Technique 1

To better prepare the ECG data for the seizure detection model, a multi-step preprocessing pipeline was implemented starting with taking the raw ECG signals from the pre-ictal phase and non-seizure events and loading them from the dataset and converting them into numerical arrays. They were given annotations marking seizure events and were then extracted to generate corresponding binary labels, where 1 indicated seizure activity and 0 represented normal heart activity.

Due to the significant class imbalance of seizure datasets, random upsampling was applied to balance seizure and non-seizure samples, ensuring the model received a more equitable distribution of both classes since there are significantly more seizure events than non-seizure. Next, the ECG signals were standardized using a Z-score transformation to normalize the variations in amplitude and patient-specific differences. Finally, to capture temporal dependencies, the data was segmented into sequences with a length of 100 time steps. This was the input of the LSTM model. Overall, preprocessing ensured that the ECG data was structured appropriately for training an effective deep learning model.

Technique 2

An LSTM network was implemented to accurately detect seizure events from ECG signals. The LSTM was chosen due to its ability to capture long-range dependencies in sequential data. This is ideal for analyzing data like ECG data. The model consisted of two stacked LSTM layers, a dense layer with ReLU activation, and a dropout layer which prevents overfitting. The final output layer used a sigmoid activation function to produce a probability score for seizure occurrence, 1 being seizure-like activity and 0 being non-seizure.

To optimize its training, the model was compiled using the Adam optimizer and Binary Focal Cross entropy loss. This helps to mitigate the major class imbalance by emphasizing harder-to-classify samples. Additionally, performance metrics such as Precision, Recall, AUC, and F1-score were measured to make sure the model was effectively distinguishing between both seizure and non-seizure events.

Early stopping and learning rate reduction techniques were also incorporated into the development of the LSTM to improve training efficiency and prevent overfitting.

Technique 3

To enhance the model's robustness and help mitigate the class imbalance, more data preprocessing and augmentation techniques were applied to the ECG signals before beginning to train the LSTM model. First, the raw ECG signals were standardized using StandardScaler to help ensure that all of the features had a mean of zero and a standard deviation of one to stabilize model training.

Because seizure events are significantly underrepresented in the dataset, resampling techniques were necessary to employ for balancing the two classes. Specifically, seizure instances were upsampled using random resampling. This ensured a more equal distribution of seizure and non-seizure events.

Further improvements to generalization were gained through data augmentation techniques. These included random noise injection and slight temporal shifts to help the model become more robust to variations in real-world ECG recordings. This ensured that the input data was properly structured and balanced.

Technique 4

To assess the trained LSTM model’s effectiveness in detecting seizures from ECG data, several evaluation metrics were implemented. The model's performance was tested using a held-out test dataset that was not seen during training or validation.

After training, the model was used to generate predictions on the test set. A classification report was generated to provide key performance metrics, including precision, recall, F1-score, and AUC (Area Under the Curve), which are critical for evaluating the reliability of a binary classification model.

Since the dataset exhibited an imbalance between seizure and non-seizure samples, traditional accuracy was not a sufficient measure of model performance. Instead, F1-score, which balances precision and recall, was incorporated into the model evaluation. This ensured that both false positives and false negatives were taken into account, making the metric more suitable for a medical application where misclassifications could have significant consequences.

The trained model's predictions on the test set were binarized using a threshold of 0.5. The classification report provided insight into how well the model generalized to unseen ECG data. These evaluations helped refine the model by identifying areas for potential improvement, such as adjusting class weights, fine-tuning hyperparameters, or exploring additional preprocessing techniques.