Project Report on

Automatic Epileptic Seizure Localisation from EEG Signals

Submitted in partial fulfillment of the requirements for the award of the degree of

Bachelor of Technology

in

Computer Science and Engineering

By

Niyatha V S (U2103163)

Paul Allen Kadayaparambil (U2103164)

Powell Moothedan (U2103165)

Rose Jacob (U2103185)

Under the guidance of

Dr. Sminu Izudheen

Department of CSE.
Rajagiri School of Engineering & Technology (Autonomous)
(Parent University: APJ Abdul Kalam Technological University)
Rajagiri Valley, Kakkanad, Kochi, 682039
April 2025

CERTIFICATE

This is to certify that the project report entitled ***"Automatic Epileptic Seizure Localisation from EEG Signals"*** *is a bonafide record of the work done by* ***Niyatha V S (U2103163)****,* ***Paul Allen Kadayaparambil (U2103164)****,* ***Powell Moothedan (U2103165)****,* ***Rose Jacob (U2103185)****, submitted to the Rajagiri School of Engineering & Technology (RSET) (Autonomous) in partial fulfillment of the requirements for the award of the degree of Bachelor of Technology (B. Tech.) in "Computer Science and Engineering" during the academic year 2024-2025.*

Dr. Sminu Izudheen
Project Guide
Professor
Dept. of CSE
RSET

Ms. Sangeetha Jamal
Project Co-ordinator
Assistant Professor
Dept. of CSE
RSET

Dr. Preetha K G
Professor & HOD
Dept. of CSE
RSET

ACKNOWLEDGMENT

We wish to express our sincere gratitude towards **Rev. Dr. Jaison Paul Mulerikkal CMI**, Principal of RSET, and **Dr. Preetha K G**, Head of the Department of Computer Science and Engineering for providing us with the opportunity to undertake our project, "Automatic Epileptic Seizure Localisation From EEG Signals".

We are highly indebted to our project coordinator, **Ms. Sangeetha Jamal**, Assistant Professor, Department of Computer Science and Engineering, for her valuable support.

It is indeed our pleasure and a moment of satisfaction for us to express our sincere gratitude to our project guide, **Dr. Sminu Izudheen**, Professor, Department of Computer Science and Engineering, for her patience and all the priceless advice and wisdom she has shared with us.

Last but not the least, we would like to express our sincere gratitude towards all other teachers and friends for their continuous support and constructive ideas.

Niyatha V S

Paul Allen Kadayaparambil

Powell Moothedan

Rose Jacob

Abstract

Epilepsy is a neurological condition affecting both children and adults. Rapid, recurring epileptic seizures are its defining feature. In essence, a set of brain cells' electrical activity is abruptly disturbed. Epileptic seizures are brought on by an overabundance of electrical activity in the brain. These seizures can be detected using EEG. Epileptic seizure localisation is a vital aspect of neurological care, where accurate and timely identification of seizure events can greatly enhance patient management and outcomes. Using the publicly accessible CHB-MIT dataset, this project investigates the use of deep learning models to localize seizures associated with epilepsy using EEG data. The raw EEG signals are preprocessed, and meaningful features are extracted to train models capable of classifying seizure and non-seizure events. To ensure that these models are interpretable, we integrate advanced post-hoc explanation techniques, including LIME (Local Interpretable Model-Agnostic Explanations) and SHAP (SHapley Additive exPlanations). These techniques provide us a better comprehension of the decision-making process by enabling us to see and measure how each particular EEG feature contributes to the model's predictions. For clinical adoption, this degree of explainability is essential since it enables medical professionals to assess and rely on the model's results. The performance of the deep learning model and explainable AI models is evaluated on benchmark EEG datasets, with a focus on enhancing their explainability to facilitate their adoption in clinical settings. The results demonstrate the possibility of merging deep learning with SHAP and LIME to create an interpretable and effective seizure localisation system, offering a valuable tool for clinicians in the management of epilepsy.

Contents

List of Abbreviations

ACC - Accuracy

AI - Artificial Intelligence

Bi-LSTM - Bidirectional Long Short-Term Memory

CHB-MIT - Children's Hospital Boston-Massachusetts Institute of Technology

CNN - Convolutional Neural Network

DL - Deep Learning

DWT - Discrete Wavelet Transform

EEG - Electroencephalogram

FN - False Negative

FP - False Positive

GPU - Graphics Processing Unit

GMM - Guassian Mixture Model

MAD - Median Absolute Deviation

IQR - Interquantile Range

LIME - Local Interpretable Model-Agnostic Explanations

Grad-CAM -Gradient-weighted Class Activation Mapping

ML - Machine Learning

PRC - Precision

PSD - Power Spectral Density

SHAP - SHapley Additive exPlanations

TN - True Negative

TP - True Positive

VQA - Visual Question Answering

XAI - Explainable Artificial Intelligence

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background

Epilepsy is a neurological disorder that has a significant bearable burden across the globe including children and adults alike. It is a chronic disorder emanating from the brain defined by a number of recurrent surprise, unpredictable seizures due to an uncontrolled electrical activity in the brain. These are followed by subsequent disturbances in the normal cyclical activities of the brain lasting a few minutes and causing some body parts to shake irritably, loss of bodily sensation and intermittent shaking of the entire body. Hence the timely and accurate diagnosis proves to be important for the improvement of the overall patient management [1],[2].

Electroencephalography (EEG) helps to monitor and record brain waves. This EEG test usually comes 20 to 40 hours of EEG monitoring concerning different sheets of paper. One sheet represents 1 second of the EEG data. The medical practitioners usually interpret this data manually and provide annotations which is very tedious and error infested. This illustrates the problem that one has to solve with the help of localization methods that are fast and precise. Advanced deep learning systems and technologies can be used to solve this problem, which is largely because attention-based models like CNNs have been used effectively in seizure detection and localization particularly achieving optimal results. Although CNNs perform well in extracting or learning spatial characteristics of EEG data, they are perceived to be black box models because of the difficulty in comprehending how they arrive at their conclusions [1],[3].

One of the main reasons why clinical utilization is a big challenge is because there is no clarity on what led to a prediction. For the neurologists to trust the model's judgments, they must comprehend how they are made. Explainable AI (XAI) is useful in this situation. SHAP and LIME are the commonly used techniques to explain the decisions of the model in a way that can be perceived. These techniques aid in the depiction and estimation of the amount of each feature that contributed to certain predictions by the model [2], [3]. This kind of explainability increases confidence on the model and also assures that the model is fit for use in the care of patients.

Hence, there is great need for such explainable seizure localisation systems to be developed not only to achieve the performance of the models but also to facilitate their clinical use. These systems present a rationale for the predictions made, thereby allowing clinicians to make better decisions which in turn leads to improved management of epilepsy and its associated problems [1], [3].

1.2 Problem Definition

The project aims to create an explainable deep learning based model to detect the ictal phase of epileptic seizures by the use of EEG data. This approach aims at helping the neurologists analyze and localize the seizures faster and more accurately by analyzing the data rooted in XAI techniques whose focus is to make the decisions understandable. This seeks to interlink the technology and its practical clinical application.

1.3 Scope and Motivation

The project aims to create and test deep learning methods which will help to focus on the automatic localization of seizures in patients using EEGs. The project does preprocessing of the raw EEG signals, extracting relevant features from the EEG signals, and using models like CNNs to classify the datapoints as seizure and non-seizure events. Implementing methods of explainable artificial intelligence such as SHAP and LIME is intended to fulfil a pre-condition of this approach – ability to trust the algorithm in a clinical context. Therefore, this project also focuses on developing a system that is both accurate and bringing interpretability to the output of the model so that medical practitioners are able to use it in their practice.

The background for this project is caused by the heavy impact that epilepsy has, not only on patients and their families, but also on the healthcare systems. Existing approaches to monitoring and diagnosing seizures tend to rely on too much human effort, which is also the source of the mistakes, indicating the relevant demand for such systems to be target ready. Deep learning's potential to improve the standard of treatment for epileptic patients through quick, precise, and early detection is what motivates this study. In addition, the use of explainable AI also overcomes one of the major barriers to implementation since it helps clinicians see the point of such a model and makes them believe in it. In order to provide the best possible therapeutic use of this technology, this project aims to improve the current practice and solve its limits.

1.4 Objectives

1. To collect, preprocess and tranform raw EEG signals into the appropriate format for deep learning models.
2. To develop a deep learning model based on seizure detection using CNN as part of the project
3. To improve the explainability and clinical trustworthiness of the model, efforts will be made to integrate the following explainable AI techniques: LIME and SHAP.
4. Developed models will be assessed based on their accuracy, precision, confusion matrices, and other measures to ensure reliability and effectiveness of the findings.
5. To automate threshold fixing to guarantee reliable seizure event boundary detection where the prediction time threshold is adjusted using an auto-calibration technique.
6. To develop a simple and straightforward framework that interfaces with neurologists ensuring that the explanations given by the model will lead to improved patient outcomes.

1.5 Challenges

The project experiences a number of issues, more specifically, the issues that can be associated with, data variability, overfitting, explainability, and computational limitations in the order of their familiarity. Data variability poses a challenge in that it is common that the EEG waveforms of different patients may differ and therefore constructing a model that is capable of performing well across different datasets becomes an uphill task [2]. Another problem associated with models is overfitting, primarily because model can be provided with a limited quantity of training data and as such there is a dip in the quality of the model to predict new data [3]. Explainability also remains a key hurdle, ensuring that the model's predictions, as well as the accompanying explanations, are produced in an acceptable form and clarity is very important for the model to be applied in practice [1]. Lastly, there are also resource constraints that inhibit the training and testing of models, particularly in terms of GPU computational resources [4].

1.6 Assumptions

1. Quality of the EEG Data: The EEG data obtained from the Bonn University database is presumed to be clean. Noise, as well as artifacts, present in the EEG signals, are reduced via preprocessing step hence the data is fit for the purpose of model training and evaluation.

