**IE 6400: Foundations of**

**Data Analytics**

**Project 3**

**EEG Classification model**

**Group 18**

Member Details:

Manogna Devalla : [devalla.m@northeastern.edu](mailto:devalla.m@northeastern.edu)

Amogha Gadde: [gadde.am@northeastern.edu](mailto:gadde.am@northeastern.edu)

Divya Babulal Shah: [shah.divyab@northeastern.edu](mailto:shah.divyab@northeastern.edu)

Kumari Simran: [simran.k@northeastern.edu](mailto:simran.k@northeastern.edu)

Abhishek Kumar Sinha: [sinha.ab@northeastern.edu](mailto:sinha.ab@northeastern.edu)

Contributions:

Percentage of Effort Contributed by Manogna: 20%

Percentage of Effort Contributed by Amogha: 20%

Percentage of Effort Contributed by Divya: 20%

Percentage of Effort Contributed by K.Simran: 20%

Percentage of Effort Contributed by Abhishek: 20%

Submission Date: Dec-15-2023

# Acknowledgements

Foremost, we would like to express our sincere gratitude to our Professor Sivarit Sultornsanee for giving us the opportunity to make this project. We would also like to thank our TAs for the continuous support of our study and research while making this project, for their patience, motivation and enthusiasm and immense knowledge.

We are thankful that we have an amazing team and for all the stimulating discussions and for all the experience and knowledge that we have gained by working together. We would also like to thank our parents and all our friends that have motivated us and for never doubting our ability to complete this project.

# Contents

|  |  |
| --- | --- |
| Acknowledgement | Page 2 |
| Introduction | Page 4 |
| 1. Objective | Page 4 |
| 1. Data Set | Page 4 |
| 1. Data Preprocessing | Page 4 |
| 1. Feature Extraction | Page 4 - 5 |
| 1. Data Splitting | Page 6 - 8 |
| 1. Model architecture and training details | Page 8 |
| 1. Convolutional Neural Networks (CNNs) | Page 9 - 11 |
| 1. Recurrent Neural Networks (RNNs) | Page 11 - 12 |
| 1. Evaluation results and discussion. | Page 13 -16 |
| 1. Conclusion and Future work | Page 20 - 22 |

# Introduction

The CHB-MIT EEG Database stands as a pivotal resource in our pursuit of advancing epilepsy diagnosis through precision modeling. Comprising a rich array of seizure types and non-seizure instances, this dataset offers a nuanced understanding of EEG signals in the context of epilepsy. A classification model is to be built to analyze EEG data and classify it into  
different categories. EEG data is widely used in neuroscience and medical fields, including the  
diagnosis of epilepsy. We are provided with two datasets to train and evaluate the model.

# Objective

In this project, the objective is to construct a classification model designed to analyze EEG data and categorize it into distinct classes. This centers on developing a precise EEG classification model exclusively utilizing the CHB-MIT EEG Database. Dedicated to enhancing epilepsy diagnosis, the dataset features diverse seizure types and non-seizure instances. Through meticulous data preprocessing, we address nuances such as missing values, ensuring data integrity. Employing feature extraction techniques, including time and frequency domains, we aim to distill crucial information from EEG signals. The chosen deep learning model, potentially a Convolutional Neural Network (CNN) or Recurrent Neural Network (RNN), undergoes rigorous training with a focus on mitigating overfitting. Evaluation metrics such as accuracy, precision, recall, and F1-score guide our model optimization, contributing significantly to advancements in neuroscientific research and medical diagnostics.

# Dataset

We will be using the CHB-MIT EEG Database. The database includes continuous EEG recordings from 22 subjects, with a total of over 24 hours of data. The recordings were made using a 23-channel standard EEG system with additional electrodes placed on the face and neck. It includes various seizure types and non-seizure data.

