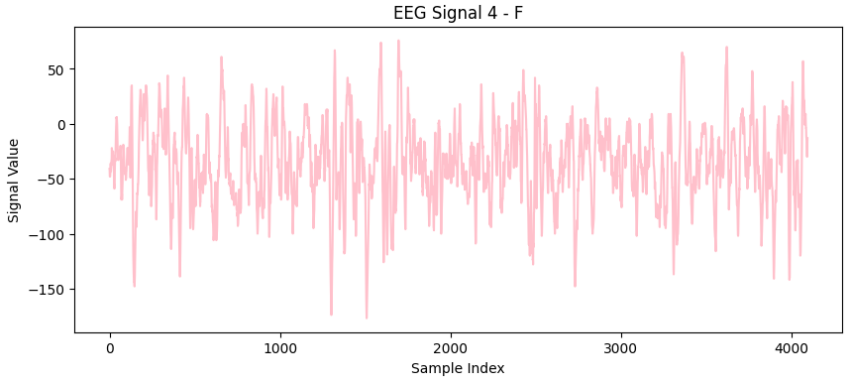
   - The graph shows large, periodic spikes (up to +800 and below -200), indicating strong oscillatory activity.
   - Such a pattern could represent seizure activity or other intense neural events.
   - Interpretation: Abnormal signal, likely indicative of a high-arousal or pathological state.

3. **Signal 3:**



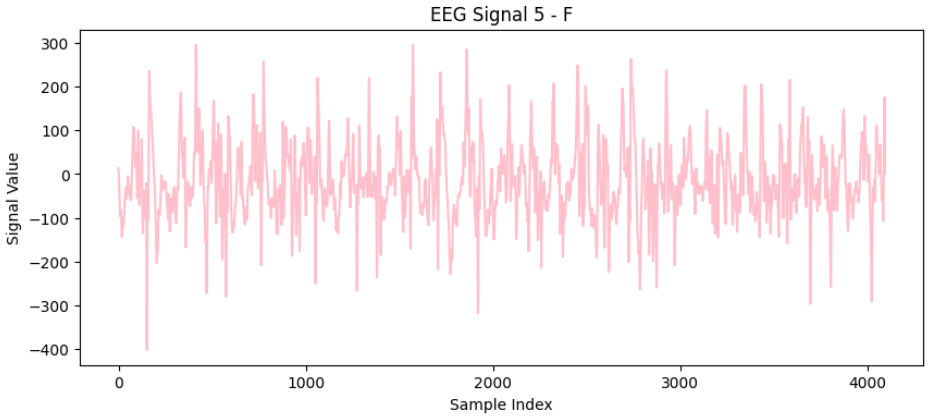
- The variability in this signal is more scattered, with peaks around -200 and +200 but lacking periodicity.
- There might be some level of irregular brain activity but without a clear, structured pattern.
- Interpretation: Could indicate mild abnormality or transitional neural states.

4. **Signal 4:**



   - This signal exhibits lower variability compared to Signal 3, with peaks staying below ±100.
   - The pattern is more stable but lacks distinctive features.
   - Interpretation: Likely a resting state or a minimally active brain region.

5. **Signal 5:**



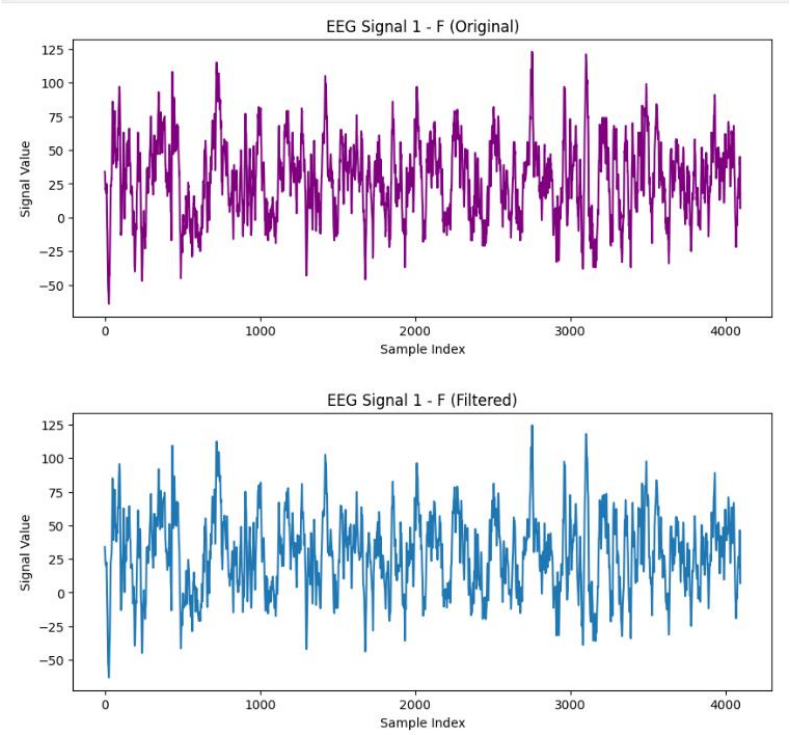
   - The variability is higher, with peaks extending from -400 to +300.
   - There are sharp oscillations, indicating potentially heightened activity.
   - Interpretation: Could suggest a state of alertness or engagement, or mild irregularity.