2. Patient Variability: It is assumed that the seizure patterns would be different for different persons, however it's expected that the decision of the CNN model will be correct regarding these differences although the EEG patterns of individuals are considered to be different.

3. Data Availability: The undertaking assumes the existence of sufficient quantity of EEG data for model training and testing. This ensures that the model can learn to detect seizures accurately across various scenarios.

4. Computational Resources:It is assumed that sufficient high performance technical equipment or processing units, for example, graphic processing units, shall be present to help train the model in an effective way. If there are no graphical processing units in the setting, model training efficiency shall drop drastically.

5. Model Interpretability: The system proposes that employments of XAI techniques will offer appropriate and useful output, which healthcare practitioners can have confidence to use it in practice.

1.7 Societal / Industrial Relevance

The undertaking is of immense relevance both to society as well as to industry with a great focus on healthcare.As deep learning models are implemented, one can detect the occurrence and the position of epileptic seizures with a greater degree of accuracy than before. This could change the way seizures are controlled, introducing a more dependable and faster way of the seizure detection system. It is expected that this will help the patients by giving neurologists better information on how often seizure episodes occur allowing the better treatment and management of the patients.

This study has practical implications for those working in or running businesses involved in the manufacturing of medical equipment and devices related to the treatment of diseases affecting the brain. Attempts to explain black-box seizure detection models using explainable AI techniques such as SHAP and LIME would help build trust in the models, enabling them to be used in practice. Moreover, advancements in deep learning for applications such as real-time monitoring systems can stimulate the development of innovative epilepsy healthcare wearable and remote monitoring devices that can on the patients' behalf manage and control seizures for additional comfort and safety of the patients suffering from epilepsy.

Additionally, the interpretability oriented aspect of the project will be useful for any industry that will seek to implement AI models in complex environments like healthcare, where there is a need to comprehend the decisions made by the model for safety and legal purposes. This research may lead to more user-friendly, efficient, and non-biased AI systems in the field of health care, moving away from manual evaluation techniques to supporting operational evidence to aid the clinical staff in decision making.

1.8 Organization of the Report

The seven chapters that make up this report each contribute to a thorough understanding of the planning, creation, and assessment of an explainable AI-based seizure detection system that uses EEG data. The project theme is introduced in Chapter 1, which also outlines the report structure and provides specifics on the problem definition, motivation, objectives, scope, challenges, assumptions, and societal relevance. A comprehensive review of the literature on current seizure detection techniques is provided in Chapter 2, with an emphasis on machine learning, deep learning (especially CNNs), and explainable AI approaches such as SHAP, LIME, and Grad-CAM. Important research gaps are also identified. Chapter 3 describes the hardware and software requirements, and the budget. With a focus on clinical workflow integration, Chapter 4 describes the system's architectural design. EEG preprocessing, CNN-based classification, statistical thresholding with MAD and GMM, and the addition of SHAP and LIME for model interpretability are all covered in detail in Chapter 5. The experimental findings, performance indicators, and interpretability outputs are covered in Chapter 6, along with a comparison to ground-truth seizure intervals. Chapter 7 ends with a summary of the results, highlighting the system's clinical applicability and outlining potential future improvements like real-time deployment, more extensive dataset generalisation, and sophisticated explainability mechanisms.

Chapter 2

Literature Survey

2.1 Existing Systems

Because deep learning can learn intricate spatial and temporal patterns from EEG data, its application in seizure detection has drawn a lot of attention in the last ten years. The CNN, one of the most popular architectures in this field, has shown excellent accuracy in distinguishing between seizure and non-seizure events. Conventional models, which have mostly concentrated on patient-specific data, frequently perform well but are not very generalisable. Systems that use the CHB-MIT dataset, which is a well-annotated EEG dataset of paediatric patients with epilepsy, for example, frequently train on data from particular patients and have occasionally achieved accuracy of over 96%.In order to extract meaningful features across various frequency bands, such as delta, theta, alpha, beta, and gamma, these models typically use preprocessing techniques like segmenting EEG signals into overlapping windows and applying Discrete Wavelet Transform (DWT).

Typically, the model architectures incorporate Bi-directional Long Short-Term Memory (Bi-LSTM) layers for learning temporal dependencies and 1D-CNNs for feature extraction. Sometimes, in order to improve classification performance, attention mechanisms are also incorporated to highlight important time segments. Even though these models work well, they frequently act as "black boxes," offering little insight into the decision-making process. Since medical professionals need explainability to validate automated decisions, this lack of transparency is a significant obstacle to clinical adoption.

Newer systems have begun using Explainable AI (XAI) techniques to address this. Method using XAI techniques is a noteworthy framework that presents a seizure detection method that is independent of the patient. This technique visualises the contribution of various EEG channels and time segments towards predictions using both 1D and 3D CNNs in conjunction with SHAP (SHapley Additive exPlanations).This system extracts spectral and time-frequency domain features from EEG signals using Morlet wavelet transforms and Welch's method. The SHAP values are then used to create heatmaps that highlight the importance of features, giving clinicians a clear window into the model's decision-making process. Despite all of the advancements made many existing models still struggle to generalise different patients and EEG configurations, especially when they are used in real-time situations. These gaps underline the need for a clinically appropriate seizure detection system that is dependable, interpretable, and patient-independent.

2.2 Proposed System

The proposed system aims to address the limitations of existing seizure detection frameworks by introducing a deep learning-based model that combines Explainable AI techniques with CNN. It is efficient, interpretable, and also generalisable. Unlike all the traditional systems that focus on patient-specific data, this system prioritises a patient-independent approach, which allows it to function consistently across a variety of EEG profiles. The process begins with a raw EEG data in the .edf format, which is then converted into.csv files. The data is separated into discrete windows that correspond to pre-ictal, ictal, and interictal phases. Adding annotations with binary labels—0 for non-seizures and 1 for seizures—ensures consistency in supervised learning.

Using a 2D CNN architecture, the core classification model converts the EEG signals into matrix representations, with rows representing time samples and columns representing EEG channels. Unlike the 1D-CNNs, this method allows the model to learn spatial patterns across electrode locations. Regularisation techniques and dropout layers are used to improve generalisation and avoid overfitting. The system is appropriate for binary seizure classification tasks since the final classification is based on a sigmoid activation function.

One of the important innovation in this system is the combination of two powerful XAI tools, SHAP and LIME. SHAP values, which indicate the relative contribution of each EEG channel and time period to the model's prediction, are computed to generate heatmaps consistent with clinical EEG interpretations. However, LIME provides local interpretability by adjusting inputs and monitoring the effect on output, allowing neurologists to understand specific predictions on a case-by-case basis. Together all these techniques transform the opaque nature of CNNs into outputs that are understandable and appropriate for clinical interpretation.

The system is evaluated using common performance metrics such as precision, sensitivity, specificity, and accuracy. In order to assess learning behaviour, plots are also used to visualise training and validation performance. In addition to providing the transparency required for clinical acceptance, this proposed system combines deep learning and explainable models to achieve high seizure detection accuracy. Its modular, automated pipeline, which runs from data ingestion to explainable output, makes it a promising tool for real-time, patient-independent seizure monitoring and diagnosis support.

2.3 Epileptic Disorder Detection of Seizures Using EEG Signals

- Uncontrolled brain electrical activity which results in an excessive and rapid neuronal discharge is known as epilepsy. For the diagnosis and treatment of epilepsy the EEG signals that measure electrical activity from the nerve cells in the cerebral cortex have drawn a lot of attention. In order to improve the seizure detection accuracy this study suggests an EEG-based seizure identification system that addresses data imbalance issues using the CHB-MIT dataset and makes use of deep learning models like Bi-LSTM, 1D-CNN, and attention mechanisms.

- This study integrates data from the CHB-MIT dataset, which provides us with comprehensive EEG recordings from patients diagnosed with epilepsy. This dataset has been meticulously supplemented with labels that differentiate between pre-seizure (pre-ictal), seizure (ictal), post-seizure (post-ictal), and non-seizure intervals. Such comprehensive labelling is necessary to train the deep learning model to correctly identify and classify seizure events across the various stages of seizure progression. This structured annotation helps the model learn temporal transitions and improves the model's performance in real-time seizure detection scenarios.

- During the data preprocessing step the EEG signals are divided into 10-second windows with a 1-second overlap especially for seizure segments in order to guarantee constant input sizes suitable for neural network training. On the other hand, non-seizure data segments are kept non-overlapping to minimise redundancy and avoid imbalance in the training dataset. To enhance the signal quality and spot the important patterns the EEG signals are divided into five distinct frequency bands using a multi-level Discrete Wavelet Transform (DWT): delta, theta, alpha, beta, and gamma. This decomposition effectively separates features associated with different brain states while reducing noise, improving the accuracy and reliability of the feature extraction process.

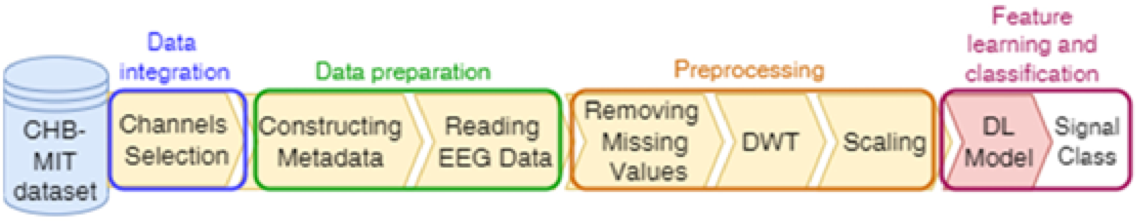


Figure 2.1: Proposed compatibility framework architecture.