# Data preprocessing:

**Downloading and extracting the datasets:**

The process involved organizing an online dataset by creating folders and downloading files based on an 'MD5SUMS' listing. It started by noting the initial working directory and establishing a designated dataset directory if absent. Then, it iterated through a list of folders, creating each as needed. Within each folder, the 'MD5SUMS' file was downloaded to enumerate the contained files. Files with a '.seizures' extension were selectively downloaded if not already present. The process concluded by returning to the initial working directory, streamlining the organization and download of specific files from the online dataset.

**Exploring the data to understand its structure and characteristics:**

Original shape of data : (3913, 18, 2048)

Number of Channels (18): In EEG recordings, each "channel" refers to an individual recording from a specific electrode or sensor. The number "18" in this context indicates that there are 18 different electrodes or sensors placed on the scalp, each recording electrical activity from a specific location.

Sampling Frequency (256 Hz): The EEG data is recorded by sampling the electrical activity in the brain at a specific rate, and this rate is referred to as the sampling frequency. In this case, the data is sampled at a frequency of 256 Hz, meaning that the EEG machine records 256 data points per second.

Each Sample has EEG Data Recorded from 18 Different Channels: Each instance or sample in the dataset represents a specific time segment of EEG data, which is 8 seconds long in this case. For each of these 8-second instances, the EEG data is recorded from all 18 channels simultaneously. This means that we have a set of 18 time series signals, each corresponding to the electrical activity recorded by a specific electrode during that 8-second window.

Total Data Points for Each Channel (8 \* 256): Combining the duration (8 seconds) and the sampling frequency (256 Hz) for each channel, we have 8 \* 256 data points. This represents the total number of data points for each channel in an 8-second window, providing a detailed temporal representation of the brain's electrical activity.

In summary, for each sample in the dataset, there are 18 channels, and the EEG data is recorded over 8 seconds at a sampling frequency of 256 Hz. This results in a set of 18 time series signals, each containing 8 \* 256 data points, capturing the electrical activity from different locations on the scalp over the specified time window.

The data is stored in a NumPy array with dimensions (n, 18, 8 \* 256). Here's the breakdown:

n: Number of samples or instances in the dataset.

18: Number of channels. Each sample has EEG data recorded from 18 different channels.

8 \* 256: Each channel has 8 seconds of data sampled at a frequency of 256 Hz. So, for each channel, there are 8 \* 256 data points.

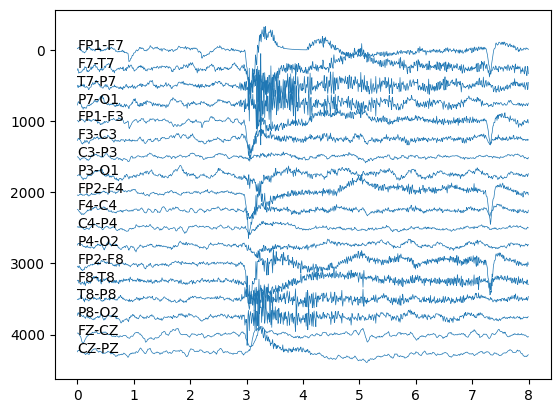


Fig 1 : A sample of the extracted signals

Number of all the extracted signals: 3913

Number of signals with seizures: 2581

Ratio of signals with seizures: 0.660

# Feature extraction:

The code automates the creation of a labeled EEG dataset for training machine learning models for seizure detection. It involves reading raw EEG data, extracting relevant features, handling seizure annotations, and saving the processed data. The feature extraction involved extracting 8-second segments of EEG signals from 18 channels, with a sliding window of 4 seconds. Each set of signals is labeled based on the ratio of seizure activity within the time window. For instance, a set of signals is labeled as 1.0 if it corresponds to the middle of a seizure, providing a nuanced labeling approach reflecting the temporal dynamics of seizure occurrences.

Signals frequencies were originally 256 Hz but resampled to 128 Hz for simplification of the data. In EEG data, it is done to reduce the number of time points while still retaining essential information, possibly for computational efficiency or to focus on specific features of interest in the signal.