Key Insights:

- **Normal vs Abnormal:** Signals 1 and 4 appear closer to normal activity patterns, while Signal 2 likely represents a pathological state (e.g., seizure).
- **Variability:** Signals 3 and 5 exhibit moderate variability, which could correspond to states of mild abnormality or heightened activity.
- **Next Steps:** These insights can be further validated using statistical measures (e.g., mean, standard deviation) and classification algorithms to confirm the state (normal vs abnormal) of each signal.

Interpretation of Cleaned EEG Signal

Signal 1:

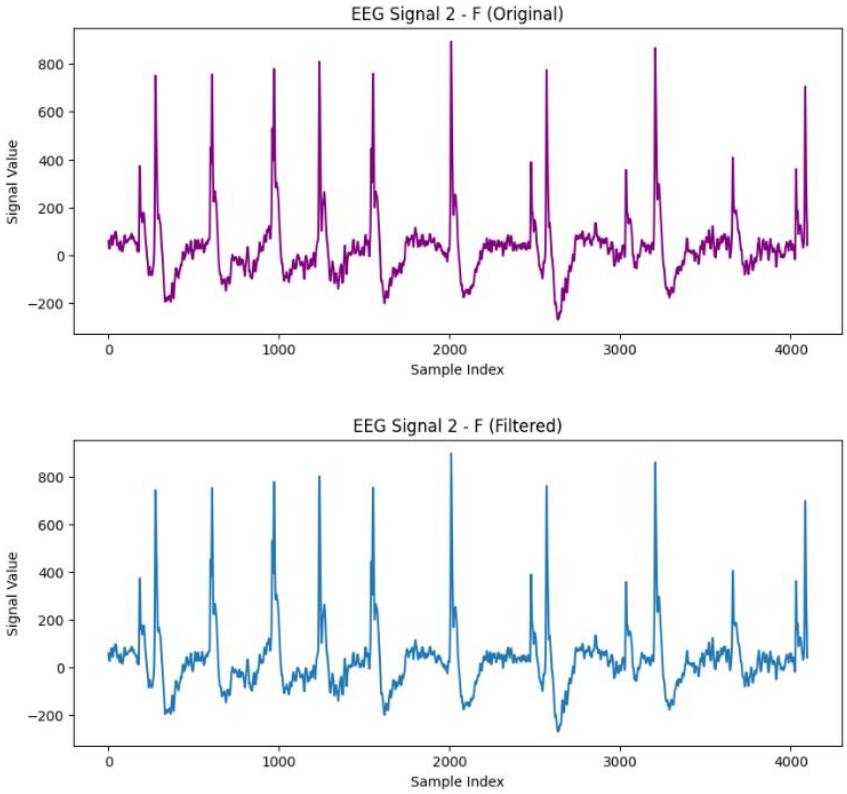


1. **Original EEG Signal:**
    - The signal contains high variability with visible noise or irregular spikes across the sample indices.
    - Noise in this signal could be caused by external interference or artifacts from the EEG recording process (e.g., muscle movements, electrical noise).
    - While some patterns are discernible, the high-frequency noise obscures finer details.
2. **Filtered EEG Signal:**
    - After filtering, the signal is smoother and cleaner.
    - High-frequency noise and irregularities have been removed, making the underlying pattern clearer.
    - The main characteristics of the EEG signal, such as significant peaks and valleys, are retained, but they are more defined due to the noise reduction.