- The proposed deep learning model integrates recurrent layers such as CNNs and Bi-LSTM with attention mechanisms to effectively capture both temporal and spatial patterns in EEG data. The 1D-CNN is the primary feature extractor; it scans the EEG time series and produces feature maps that highlight significant spatial features. Max pooling is applied to all these feature maps to minimise overfitting, lower dimensionality, and draw attention to important features. The CNN layer is followed by bi-LSTM layers which model temporal dependencies in the EEG signals by processing data both forward and backward. This bidirectional approach improves the model's sensitivity to sequential patterns which are crucial for accurately identifying different seizure phases. Finally, fully connected (dense) layers are used to combine the extracted features.

- With an accuracy of 96.87% and a sensitivity of 96.85%, the model performed noticeably better than other models evaluated on the CHB-MIT dataset. These results show that the model performs well and consistently detects seizure events even with fewer EEG channels. This method's high sensitivity and accuracy demonstrate how well it captures relevant patterns associated with epileptic seizures, suggesting that it could be used in clinical settings for seizure monitoring and diagnosis.

$$\text{Sensitivity (recall)} = \frac{TP}{FN + TP}$$

$$\text{Precision (PRC)} = \frac{TP}{TP + FP}$$

$$\text{Accuracy (ACC)} = \frac{TP + TN}{\text{Total Samples}}$$

EXP No.	DB	Avg. Epoch ACC	Avg. Epoch Sen. for Seizure	Avg. Epoch Sen. for No-Seizure	Avg. Epoch PRC for Seizure	Avg. Epoch PRC for No-Seizure
1	CHB-MIT	79.25	64.16	93.14	89.2	75.29
2	CHB-MIT	81.93	68.43	94.41	91.54	78.03
3	CHB-MIT	75.38	54.95	94.02	89.26	70.53
Avg.	**CHB-MIT**	78.85	62.51	93.86	90	74.62

Figure 2.2: Comparison on the performance of the DL model with and without data integration using CHB-MIT dataset.

- This framework, leveraging the CHB-MIT dataset, addresses challenges in seizure detection by enhancing data balance and capturing temporal dependencies in EEG signals. By incorporating wavelet decomposition, 1D-CNN, Bi-LSTM, and attention mechanisms, the proposed system achieves high sensitivity and accuracy. This model has the potential to reduce the workload on clinical professionals, providing timely and accurate detection of seizures and improving treatment response times.

2.4 Patient-Independent Epileptic Seizure Detection Using Explainable AI Techniques

- Epilepsy is one of the major neurological conditions, and more than 65 million people suffer from this disorder worldwide; generally, seizures require monitoring in real time for proper seizure detection. This method introduces a new patient-independent seizure detection avenue through the integration of deep learning with XAI, helping to close the interpretability gap. The approach utilizes EEG data reflecting frequency bands, positions of electrodes, and temporal features for the advancement of predictability capabilities. The proposed system aims at producing interpretable results for medical experts in 14 helping the diagnostic and treatment processes by identifying the relevant EEG regions across spectral, spatial, and temporal dimensions. In that direction, it applies two CNN- based methods of detection: 1D-CNN and 3D-CNN, along with SHAP-based explanations presented in heatmap forms to better provide time efficiency and reliability for clinical decision-making purposes. [5]

- The proposed method uses EEG data from multichannel recordings that contain seizure and non-convulsive intervals, which are further separated into subsequences for analysis. These EEG segments are filtered in the spectral domain because the frequency range of 0.5 Hz to 12.5 Hz is optimal for identifying seizures in newborns. A segment is categorised as a seizure if at least one of its subsequences is identified as such. There are two primary parts to the deep learning architecture. By converting the EEG intervals into frequency domain features the 1D-CNN (Detector 1D) carries out spectral analysis. This model uses dropout and L2 norm for regularisation and consists of three convolutional layers with filter sizes of 64, 128 and 256, respectively. The purpose of these features is to precisely identify seizure patterns by capturing changes in the power spectrum over time. In contrast the 3D-CNN (Detector 3D) uses Morlet wavelet transformations to transform EEG data into time-frequency maps which are then assembled into a 3D structure while preserving spatial information. In order to efficiently capture the spatiotemporal features of the EEG signals it also uses three convolutional layers with progressively larger filter sizes of 32, 64, and 64. SHAP-based post-hoc explanations are used for interpretability and provide visualisations of the contributions of spectral, spatial, and temporal features. These are displayed as heatmaps that correspond to the locations of the EEG electrodes and provide easily understood insights into the model's decision-making process, which helps with clinical comprehension.

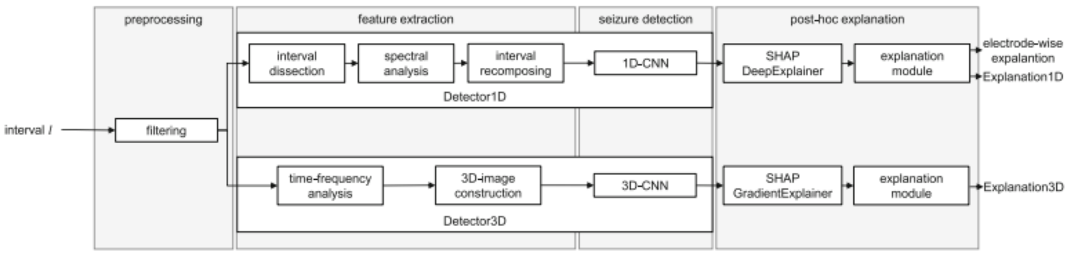


Figure 2.3: Architecture diagram of 1D CNN and 3D CNN models

- **Formulae and Metrics Used**
  - **Morlet Wavelet Transform:** EEG signals are transformed using the Morlet wavelet for time-frequency analysis:

    $$\psi(t) = \frac{1}{\sqrt{s}} e^{j2\pi f_0 t} e^{-\frac{t^2}{2\sigma^2}}$$

    where s is the scale parameter, f_0 is the central frequency, j is the imaginary unit, and σ is the width of the wavelet.
  - **Welch's Method for Power Spectral Density:** Used to calculate the power spectral density (PSD) of EEG signals:

    $$P_{xx}(f) = \frac{1}{L} \sum_{i=1}^{L} |X_i(f)|^2$$

    where $P_{xx}(f)$ is the PSD, L is the total number of segments, and $X_i(f)$ represents the Fourier transform of each segment.
  - **SHAP Value Calculation for CNN Predictions:** Model interpretation and determining each feature's contribution to predictions are done using SHAP values:

    $$\phi_i = \sum_{S \subseteq N \setminus \{i\}} \frac{|S|!(|N| - |S| - 1)!}{|N|!} [f(S \cup \{i\}) - f(S)]$$

    where ϕ_i is the SHAP value for feature i, S is a subset of features, and $f(S)$ is the model prediction with subset S.
  - **Performance Metrics:**
    * Sensitivity $= \frac{TP}{TP+FN}$
    * Specificity $= \frac{TN}{TN+FP}$
    * Precision $= \frac{TP}{TP+FP}$
    * Balanced Accuracy $= \frac{Sensitivity+Specificity}{2}$

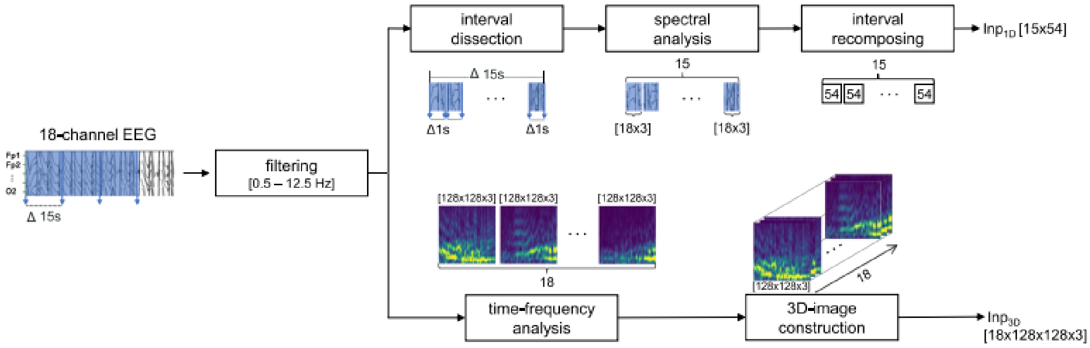


Figure 2.4: EEG preprocessing and feature extraction

- The method introduces an innovative explainable seizure detection approach that balances interpretability with model accuracy. Visual feature contribution mapping appears to be a design configuration that proved effective for clinicians and shows promise for broader clinical adoption and possibly further advancement into patient-independent seizure monitoring. The model could capture both spatial and temporal information with high specificity and for complex seizure patterns.

2.5 "Why Should I Trust You?": Explaining the Predictions of Any Classifier

- By locally approximating the predictions of any machine learning model using an interpretable model, XAI method LIME can faithfully explain the predictions.[6]
- Two systems of data representation are followed throughout this explanation: the Feature Representation and the Interpretable Data Representation.
  - Feature Representation - The feature representation form presents complex raw data, encompassing all the features or attributes that the model processes. This representation is optimized for Machine Learning (ML) models and will be difficult for humans to comprehend.
  - Interpretable Representation of Data - A condensed, comprehensible version of the feature representation is used to represent interpretable data. The main points of the data are highlighted in this representation, while unimportant details are left out. Creating understandable justifications for intricate model predictions requires this type of representation.

$$\mathbf{x} \in \mathbb{R}^d \quad \underset{\mathbf{x} = h_x(\mathbf{x}')}{\overset{x' = h_x^{-1}(\mathbf{x})}{\rightleftarrows}} \quad \mathbf{x}' \in \{0,1\}^{d'}$$

here,

- $\mathbf{x} \in \mathbb{R}^d$ is the feature representation of the data, where $\mathbb{R}$ is the set of real numbers and d denotes the dimensionality of the feature vector.
- $\mathbf{x}' \in \{0,1\}^{d'}$ is the interpretable data representation, where $\{0,1\}$ is a binary set indicating feature presence (1) or absence (0) and d' denotes the dimensionality of the interpretable data vector.