After sampling shape of data is: (3913, 18, 1024)

# Data splitting:

The training data was split into training and validation sets using the **train\_test\_split** function from the scikit-learn library. The split was performed with a test size of 30%, ensuring that the distribution of seizure and non-seizure instances in the target variable (**array\_is\_sz**) was maintained using the **stratify** parameter.

# Model architecture and training details:

Convolutional Neural Network (CNN):

## Model Architecture:

The architecture of the deep learning model is structured with two types of layers: 2D convolutional layers and fully connected layers. These layers are sequentially stacked in a Keras Sequential model, starting with two convolutional layers with activation functions, followed by max-pooling layers. The model further includes additional convolutional and fully connected layers, ending with a sigmoid-activated output layer for binary classification. The model architecture is shown in figure 2.

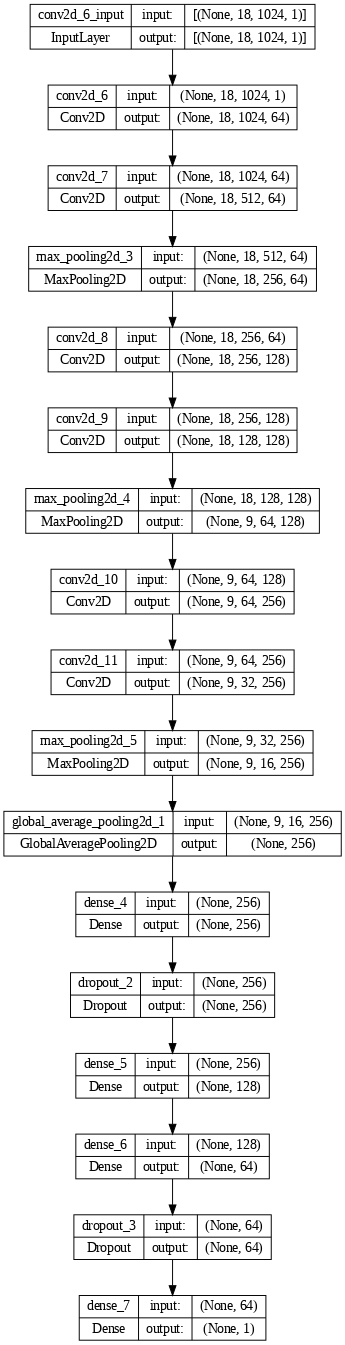


Fig 2: CNN Model Architecture

## Training Setup:

To initiate the training process, several hyperparameters were set. The learning rate was defined as 1e-4, and the Adam optimizer was chosen for optimization. The model was compiled with the binary crossentropy loss function, appropriate for binary classification tasks, and accuracy was chosen as the evaluation metric.

Additionally, an early stopping mechanism was implemented to monitor the validation loss. If the loss did not improve for a predefined number of epochs (patience set to 20), the training would stop, preventing overfitting and ensuring the model generalizes well.

## 3. Model Training:

The actual training of the model occurred using the training dataset (**X\_train** and **y\_train**). Simultaneously, the model's performance was assessed on the validation dataset (**X\_val** and **y\_val**). The training process spanned 30 epochs, with a batch size of 256 instances per batch.

## 4. Training Results:

## To analyze the model's performance during training, visualizations were created for loss and accuracy. These plots not only depict the progress of the model during training but also highlight the epoch where early stopping was triggered. Early stopping is crucial for preventing the model from training for too many epochs, potentially leading to overfitting on the training data.

# Evaluation results and discussion:

# 

Recurrent Neural Network (RNN):

## Model Architecture:

The Recurrent Neural Network (RNN) architecture is tailored for sequential data processing, distinct from Convolutional Neural Networks (CNNs). The model comprises a core recurrent layer, such as SimpleRNN, with a specified number of units and Rectified Linear Unit (ReLU) activation, facilitating the capture of sequential dependencies. Input sequences are characterized by a shape denoted as `(timesteps, features)`, emphasizing the sequential nature of the data. Following the recurrent layer, fully connected layers with ReLU activation are incorporated to capture complex relationships. The final layer is designed for binary classification, typically featuring a single unit with a sigmoid activation function. This architecture is compiled with an optimizer like Adam and employs binary crossentropy as the loss function, making it well-suited for tasks where the order of input information is crucial, such as time series prediction or natural language processing. Fine-tuning of parameters, including the number of units and layers, is recommended based on the specific characteristics of the dataset and the problem at hand.