Key Observations:

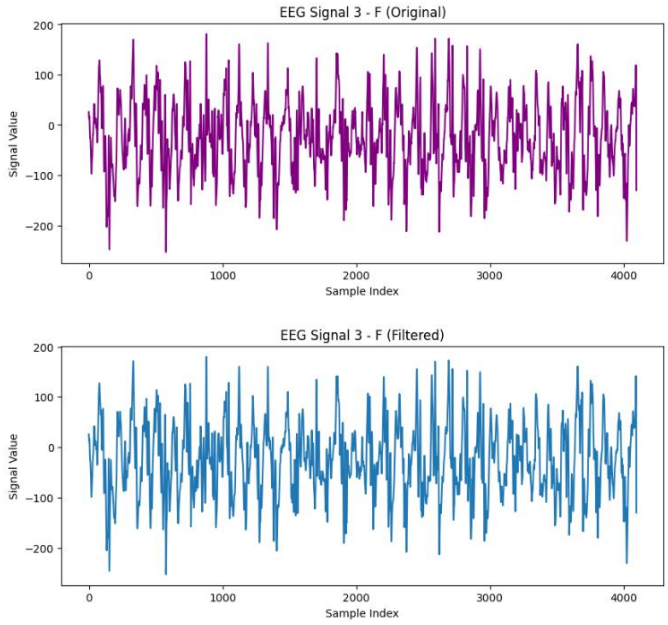
- **Noise Reduction:** The filtering process effectively removes unwanted high-frequency components while preserving the core signal characteristics.
- **Improved Analysis:** The filtered signal is more suitable for further analysis, such as feature extraction or classification, as it focuses on the meaningful variations in the EEG activity.

Signal 2:



1. **Original Signal :**
    - The signal demonstrates strong periodic spikes, reaching up to 800, indicating significant oscillatory activity.
    - Noise or irregularities are visible in the baseline, which could obscure finer details in the low-amplitude regions.
2. **Filtered Signal:**
    - The filtering process removes high-frequency noise while preserving the prominent periodic spikes.
    - The signal is cleaner, with a more consistent baseline and defined oscillations.
3. **Key Insight:**
    - The periodic spikes suggest specific neural events, possibly indicative of seizures or rhythmic brain activity.
    - Filtering enhances the signal's clarity, making it more suitable for classification or diagnosis.

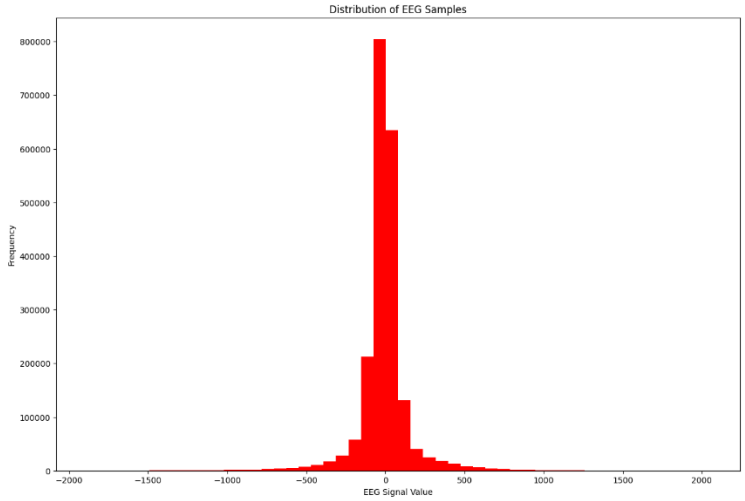
EEG Signal 3



1. **Original Signal :**
    - This signal exhibits high variability with peaks between ±200, but there is no evident periodicity.
    - The presence of noise makes it challenging to discern underlying patterns.
2. **Filtered Signal :**
    - The filtering process has smoothed out irregular noise, highlighting the actual variations in the signal.
    - Despite the noise removal, the signal still lacks clear periodic features.
3. **Key Insight:**
    - The signal likely represents irregular brain activity without distinct patterns. It may correspond to a transitional neural state or mild abnormality.
    - The filtered version is more appropriate for analyzing subtle trends or changes.

Overall Observations:

- **Signal 2:** The filtered version is highly indicative of structured activity and is ideal for further analysis of periodic neural events.
- **Signal 3:** Filtering removes unnecessary noise, making it possible to focus on the signal's general variability. Additional analysis might be needed to determine its significance.