- A LIME explanation is an interpretable model.The representation of the interpretable model is $g \in G$, where G is the set of interpretable models. These interpretable models provide valuable insights into the predictions of the complex model f. The LIME equation is expressed as

$$\xi(x) = \arg\min_{g \in G} L(f, g, \pi_x) + \Omega(g) \tag{2.1}$$

Here,

- $\xi(x)$ represents the LIME explanation for the instance x.
- $L(f, g, \pi_x)$ is the loss function (or fidelity function) which shows how well the complex model f can be approximated by the interpretable model g in the locality defined by π_x.
- $\Omega(g)$ is the complexity function, which ensures that the interpretable model is not overly complex.

In the immediate region surrounding the instance x, the LIME optimisation problem aims to find g, an interpretable model that closely resembles the behaviour of the complicated model f, while making sure that g stays simple and doesn't get too complex.

The LIME expression can be adapted to work with different types of interpretable models G, loss functions L, and complexity measures Ω.

- The local behavior of the complex model f is explained using the interpretable model g, while retaining its model-agnostic nature, LIME generates perturbed samples of instance x, which are then used in training the interpretable model g. This process involves:

  1. Perturbed Sampling - Samples are generated around instance x by randomly altering some feature values in its feature vector.
  2. Weighing by Locality - The generated samples are assigned weights based on their proximity to x using an exponential kernel π_x, with samples closer to x receiving higher weights.
  3. Building Perturbed Sample Set - Each sample, along with its weight and the prediction from the complex model, is added to the set of perturbed samples Z.

  Using this dataset of perturbed samples Z, LIME produces the interpretable model g which can locally explain the complex model f.

- Sparse linear explanations are interpretable models g that explain the complex model f by utilizing only a small subset of the features. By enforcing sparsity these explanations focus only on the most relevant features, ensuring simplicity and interpretability.

  For G, taken as the set of linear models, the loss function $L(f, g, \pi_x)$ and complexity function $\Omega(g)$ can be defined as:

$$L(f, g, \pi_x) = \sum_{z,z' \in Z} \pi_x(z) \left(f(z) - g(z')\right)^2 \tag{2.2}$$

  Here,

  - The proximity measure $\pi_x(z)$ allocates weights to perturbed samples according to their distance from instance x, with closer samples being given a higher weight.

– $(f(z) - g(z'))^2$ is the squared difference between the complex model's prediction $f(z)$ and the interpretable model's prediction $g(z')$

$$\Omega(g) = \lambda \cdot \sum_{i=1}^{k} |w_i|^0 \tag{2.3}$$

Here,

– λ is a regularization parameter that balances the trade-off between fidelity and interpretability. A large value of λ prioritizes interpretability, while a smaller λ value emphasizes fidelity.

– $\sum_{i=1}^{k} |w_i|$ represents the total complexity of the model, where w_i are the coefficients (or weights) of the features.

- LIME Explanation Algorithm

Algorithm 1 LIME Algorithm

Require: Classifier f, Number of samples N

Require: Instance x, and its interpretable version x'

Require: Similarity kernel π_x, Length of explanation K

1. $Z \leftarrow \{\}$
2. **for** $i = 1$ to N **do**
   (a) $z'_i \leftarrow \text{sample\_around}(x')$
   (b) $Z \leftarrow Z \cup \{z'_i, f(z_i), \pi_x(z_i)\}$
3. **end for**
4. $w \leftarrow$ K-Lasso(Z, K) {with z'_i as features, $f(z)$ as target}
5. **return** w

- **Step-by-Step Application of LIME**

1. Instance Selection - The process begins by selecting an instance x from the dataset for which an explanation is desired.

2. Perturbed Sample Generation - Next, perturbed samples of the instance x are generated and labeled using the complex model f.

3. Fit Interpretable Model - After generating perturbed samples, LIME fits an interpretable model g by training it on the perturbed sample dataset.

4. Output Explanation - Finally, the ideal interpretable model g that best satisfies the LIME equation is produced as the explanation for the instance.

2.6 Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization

- Convolutional neural networks, or CNNs, perform well on tasks like object detection, picture categorization, and visual question answering, but their intricate inner workings have created a "black-box" problem. This lack of interpretability means it's challenging for users to understand why a certain prediction was made by the model, which could be a problem, especially in applications like healthcare, security, or self-driving. Making models interpretable allows users to see why a model made specific predictions. This helps in detecting where the model might fail, identifying biases in the dataset, building trust with users, and improving the design of the model. Grad-CAM builds on CAM by creating visual explanations that work across different architectures without requiring structural changes. By using the gradients flowing into the last convolutional layer, Grad-CAM reveals biases and failure mechanisms in CNN architectures by highlighting areas of an image that affect the prediction for a specific notion.[7]

- The methodology in Grad-CAM

1. **Grad-CAM (Gradient-weighted Class Activation Mapping)**

   Grad-CAM calculates "importance weights" by looking at a target class's gradients (e.g., 'dog') score based on feature maps from the model's last convolutional layer. The process generates a "heatmap" that highlights regions critical to the model's prediction.

   Formulation: Grad-CAM, in the case of a target class c, calculates the gradients of the score y^c (for class c) based on feature maps A^k. This gives weights α_k^c through global average pooling of gradients:

$$\alpha_k^c = \frac{1}{Z} \sum_i \sum_j \frac{\partial y^c}{\partial A_{ij}^k}$$

Using these weights, the localization map $L_{\text{Grad-CAM}}^c$ is created:

$$L_{\text{Grad-CAM}}^c = \text{ReLU}\left(\sum_k \alpha_k^c A^k\right)$$

Only features that have a positive impact on the class c prediction are highlighted by the ReLU.

2. **Guided Grad-CAM**

   Grad-CAM can be combined with Guided Backpropagation to achieve finer detail and class-specific focus. This results in "Guided Grad-CAM," a high-resolution and discriminative visualization.

3. **Applications**

   – **Weakly-Supervised Localization:** Tested on ImageNet's localization challenge, Grad-CAM successfully identified relevant regions without needing bounding box labels.

   – **Image Captioning and VQA:** Grad-CAM highlights regions related to specific words or answers, even when the model doesn't use attention mechanisms.

   – **Counterfactual Explanations:** By slightly altering Grad-CAM, it can show which image regions could change the prediction if removed or altered.

- The following figure shows the Grad-CAM workflow. A raw category score is generated by forward-propagating an input image via a CNN with a target class (such as "tiger cat"). The signal is then backpropagated to the convolutional feature maps after gradients for all classes other than the target are set to zero. These gradients are used to create a crude localization map, or heatmap, which shows the areas the target class model concentrates on.

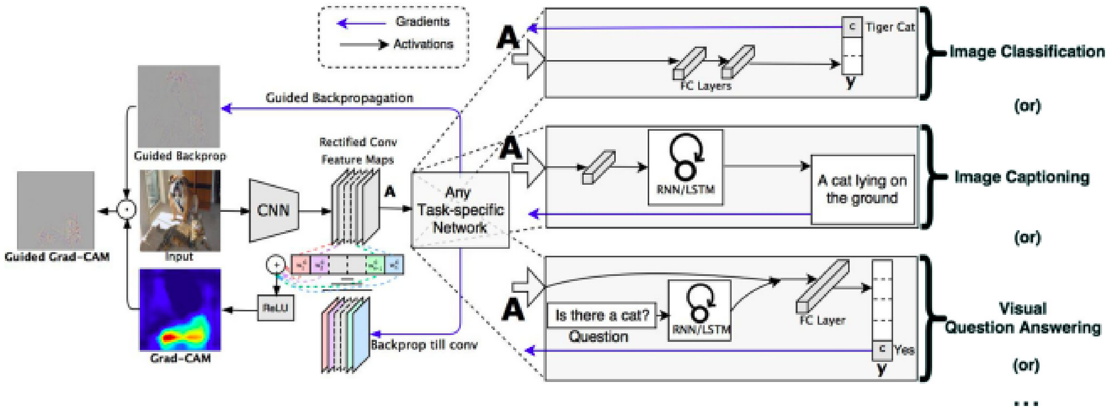


Figure 2.5: Grad-CAM Workflow Diagram

- Grad-CAM's quantitative evaluations show how effective it is at a variety of tasks. When used with CNN architectures like VGG-16 and GoogleNet in the ImageNet localisation challenge, it produced accurate localisation with reduced error rates and no need for model retraining. By successfully emphasising significant features, Grad-CAM enhanced Intersection over Union (IoU) scores for segmentation tasks on the PASCAL VOC dataset. Furthermore, user studies verified that Grad-CAM and Guided Grad-CAM improved faithfulness and trust, visualising the areas of focus to make model predictions easier to understand. Grad-CAM was useful for failure mode analysis on the qualitative side, showing that even inaccurate predictions frequently concentrate on semantically relevant areas, like a background that resembles the target object. Additionally, it remained interpretable in the presence of adversarial noise, in which the model was intended to be misled by tiny image perturbations. Additionally, by revealing the model's dependence on facial features, Grad-CAM assisted in identifying bias in a study involving the classification of "doctor" versus "nurse," indicating the existence of gender bias in the training data.

- Grad-CAM provides class-specific visual explanations across diverse CNN architectures and tasks without needing model modifications. This is done by using gradients to deliver straightforward, computationally efficient visualizations.Grad-CAM improves model transparency, which can enhance user trust, help diagnose model weaknesses, and highlight biases. Additionally, it works across tasks such as image captioning and VQA, showing its versatility. Future directions include extending Grad-CAM to non-visual tasks, such as language models, and further improving its faithfulness in representing model decisions. There's also potential to use Grad-CAM in developing automatic bias detection tools, given the ethical consequences of biased AI models.