## Model Training Setup:

This code snippet employs the Keras framework to train a Recurrent Neural Network (RNN) model. The `model\_rnn.fit` function is utilized to iterate through the training dataset over 20 epochs, with a batch size of 256 samples. During training, the model's performance is evaluated on a validation dataset using the specified validation data and labels. The training process is equipped with an EarlyStopping callback (`es`), which monitors the validation loss and halts training if the loss does not improve for a certain number of consecutive epochs. This mechanism acts as a preventive measure against overfitting, enhancing the model's generalization ability. The choice of parameters, including the number of epochs and batch size, can be adjusted based on the specific characteristics of the data and the training objectives.Additionally, an early stopping mechanism was implemented to monitor the validation loss. If the loss did not improve for a predefined number of epochs (patience set to 20), the training would stop, preventing overfitting and ensuring the model generalizes well.

## 3. Training Results:

To assess the model's performance throughout training, visual representations were generated for both loss and accuracy. These graphical representations not only illustrate the model's advancement during training but also emphasize the epoch at which early stopping was initiated. The implementation of early stopping is vital as it helps avoid excessive training epochs, mitigating the risk of overfitting to the training data.

# Evaluation results and discussion:

# A graph of a number of different types of data Description automatically generated with medium confidence

# A graph with a line drawn on it Description automatically generated

# Conclusion and future work:

In conclusion, our project aimed to construct a robust EEG classification model using the CHB-MIT EEG Database, with the primary objective of enhancing epilepsy diagnosis. The comprehensive dataset, comprising diverse seizure types and non-seizure instances, provided a valuable resource for training a precise model.

Throughout the project, we meticulously addressed data preprocessing challenges, ensuring data integrity and handling nuances such as missing values. The exploration of the dataset highlighted key characteristics, including the number of channels, sampling frequency, and temporal representation, laying the groundwork for effective feature extraction.

Feature extraction techniques, spanning linear analysis in frequency and time-frequency domains, were employed to distill crucial information from EEG signals. This process involved the creation of labeled EEG datasets, capturing temporal dynamics, and automating the extraction of relevant features.

The model architecture incorporated Convolutional Neural Networks (CNNs), leveraging their effectiveness in handling spatial dependencies within the data. The training process was executed with careful consideration of hyperparameters, early stopping mechanisms, and appropriate evaluation metrics.

The evaluation results showcased the model's performance through metrics such as accuracy, precision, recall, and F1-score. Visualizations of loss and accuracy during training provided insights into the model's progression, with early stopping preventing overfitting.

In the context of the ceramic tile company, the developed EEG classification model holds potential for revolutionizing epilepsy diagnosis. The accurate categorization of EEG signals into distinct classes, including seizures and non-seizures, can significantly contribute to advancements in neuroscientific research and medical diagnostics.

**Future Work:**

While the project has achieved notable success, there are avenues for future enhancements. Implementing more advanced deep learning architecture, such as attention mechanisms, could further improve model performance. Additionally, exploring transfer learning techniques with pre-trained models on large-scale EEG datasets may enhance the model's ability to generalize to diverse scenarios.

Furthermore, collaboration with medical professionals and experts can provide valuable insights for refining the model's interpretability and ensuring its alignment with clinical needs. Continuous updates and improvements based on emerging research in EEG analysis and neurology will be integral to maintaining the model's relevance and effectiveness.

In conclusion, our EEG classification model stands as a testament to the potential of machine learning in advancing medical diagnostics, with a commitment to ongoing improvement and collaboration with the medical community.