Interpretation of the EEG Signal Value Distribution:

1. **Shape of the Distribution:**
    - The histogram shows a sharp peak centered around 0, indicating that the majority of EEG signal values are close to zero.
    - The distribution tapers symmetrically on both sides, suggesting a normal-like distribution with some variation in signal amplitude.
2. **Frequency Range:**
    - The highest frequency corresponds to signal values near 0, suggesting that the baseline or resting state of the EEG signal is dominant.
    - Lower frequencies occur at extreme values (e.g., ±1000 to ±2000), representing occasional outliers or significant deviations in the signal.
3. **Potential Outliers:**
    - The presence of values far from the center (e.g., ±2000) might indicate sporadic events such as noise, artifacts, or high-amplitude neural activity like seizures.
4. **Insights:**
    - The symmetric distribution suggests the data is well-balanced without significant bias.
    - High-amplitude values (outliers) may need further investigation to determine if they are biologically relevant or artifacts.

3. Analytical Techniques and Models

3.1 Signal Preprocessing

- Applied filters to remove noise and irrelevant frequencies.
- Extracted features from the EEG data using statistical and signal processing methods.

3.2 Model Classification

This code implements two deep learning models for EEG signal classification: 1D CNN and LSTM.

1. Data Preparation:
    - The EEG data is reshaped to meet the input requirements of the models, ensuring compatibility with their 3D input format.
2. 1D CNN Model:
    - Extracts spatial features from the EEG signal using convolution and pooling layers.
    - Fully connected layers are used for binary classification (e.g., normal vs abnormal).
    - The model is trained using the training dataset, validated on a separate set, and tested for accuracy on unseen data.
3. LSTM Model:
    - Designed to capture temporal dependencies in the EEG data by processing sequential information.
    - Reshaped data is used to train and validate the model.
4. Evaluation:
    - The CNN model's accuracy is calculated after making predictions on the test set.
    - The LSTM model follows a similar approach for training and evaluation.

3.2.1 Model Training

Model Training of CNN Model

```
Epoch 1/10
c:\Users\dhanu\AppData\Local\Programs\Python\Python312\Lib\site-packages\keras\src\layers\convolutional\base_conv.py:107:
  super().__init__(activity_regularizer=activity_regularizer, **kwargs)
10/10 ━━━━━━━━━━━━━━━━━━━━ 1s 60ms/step - accuracy: 0.5745 - loss: 646.3426 - val_accuracy: 0.4000 - val_loss: 0.6933
Epoch 2/10
10/10 ━━━━━━━━━━━━━━━━━━━━ 1s 50ms/step - accuracy: 0.4128 - loss: 0.6855 - val_accuracy: 0.4000 - val_loss: 0.6935
Epoch 3/10
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step - accuracy: 0.3911 - loss: 0.6934 - val_accuracy: 0.4000 - val_loss: 0.6932
Epoch 4/10
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 42ms/step - accuracy: 0.4573 - loss: 0.6932 - val_accuracy: 0.6000 - val_loss: 0.6929
Epoch 5/10
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step - accuracy: 0.6188 - loss: 0.6928 - val_accuracy: 0.6000 - val_loss: 0.6925
Epoch 6/10
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 43ms/step - accuracy: 0.6301 - loss: 0.6921 - val_accuracy: 0.6000 - val_loss: 0.6919
Epoch 7/10
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step - accuracy: 0.5963 - loss: 0.6919 - val_accuracy: 0.6000 - val_loss: 0.6915
Epoch 8/10
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 41ms/step - accuracy: 0.6200 - loss: 0.6910 - val_accuracy: 0.6000 - val_loss: 0.6910
Epoch 9/10
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 42ms/step - accuracy: 0.5963 - loss: 0.6909 - val_accuracy: 0.6000 - val_loss: 0.6905
Epoch 10/10
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 43ms/step - accuracy: 0.6090 - loss: 0.6901 - val_accuracy: 0.6000 - val_loss: 0.6900
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 20ms/step
Accuracy of CNN model: 0.6
```

The CNN model shows inconsistent training, with accuracy fluctuating and reaching 60% by the final epoch. Validation accuracy remains stagnant between 40-60%, indicating poor generalization to unseen data. The stable validation loss suggests the model may be overfitting or underperforming, requiring tuning or a more robust architecture for improvement.