2.7 Summary and Gaps Identified

2.7.1 Gaps in the current state-of-art

- **Real-Time Application Limitation:** The existing seizure detection models are not feasible for practical application in a clinical environment because of high computational complexity.

- **Explainability:** Deep learning models such as CNNs gain very high accuracy but cannot be explained like a black box, making it difficult for neurologists to trust and adopt the seizure detector.

- **Lack of Generalization:** Those models trained on limited datasets fail to generalize to diverse patient populations or varying EEG recording settings. Their real-world effectiveness is comparatively reduced in this case.

- **Scalability Problems:** Many methods are computationally extensive and suffer a problem with scalability on large data sets or real deployments on resource-constrained settings like portable EEG devices.

- **Lack of Integration with Explainable AI (XAI):** Though SHAP, LIME, and Grad-CAM do exist, there is still a scarcity of research focused on systematic integration of such techniques with seizure detection models for the ultimate goals of accuracy and interpretability.

- **Poor noise and Artifact handling:** Noises are typically involved in the EEG data; current models cannot handle the artifacts effectively, and so can prove to be a matter of reliability in clinical practices.

2.7.2 Summary

Table 2.1: Comparison of XAI Methods for Seizure Detection and Model Explanation

Paper	**Advantages**	**Disadvantages**
XAI4EEG: Spectral and Spatio-Temporal Explanation of Deep Learning-Based Seizure Detection in EEG Time Series	Provides consistent as well as accurate attributions. Model-agnostic; works with any machine learning model. Offers both local and global interpretability.	Computationally costly for intricate models or big datasets. Scaling for real-time applications is challenging. Interpreting high-dimensional data can be difficult.
"Why Should I Trust You?": Explaining the Predictions of Any Classifier	Model-agnostic; can explain any machine learning model. Fast and efficient for local explanations. Provides simple and interpretable explanations	Sensitive to sampling. Struggles with explaining highly complex or non-linear models. Limited global interpretability; focuses mainly on local explanations.
Mask-GradCAM: Object Identification and Localization of Visual Presentation for Deep Convolutional Network	Highly effective for visualizing deep learning models, particularly CNNs. Produces class-specific heatmaps to show important regions in input data.	Limited generalization to other model types. Resolution of heatmaps can be coarse. Difficult to apply to non-visual tasks like text or tabular data

Chapter 3

Requirements

3.1 Hardware requirements

- Minimum: Intel i5 Processor, 8GB RAM, and 256GB SSD for basic model training.
- Recommended: NVIDIA GTX 1060 or better for accelerated deep learning training.
- Storage: Large enough storage to accommodate EEG datasets (at least 400GB).
- Cloud Option: Google Colab or AWS for cloud-based model training using GPUs.

3.2 Software requirements

- Operating System: Windows 10, Linux, or macOS.
- Programming Language: Python 3.7+.
- Libraries:
    - TensorFlow/Keras: For CNN model building.
    - SHAP, LIME: For explainability.
    - Matplotlib and Seaborn: For visualization of model performance.
    - Scikit-learn: For model evaluation.
    - IDE: Visual Studio Code, Jupyter Notebook, or PyCharm for development.

3.3 Budget

Items	Amount (Rs)
Software/Services	
Google Colab Pro+ (GPU for model training)	Rs. 19,793.41 (60/month × 4)
Google One Premium (Storage for datasets)	Rs. 3,250 (650/month × 5)
Hostinger (Web Hosting)	Rs. 1,245 (249/month × 5)
Miscellaneous (Additional compute units for Colab, etc)	Rs. 3,000 (10/100 units)
Publication charges	Rs. 10,000
Travel/Contingency	Rs. 1,000
Total	**Rs. 38,288.41**

Table 3.1: Budget

Chapter 4

System Architecture

The chapter gives an in depth clear understanding of the components in the system, the technological frameworks that were used in it and the techniques that were implemented.

4.1 System Architecture

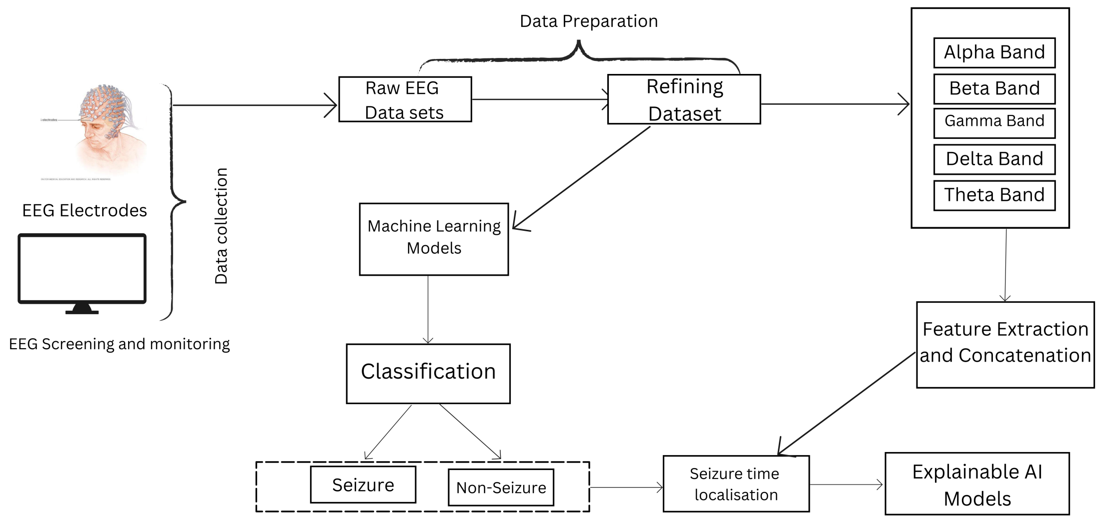


Figure 4.1: Architecture diagram

The architecture diagram for automated epileptic localisation, localisation od the seizure time and interpretability of the output of the CNN model is depicted in fig.4.1. The EEG data is collected in .edf format using an electroencephelogram. A 10-20 international standard electrode system is attached to the scalp of the patient after marking the scalp. The brain signals through each of the 23 channels are collected and saved. The data collected has a sampling rate of 256 Hz which means that each second is further divided into 256 samples. The first step after data collection is data preprocessing. The collected

EEG data is prone to noise. The dataset is cleaned from the occurrence of Nan values, extra electrode data as well as ECG data if present. After the preprocessing stage, the brain signals are divided into discrete frequency bands alpha, beta, theta, delta and gamma based on their frequency ranges. This converts the temporal domain information into frequency domain. This is crucial for understanding the behaviour of the brain waves to extract patterns suggestive of seizure activity.

For each of the frequency band that were extracted, power spectral density was calculated. The features and results thus observed are saved for incorporation into seizure time localisation module. The CNN model is trained on annotated EEG data which is expected to categorise a given data sample into seizure or non seizure event. Data points that were identified as seizure points are processed in the time localisation module to obtain the start time and end time of the seizure. The module employs the use of statistical methods to pinpoint the precise time intervals of seizure activity. Automated threshold based analysis is done over several time periods to increase the accuracy of the localisation.

The architecture also incorporates the use of a XAI module which is primarily composed of two methodologies - SHAP and LIME. These serve as explainability modules to bring interpretation to the decision made by the CNN model. Since CNN models are black box models, it doesnot specify the decision making process of the model. But the incorporation of such models into critical fields such as medical industry, the explanations are required to build trust in the model. The pipeline is therefore not only accurate but also interpretable and therapeutically meaningful due to the integration of statistical seizure time prediction and post-hoc XAI, assisting neurologists in diagnosis and treatment planning without seeking to replace them.

4.1.1 Front-end (Python GUI)

The frontend of the system consists of a Python based GUI which is used as an interface for the neurologists to interact with the system efficiently. The GUI provides a feature that allows the users to upload the raw EEG data in '.edf' format thus making the usage of the system more user friendly. It also helps in visualising line graphs corresponding to each of the electrode thereby allowing the user to visualise either single or combinations of electrodes. The system also consists of a feature to visualise the uploaded EEG data in '.csv' format. The GUI also dislays the seizure detection results as predicted seizure intervals within the files along with the visualisations in the form of plots produced by SHAP and LIME. This GUI interface is developed using Python libraries such as Tkinter, PyQt or Kivy whereas the visualisations are generated using Matplotlib or Seaborn.

4.1.2 Back-end

The back-end of the project manages data processing and facilitates interaction between the front-end, storage, and machine learning modules. It includes features such as converting .edf files to CSV format and preprocessing EEG data for further analysis. The back-end also interfaces with the machine learning model to enable seizure detection and localization. Additionally, it handles file uploads and downloads from Google Drive. Technologies used in the back-end include Python, along with libraries like Pandas and NumPy for efficient data preprocessing.

4.1.3 Database (Google Drive)

Currently, Google Drive is used as the storage medium for managing as well as organising the large EEG data and results being generated. The EEG data being used was obtained from the publicly available dataset by University of Bonn on pediatric seizures. The database stores the raw EEG data in the form of '.edf' and the corresponding processed '.csv' files ensuring easy retrieval and usage. Additionally, it maintains logs of the model's performance, performance metrics as well as the outputs generated by the explainable AI models. This helps in keeping a comprehensive report on the progress and performance of the model. The back end works as a connection channel between the front end where the raw EEG data is uploaded and the Machine learning models that generate the predictions from the model, thereby ensuring a smooth management as well as working of the system.

4.1.4 Machine Learning Model

The patient specific CNN-based architecture, which was created to locate the patient's seizure occurrences, is the main part of the system. It takes the preprocessed EEG data in matrix format as the input. A ratio of 70%-20%-10% is maintained while splitting the dataset for testing, training as well as validation purposes. The model uses the annotated EEG data to learn seizure and non-seizure occurrences during the training phase. The model localises the seizure occurrences as probable seizure time segments and thereby providing insights into the potential seizure events. The model development as well as training are carried out using Tensorflow with GPU acceleration to ensure faster training of the model and thereby enhancing the overall performance and efficiency of the model being developed.

4.1.5 XAI Models

In order to interpret CNN predictions and offer insight into the model's decision-making process, the explainability module integrates XAI approaches. It utilizes SHAP to quantify the contribution of individual features, such as EEG channels, to the predictions of the model. LIME is employed to explain local predictions by fitting interpretable surrogate models. The output consists of explanation summaries that are displayed in the GUI, allowing users to build trust and understand the model's predictions. These techniques are implemented using libraries like SHAP and LIME in Python.

4.2 Component Design

1. Data Preprocessing

- File Conversion : This converts the raw EEG data stored in '.edf' format into '.csv' format for easier usage and interpretation.
- Matrix Creation : The EEG signals are extracted from '.edf' format and represented in the form of matrices where the rows represent the time samples denoted in 256 sampling rate and columns represent the various channels. This format serves as the input that is given to the CNN model,
- Filtering : Noise as well as irrelevant frequencies are removed. This ensures that only the frequency bands relevant for seizure localisation are passed through.
- Annotation Integration : The '.csv' files that are created are annotated by assigning corresponding seizure labels where 1 refers to a seizure event whereas 0 represents a non seizure event. This faciliatates in having supervised learning in the CNN model.

2. Feature Extraction

The EEG data, which were initially captured in the time domain, are first converted into the frequency domain using Power Spectral Density (PSD) estimation in order to extract significant patterns linked to seizure onset. The frequency components of the signal can be more clearly analyzed thanks to this conversion, making them more useful for identifying changes linked to seizures. Ten one-second ictal frames make up each seizure segment, while thirty one-second preictal frames reflect the previous time. Frequency-specific properties, specifically the relative alpha power and relative delta power, which show the percentage of energy in these bands compared to the total power across all bands, are computed for each of these frames. These characteristics are helpful in spotting distinctive patterns, such as the alpha slowing phenomena, which is frequently seen during seizures. Critical inputs for seizure categorization and temporal prediction tasks are the derived frequency-domain features.

3. CNN Model

- Architecture Design : To understand the temporal and spatial patterns in the EEG signals, patient specific 2D Convolutional Neural Network (CNN) architectures are developed.

- Training Data : EEG data is used for training by the CNN model. The EEG data used has been preprocessed and annotated to classify as seizure or non seizure events.

- Hyperparameter Tuning : The key parameters such as number of layers, batch size and learning rate are optimised to constantly improve the performance of the model.

- Regularization : To prevent overfitting on the given input data, dropout layers are implemented. This ensures generalisation over unseen data.

- Output : To indicate the likelihood of seizure events a probability score is generated and is categorised on the basis of the sigmoid function.

4. Seizure time localisation

Autocalibartion which is a statistical method, is used to dynamically adjust to the unique EEG patterns of individual patients. It can be used to precisely localise the start time and the end time of a seizure. In contrast to fixed threshold methods which fail to generalise over different patients, the autocalibration method learns the patient's baseline EEG patterns and then personalises the prediction process. The technique uses Median Absolute Deviation(MAD) and Gaussian Mixture Models(GMM) to find the unique patterns that could be suggestive of a seizure onset. GMM is used to understand the distribution of features such as relative delta power and hence analyse small shifts or changes that occur thoughout the preictal phase of a seizure. MAD on the other hand first removes noise from the data and makes sure that it only records the significant shifts in EEG patterns. Auto-calibration method thereby accurately detects intervals of seizure occurences by examining the temporal patterns of the derived characteristics over sliding windows of one second each. By recording the dynamic evolution of brain activity before and during seizures, this enhances the accuracy of time prediction and supports real-time clinical monitoring and early warning systems.

5. Explainability

- SHAP (SHapley Additive exPlanations) : SHAP provides global as well local explainability to the predictions of the model. This helps in recognising the most important features for detecting the seizure events. Shapley values are used to determine each input feature's contribution to the final prediction of the model. A positive shapley value indicates a positive contribution to the model's output, whereas a negative shapley value indicates a negative contribution.

- LIME (Local Interpretable Model-agnostic Explanations) : For every prediction the model makes, LIME produces locally interpretable explanations. This is accomplished by altering the input data and then monitoring how it affects the model's output. This technique helps in understanding why the model predicted in a certain manner for a specific datapoint in the EEG data.

4. Evaluation

The model's performance is assessed using accuracy, which measures the overall percentage of correctly classified data points in the EEG recordings. A higher accuracy indicates that the model effectively distinguishes between seizure and non-seizure states. This metric provides a straightforward evaluation of how well the thresholding approach optimizes seizure detection, ensuring that the classification aligns closely with actual seizure events in the data.

4.2.1 Interaction between the modules

1. User Interaction with the System

   The user logs in to the software and uploads the raw EEG data in '.edf' format.

2. Data Preprocessing Module

   The uploaded EEG data is processed in this module.

   - Conversion : The raw EEG data in '.edf' format is converted to the standard CSV format. In CSV format the row represents the time samples which is denoted with a sampling rate of 256Hz and the columns represent the EEG channels.
   - Filtering and Normalization : The signals are filtered and normalised to remove noise such as empty columns, unneccessary brain wave frequencies which helps in enhancing the quality of the CSV data which is used as input for the CNN model.
   - Labeling : The timesamples are annotated with 0 or 1 indicating non seizure and seizure events which enables supervised learning.

3. Feature Extraction module

   - The EEG signal is preprocessed and then divided into fixed one-second frames for in-depth temporal analysis. Features are taken out of these frames in order to identify seizure activity.

   - Frame Segmentation: 30 one-second preictal frames (before to the commencement of a seizure) and 10 one-second ictal frames (during a seizure) are produced for every seizure event.

- Calculation of Relative Band Power: Each frame's relative delta and alpha power are computed. Because it offers important insights into seizure dynamics, the phenomena of alpha slowing—where alpha activity decreases during seizure—is given particular consideration. In the next step, time localization depends on these retrieved features.

4. CNN Model Module

   The preprocessed EEG data that has been removed of unnecessary noise and frequencies along with the annotations is given as input into the patient specific CNN models

   - Training Phase : During the training phase the CNN model learns the patterns which are indicative of the presence of seziures in the file. For this supervised learning is implemented using the annotated data.
   - Inference Phase : The trained CNN model predicts the time samples as seizure (1) or non seizure (0). The model uses a probability score to identify these events. A sigmoid function is used as the benchmark for this classification.

5. Seizure time localisation

   A reliable statistical time prediction method known as auto-calibration is utilized to pinpoint the precise beginning and length of seizures.

   - Auto-Calibration: This technique uses Median Absolute Deviation (MAD) in conjunction with Gaussian Mixture Models (GMM) to identify statistically significant variations in the EEG characteristics, including alpha and delta power.

   - Dynamic Baseline Adaptation: This method improves resilience among patients by dynamically adapting to individual EEG baselines, in contrast to fixed-threshold methods. The module accurately calculates seizure start and offset intervals by modeling feature distributions and detecting aberrant shifts.

6. Explainability Modules

   - This stage interprets the predictions made by the CNN model to underatnd why a given prediction was made and the features that contributed to the prediction:

- SHAP : It calculates the contribution made by each EEG channel to the final decision made by the model. This helps in providing local and global explanaions.
- LIME : The input EEG data is perturbed to create multiple data points around the actual data point. The impact of this perturbed data on the prediction of the model is then analysed. This offers a simple and interpretable explanation to the individual predictions of the model.

The outputs of these techniques are combined and sent to the next module.

7. Results Interpretation Module

This module combines the time interval predictions made by CNN model and the outputs from explainability techniques. The results are formatted into user-friendly visualizations, such as:
- Plots such as summary plots, force plots and beeswarm plots for visualising the outputs of SHAP and LIME.
- Charts which summarize the performance metrics such as accuracy, precision, sensitivity, etc..
- Tables which show the seizure events that have been identified by the model.