Model Training of LSTM Model

MODEL TRAINING

```
#5. MODEL TRAINING:

# Compiling and train the LSTM model
lstm_model.compile(optimizer='adam', loss='binary_crossentropy', metrics=['accuracy'])
lstm_model.fit(X_train_lstm, y_train, epochs=10, batch_size=32, validation_data=(X_val_lstm, y_val))

# Predicting on the test set
y_pred_lstm_probs = lstm_model.predict(X_test_lstm)
y_pred_lstm = (y_pred_lstm_probs > 0.5).astype(int)

# Evaluating the LSTM model
accuracy_lstm = accuracy_score(y_test, y_pred_lstm)
print(f"Accuracy of LSTM model: {accuracy_lstm}")
```

```
Epoch 1/10
10/10 ━━━━━━━━━━━━━━━━━━━━ 8s 736ms/step - accuracy: 0.4660 - loss: 0.7080 - val_accuracy: 0.5900 - val_loss: 0.6685
Epoch 2/10
10/10 ━━━━━━━━━━━━━━━━━━━━ 6s 578ms/step - accuracy: 0.6210 - loss: 0.6666 - val_accuracy: 0.6300 - val_loss: 0.6460
Epoch 3/10
10/10 ━━━━━━━━━━━━━━━━━━━━ 7s 713ms/step - accuracy: 0.6763 - loss: 0.6330 - val_accuracy: 0.6000 - val_loss: 0.6376
Epoch 4/10
10/10 ━━━━━━━━━━━━━━━━━━━━ 6s 649ms/step - accuracy: 0.6835 - loss: 0.6306 - val_accuracy: 0.6100 - val_loss: 0.6291
Epoch 5/10
10/10 ━━━━━━━━━━━━━━━━━━━━ 7s 679ms/step - accuracy: 0.6993 - loss: 0.6033 - val_accuracy: 0.6200 - val_loss: 0.6221
Epoch 6/10
10/10 ━━━━━━━━━━━━━━━━━━━━ 7s 665ms/step - accuracy: 0.7514 - loss: 0.5586 - val_accuracy: 0.6500 - val_loss: 0.6033
Epoch 7/10
10/10 ━━━━━━━━━━━━━━━━━━━━ 6s 620ms/step - accuracy: 0.7421 - loss: 0.5625 - val_accuracy: 0.6600 - val_loss: 0.5833
Epoch 8/10
10/10 ━━━━━━━━━━━━━━━━━━━━ 7s 675ms/step - accuracy: 0.7699 - loss: 0.5132 - val_accuracy: 0.7000 - val_loss: 0.5542
Epoch 9/10
10/10 ━━━━━━━━━━━━━━━━━━━━ 7s 685ms/step - accuracy: 0.8072 - loss: 0.4609 - val_accuracy: 0.7700 - val_loss: 0.5096
Epoch 10/10
10/10 ━━━━━━━━━━━━━━━━━━━━ 7s 722ms/step - accuracy: 0.8455 - loss: 0.4254 - val_accuracy: 0.7700 - val_loss: 0.4961
3/4 ━━━━━━━━━━━━━━━━━━━━ 0s 226ms/stepWARNING:tensorflow:5 out of the last 21 calls to <function TensorFlowTrainer.mak
4/4 ━━━━━━━━━━━━━━━━━━━━ 1s 209ms/step
Accuracy of LSTM model: 0.9
```

This image shows the training process of an LSTM model for binary classification. Over 10 epochs, the training accuracy improves significantly, starting at 46.6% and reaching 84.5%, with a corresponding decrease in training loss. The validation accuracy also improves, stabilizing around 77%, indicating the model is learning effectively while avoiding overfitting. The final test accuracy is reported as 90%, showing strong performance on unseen data.