4.3 Sequence Diagram

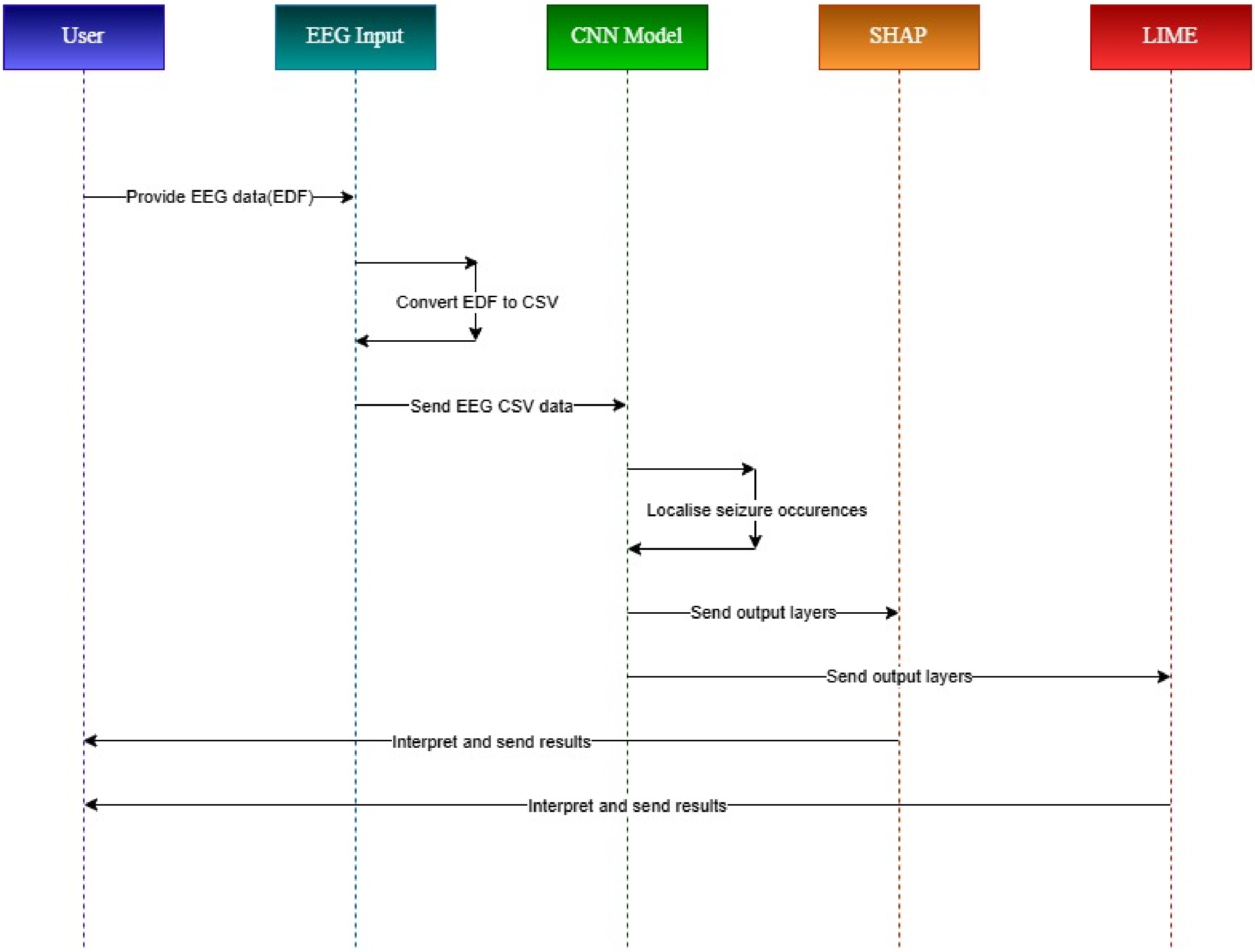


Figure 4.2: Sequence diagram

4.4 Tools and Technologies

4.4.1 Software requirements

- Operating System: Windows 10, Linux, or macOS.
- Programming Language: Python 3.7+.
- Libraries:
  - TensorFlow/Keras: For CNN model building.

- SHAP, LIME: For explainability.
- Matplotlib and Seaborn: For visualization of model performance.
- Scikit-learn: For model evaluation.
- IDE: Visual Studio Code, Jupyter Notebook, or PyCharm for development.

4.4.2 Hardware requirements

- Minimum: Intel i5 Processor, 8GB RAM, and 256GB SSD for basic model training.
- Recommended: NVIDIA GTX 1060 or better for accelerated deep learning training.
- Storage: Large enough storage to accommodate EEG datasets (at least 400GB).
- Cloud Option: Google Colab or AWS for cloud-based model training using GPUs.

4.5 Data set Identified

EEG recordings from 22 juvenile individuals with intractable seizures (5 boys aged 3–22 and 17 females aged 1.5–19) are categorized into 23 cases in the CHB-MIT dataset, which was gathered from the Children's Hospital at Boston. Each '.edf' file contains one hour of digitized EEG data, and the signals typically have 23 EEG channels with a sampling rate of 256 Hz. The International 10-20 system of EEG electrode placements and naming was used for the recordings.

4.6 Module Divisions and work break down

- Feature extraction - Rose Jacob, Powell Moothedan
- CNN model - Niyatha V S, Paul Allen Kadayaparambil
- Automated time prediction - Niyatha V S
- Shapley Additive Explanations - Niyatha V S
- Local Interpretable Model-agnostic Explanations - Paul Allen Kadayaparambil
- GUI - Powell Moothedan

4.7 Key Deliverables

- Fully Automated Seizure Localization: To achieve a seizure localisation accuracy of over 95% sing deep learning models. This is done by minimising the false positive and false negative count.
- Explainable AI : Using SHAP and LIME, the CNN model's predictions will be explainable and interpretable, guaranteeing an open, transparent, and medically reliable decision-making process.
- Improved Clinical Decision Making : To aid neurologists in diagnosing and localising seizures fater and more accurately thereby reducing the manual labour that the lab technicians have to put in.
- Feature Extraction :To use the spatial and temporal characteristics of the EEG data to extract meaningful features from it. This faciliatates in cross verifying the seizure localisation that is done by the CNN model.
- Integration with Clinical Usage : To design a trustworthy and user friendly system.
- Ethical AI Deployment : To ensure that the system developed adheres to ethical guidelines as well as HIPAA regulations.

4.8 Project Timeline

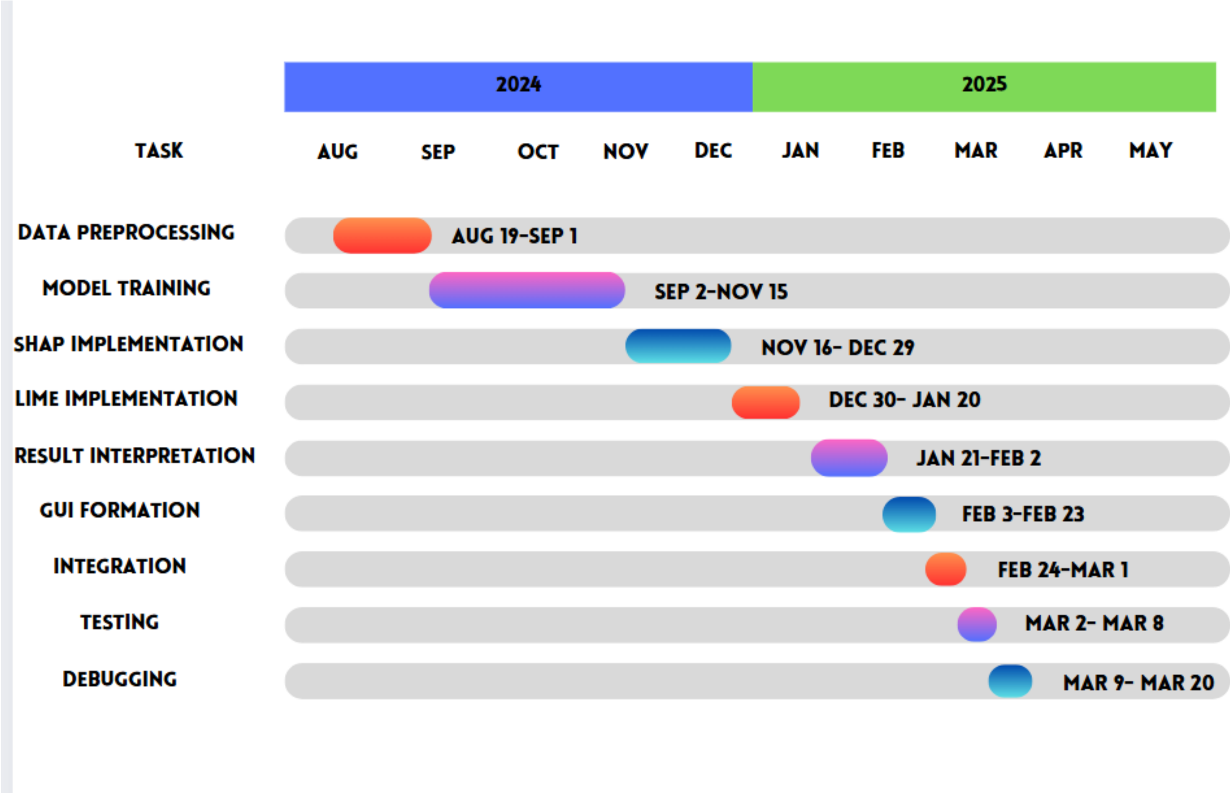


Figure 4.3: Project Timeline as a Gantt Chart

4.9 Conclusion

This chapter gave a thorough rundown of the system architecture, emphasizing the project's structural elements and underlying design. As explained in the component design section, every component was painstakingly created to guarantee scalability, effectiveness, and usability. The sequence diagram highlighted the logical flow and integration of functionality while providing a clear visual representation of the interactions between the system's many components. The technologies and tools employed were chosen with care to meet the project's specifications, guaranteeing compatibility and peak performance. These technologies were essential in turning abstract designs into a reliable and useful system.

Lastly, the chapter's main deliverables highlighted the observable results of the development process and showed advancement toward the project's overarching goals. These components work together to create a strong system foundation that opens the door for successful deployment and future advancement.

Chapter 5

System Implementation

This chapter describes in detail the methodology used in this project and how it was implemented. It details the working of the prototype developed as part of this project.

5.1 Datasets Identified

The study uses CHB-MIT Scalp EEG Dataset which is a publicly accessible dataset from the Children's Hospital Boston. The dataset includes EEG recordings of 22 paediatric patients suffering from epilepsy. The dataset also provides summary files for each patient which includes details about each file of the patient such as the file start time, end time, number of seizures per file, seizure start time and end time.

Each recording includes roughly an hour of continuous EEG data per file and is supplied in European Data Format (.edf). The recordings have a sampling rate of 256 Hz and adhere to the International 10-20 electrode placement system. The dataset is segmented into preictal as well as ictal periods. This is done by considering the 23 channels as well as the seizure event timestamps that is provided in the summary files discussed earlier. This information hence makes seizure detection using supervised learning easier.

The following preprocessing procedures are applied to the raw EEG signals in order to get the dataset ready for deep learning:

1. **Conversion of the available file format**: To make it easier to handle and interpret the signals the `.edf` files are converted to `.csv` format.

2. **Matrix Representation**: The EEG data in `.csv` format are structured so that:

    - **Columns:** represent the 23 EEG channels.
    - **Rows:** correspond to the time samples in sampling rate of 256Hz.