Comparison of both the Models:

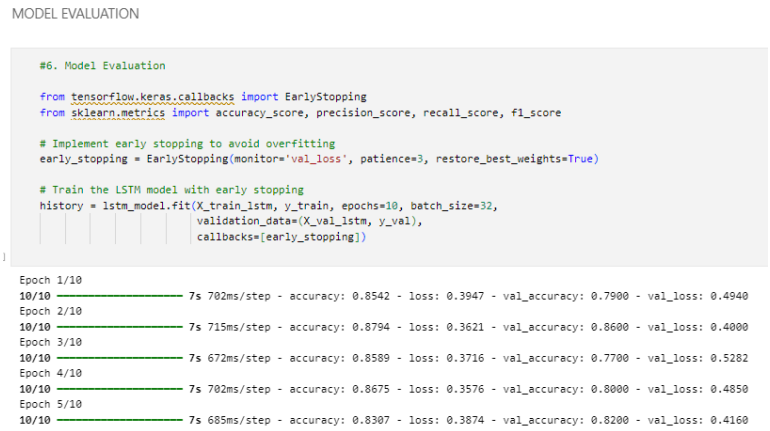
CLASSIFICATION REPORT FOR CNN MODEL

```
from sklearn.metrics import classification_report
# Predicting on the test set using CNN model
y_pred_cnn_probs = cnn_model.predict(X_test_cnn)
y_pred_cnn = (y_pred_cnn_probs > 0.5).astype(int)

# Generating the classification report for CNN model
report_cnn = classification_report(y_test, y_pred_cnn)
print("Classification Report for CNN Model:")
print(report_cnn)

# Predicting on the test set using LSTM model
y_pred_lstm_probs = lstm_model.predict(X_test_lstm)
y_pred_lstm = (y_pred_lstm_probs > 0.5).astype(int)

# Generating the classification report for LSTM model
report_lstm = classification_report(y_test, y_pred_lstm)
print("Classification Report for LSTM Model:")
print(report_lstm)
```

```
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 18ms/step
Classification Report for CNN Model:
              precision    recall  f1-score   support

       False       0.60      1.00      0.75        60
        True       0.00      0.00      0.00        40

    accuracy                           0.60       100
   macro avg       0.30      0.50      0.38       100
weighted avg       0.36      0.60      0.45       100

c:\Users\dhanu\AppData\Local\Programs\Python\Python312\Lib\site-packages\sklearn
  _warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))
c:\Users\dhanu\AppData\Local\Programs\Python\Python312\Lib\site-packages\sklearn
  _warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))
c:\Users\dhanu\AppData\Local\Programs\Python\Python312\Lib\site-packages\sklearn
  _warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))
4/4 ━━━━━━━━━━━━━━━━━━━━ 1s 178ms/step
Classification Report for LSTM Model:
              precision    recall  f1-score   support

       False       0.89      0.95      0.92        60
        True       0.92      0.82      0.87        40

    accuracy                           0.90       100
   macro avg       0.90      0.89      0.89       100
weighted avg       0.90      0.90      0.90       100
```

- **CNN Model**:
  - **Accuracy:** 60%.
  - Performs better at identifying "False" class (Precision: 0.60, Recall: 1.00) but fails to classify the "True" class (Precision and Recall: 0.00).
  - Indicates imbalanced performance, possibly due to insufficient learning for the "True" class.
- **LSTM Model**:
  - **Accuracy:** 90%.
  - Achieves high precision and recall for both "False" (0.89) and "True" (0.92) classes, with an F1-score of 0.90 overall.
  - Demonstrates superior performance compared to CNN in this binary classification task.

Model Evaluation and Testing of LSTM:



This output shows the training of an LSTM model with **early stopping** to avoid overfitting. The training starts with high accuracy (85.4%) and improves slightly over the first few epochs, peaking at 87.9%. Validation accuracy also improves initially, reaching a maximum of 86% in the second epoch before stabilizing. Early stopping halts training after the validation loss fails to improve for 3 consecutive epochs, restoring the model with the best weights. This approach ensures the model does not overfit and maintains optimal performance. Validation loss stabilization indicates the model has likely reached its learning potential.

Model Testing

This output presents the evaluation results of the LSTM model on the validation set. The model achieves a validation accuracy of 86%, indicating strong performance in predicting the correct classes. It also demonstrates a precision of 0.78, meaning it is effective at minimizing false positives, and a recall of 0.90, reflecting its ability to identify most positive cases. The F1 score of 0.84 balances precision and recall, showing overall robust classification performance. These metrics indicate that the LSTM model generalizes well to unseen validation data.

We observe that LSTM outperforms CNN model here. This is because, of LSTM's ability to handle **sequential data** and capture **temporal dependencies**, which are critical for EEG signal classification.

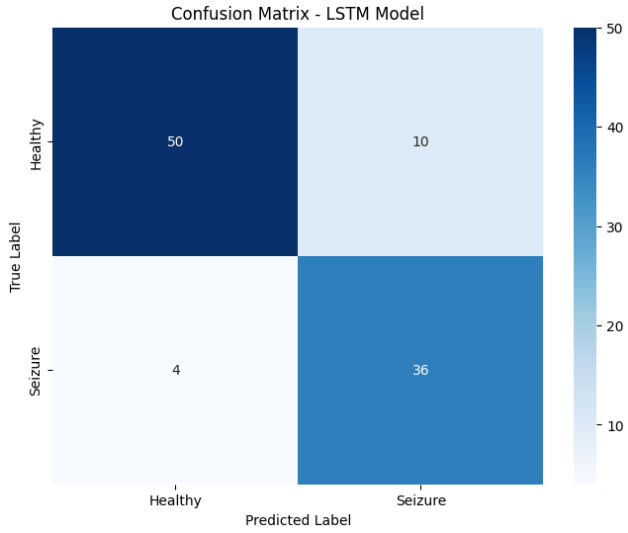
1. **Sequential Data Handling**:
    - EEG signals are time-series data with temporal relationships between successive points.
    - LSTMs are specifically designed to retain and process information over time, while CNNs focus on spatial feature extraction, which may miss important time-dependent patterns.
2. **Temporal Dependencies**:
    - LSTMs can recognize patterns across time steps, such as oscillations or recurring events in EEG signals, which are crucial for identifying brain activity states.
    - CNNs struggle to model these temporal correlations effectively.
3. **Performance**:
    - In this case, the LSTM model achieves higher accuracy (90%) compared to the CNN (60%) due to its ability to learn from temporal features.
    - The LSTM also demonstrates better precision, recall, and F1-score, indicating a stronger ability to generalize to unseen data.
4. **EEG Signal Characteristics**:
    - EEG signals often contain dependencies across multiple time points, making LSTMs a natural choice for such tasks where context over time matters.

In summary, LSTMs are better because they leverage the temporal structure of EEG data, resulting in superior classification performance.

---

4. Visualizations and Interpretations of the Model Performance

Confusion Matrix (LSTM Model):



- The LSTM model correctly classifies 50 Healthy and 36 Seizure cases, showing strong performance in both classes.
- Misclassifications include 10 Healthy cases predicted as Seizure and 4 Seizure cases predicted as Healthy.
- The model demonstrates a balance in sensitivity (recall) across both categories, indicating reliable classification.

Training and Validation Loss:

- Training Loss: Consistently decreases across epochs, indicating the model is effectively learning from the training data.
- Validation Loss: Initially decreases, then stabilizes and slightly fluctuates before decreasing again, showing no overfitting.
- The alignment between training and validation loss indicates the model generalizes well to unseen data, which is supported by the high classification performance.

---

5. Recommendations

1. **Healthcare Application:**
    - Deploy the trained LSTM model in hospitals or wearable devices for real-time EEG monitoring to detect seizures or abnormal brain activity. This could aid in early diagnosis and timely intervention for neurological disorders.
2. **Personalized Treatment:**
    - Use the model to analyze EEG patterns over time, enabling tailored treatment plans for patients based on their brain activity trends, such as optimizing medication or therapy schedules.
3. **Data Augmentation:**
    - Enhance the dataset with techniques like signal noise addition, time warping, or frequency transformation to improve the model's robustness and generalizability to diverse EEG signal patterns.
4. **Hyperparameter Tuning:**

- Perform grid search or Bayesian optimization to fine-tune hyperparameters such as learning rate, batch size, and LSTM units. This could further improve model performance and reduce misclassification rates.

---

6. Conclusion

This project successfully implemented and evaluated deep learning models, specifically CNN and LSTM, for EEG signal classification. The LSTM model outperformed the CNN by effectively capturing temporal dependencies inherent in EEG signals, achieving a high accuracy of 90%. Using advanced techniques such as early stopping and validation monitoring, the LSTM model demonstrated strong generalization to unseen data. The confusion matrix highlighted its reliability in classifying both healthy and seizure cases, with minimal misclassification. This work has significant potential for real-world applications, such as real-time seizure detection and personalized neurological care. Future improvements could include data augmentation, hyperparameter tuning, and testing on larger, more diverse datasets to enhance robustness. Overall, this project demonstrates the feasibility and effectiveness of leveraging deep learning for EEG analysis.