3. **Partitioning**:

- **Preictal stage:**: The 30 seconds leading up to the start of a seizure, represented as 30 frames (1 second per frame).
- **Ictal stage:**: A window of 10 seconds following the start of a seizure, represented as 10 frames (1 second per frame).

4. **Annotation Integration**: For the purpose of supervised learning, labels are assigned to the EEG datapoints as below:
   - **1:** for seizure events.
   - **0:** for non-seizure events.

5.2 Proposed Methodology

The pipeline of the methodology proposed includes data preprocessing, feature extraction, CNN-based seizure localisation, and explainability techniques. This ensures that the raw EEG signals are converted to segments from which features can be interpreted. SHAP and LIME are then used to obtain meaningful explanations about the model's decision making process.

5.2.1 Data Preprocessing

Preprocessing of Data

- Converting Files for Compatibility
  From the raw EEG data that is present in edf format, csv files are generated. Python library pyedflib is used for this conversion. It is done since feature extraction and analysis is easier in csv format. Csv files are also better suited for deep learning frameworks and for XAI techniques.

- Forming a Matrix Structure
  After conversion, the EEG data is shown as a two-dimensional matrix in order to satisfy convolutional neural networks' (CNNs') input specifications:
  Each EEG recording second equates to 256 time samples, which are represented by rows and recorded at a constant sampling frequency of 256 Hz. The number of

electrodes used during the recording session affects the EEG channels, which are represented by columns. The model can learn spatial links between channels (across columns) as well as temporal patterns (across rows) from the matrix arrangement.

- Cleaning Signals
  The raw EEG data is treated to improve signal quality before analysis by eliminating noise and artifacts, including flatlined or empty channels.

5.2.2 Feature Extraction

EEG signals are divided into brief, meaningful frames and given binary annotations in order to get the data ready for supervised learning:

Preictal Stage Segmentation: The 30 seconds preceding the start of each seizure are taken out. These are divided into thirty one-second, non-overlapping frames, each of which captures minute neurological alterations before to the seizure.

Ictal Stage Segmentation: Ten seconds of EEG data are taken from the beginning of the seizure and split into ten frames of one second each, which correspond to active seizure states.

By using this method, a dataset of preictal and ictal segments is produced, which aids the model in learning patterns unique to seizure transitions.

Analysis of Power Spectral Density (PSD):

- Power spectral density values for alpha and delta waves are generated using 30 one-second preictal frames and one-second ictal frames.

- Alpha Slowing Phenomenon: By taking interframe dependencies into account, it is possible to pinpoint the origins of seizures by observing a noteworthy pattern in which alpha wave values fall while delta wave values rise during seizure onset.

By emphasising noteworthy seizure-related patterns, these extracted features improve classification accuracy and are crucial inputs for the CNN model.

The following equations were used for Feature Extraction:

- **Power Spectral Density (PSD) for EEG Analysis** The Power Spectral Density (PSD) is used to estimate the power of different EEG frequency bands such as delta, theta, alpha, beta and gamma. The Welch method is commonly used in the case of EEG signals:

$$\mathrm{PSD}(f) = \frac{1}{N} \left| \sum_{n=0}^{N-1} x(n) e^{-j2\pi fn/N} \right|^2 \tag{5.1}$$

  where:

  - $x(n)$ is the EEG signal,
  - N is the number of samples in a window,
  - f is the frequency component.

5.2.3 Deep Learning Model: CNN for Seizure Detection

A 2D CNN Architecture is used on the input data to identify the seizure events. The model can capture dependencies in both spatial as well as temporal domain in the EEG signals. The key components of the model are:

- **Architecture**: A A multi-layer 2D CNN is designed for processing EEG signals which is inputted in csv format.
- **Training Phase**: The model is then trained on the annotated EEG data to classify datapoints as seizure and non-seizure events.
- **Hyperparameter Tuning**: Accuracy can be enhanced by optimising parameters such as the number of layers, learning rate, and batch size.
- **Regularization**: Dropout layers are incorporated to improve generalization and prevent overfitting.
- **Output**: The model classifies seizure (1) and non-seizure (0) events using a sigmoid function and generates a probability score for seizure detection.

5.2.4 Seizure time localisation

This module predicts seizure time intervals with great precision by using an auto-calibration technique. The method is used to dynamically determine appropriate thresholds for seizure detection based on preictal relative delta values. The methodology combines statistical methods like Median Absolute Deviation (MAD) and Gaussian Mixture Models (GMM). It also uses interquartile range (IQR), skewness, and kurtosis which act as insights to enable patient-specific threshold determination.

1. Utilization of Input Features
The main input to the time prediction algorithm is the 30 one second frames of preictal data with their delta band power calculated. This data is then flattened into a 1D array from a 2D matrix structure where each row was representing a time sample and each column was representing a feature band. Delta band was used for this method due to its high correlation with preictal and ictal brain activities.

2. Threshold Estimation Based on MAD
To minimize dependency on the outliers and capture variability in the preictal signal, a threshold based on Median Absolute Deviation (MAD) is computed. The threshold is defined as:

$$\text{MAD Threshold} = \text{median} + (\text{scale factor} \times \text{MAD}) \tag{5.2}$$

where MAD is defined as,

$$\text{MAD} = \text{median}(|X_i - \text{median}(X)|) \tag{5.3}$$

where:

- X_i are the EEG signal values,
- $\text{median}(X)$ is the median of the dataset.

This guarantees that the threshold is responsive to increased signal amplitudes linked to seizure onset while staying constant among patients.

3. Threshold Estimation Based on GMM

A Gaussian Mixture Model(GMM) is used to model the preictal data distribution is modeled which consists of several Gaussian components.The model fits upto two to three components in the data depending on the size of the data that is used. After sorting the components according to their means, it then chooses the component which has the greatest mean value . The threshold for the chosen component is then calculated as follows:

$$p(X) = \sum_{i=1}^{K} w_i \mathcal{N}(X \mid \mu_i, \sigma_i^2) \tag{5.4}$$

where:

- w_i is the weight of the i-th Gaussian component (such that $\sum_{i=1}^{K} w_i = 1$),
- $\mathcal{N}(X \mid \mu_i, \sigma_i^2)$ is the probability density function of a normal distribution with mean μ_i and variance σ_i^2,
- K is the total number of Gaussian components in the mixture.

This method captures multimodal and skewed behavior that is frequently seen in preictal dynamics.

4. Distribution based correction

Important distributional properties are retrieved in order to guarantee that the threshold is context-aware:

Data dispersion is measured by the Interquartile Range (IQR), which is used to modify the threshold's sensitivity.

$$\text{IQR} = Q_3 - Q_1 \tag{5.5}$$

where:

- Q_1 is the 25th percentile,
- Q_3 is the 75th percentile.

Skewness: A measure of signal distribution asymmetry that affects the threshold's appropriate shift.

$$\text{Skewness} = \frac{\frac{1}{N}\sum_{i=1}^{N}(X_i - \mu)^3}{\left(\frac{1}{N}\sum_{i=1}^{N}(X_i - \mu)^2\right)^{3/2}} \tag{5.6}$$

where:

- X_i is the EEG value,
- μ is the mean,
- N is the number of samples.

The base threshold is adaptively tuned by adding an auto-correction factor that is calculated using these features:

$$\text{Correction} = 0.05 \cdot \log(1 + \text{IQR}) \cdot \tanh(\text{Skewness}) \tag{5.7}$$

5. Auto Calibrated Threshold learning
Using a constrained optimization technique to learn an ideally adjusted weighting factor, an ensemble approach combines both MAD and GMM thresholds. The function of the objective:
- Maximizes the ictal data's true positive rate (TPR) and minimizes the preictal data's false positive rate (FPR).
- Penalizes criteria that substantially depart from the preictal distribution's 95th percentile.

$$\text{Final Threshold} = \alpha \cdot T_{\text{GMM}} + (1 - \alpha) \cdot T_{\text{MAD}} + \text{Correction} \tag{5.8}$$

where,

- α is the weight parameter balancing GMM and MAD thresholds,
- T_{GMM} is the threshold computed from the Gaussian Mixture Model,
- T_{MAD} is the threshold computed using the Median Absolute Deviation,
- Correction is an adjustment factor based on skewness and IQR.

5.2.5 Explainability Using SHAP and LIME

Two explainable AI strategies are combined to ensure transparency and interpretability:

1. **SHAP (Shapley Additive exPlanations)**:
   - Makes model predictions interpretable both locally and globally.
   - Determines which EEG characteristics are most important in predicting seizures by computing Shapley values.
   - SHAP Value Calculation for CNN Predictions: SHAP values are used for model interpretation and assessing the contribution of each feature in predictions:

   $$\phi_i = \sum_{S \subseteq N \setminus \{i\}} \frac{|S|!(|N| - |S| - 1)!}{|N|!} [f(S \cup \{i\}) - f(S)]$$

   where ϕ_i is the SHAP value for feature i, S is a subset of features, and $f(S)$ is the model prediction with subset S.

2. **Local Interpretable Model-Agnostic Explanations (LIME)**:
   - Alters the EEG data input and tracks variations in the model's predictions.
   - Helps provide an explanation for the reasoning behind a particular prediction made for each data point.
   - The LIME equation is given by:

   $$\xi(x) = \arg\min_{g \in G} L(f, g, \pi_x) + \Omega(g)$$

   where $\xi(x)$ represents the LIME explanation for the instance x. The term $L(f, g, \pi_x)$ is the loss function (or fidelity function), which measures how well the complex model f can be approximated by the interpretable model g within the locality defined by π_x. The function $\Omega(g)$ acts as a complexity term, ensuring that the interpretable model remains simple and understandable.

5.2.6 Model Evaluation and Result Interpretation

Several important metrics are used to evaluate the CNN model's performance:

The model's performance is assessed using accuracy, which measures the overall percentage of correctly classified data points in the EEG recordings. A higher accuracy indicates that the model effectively distinguishes between seizure and non-seizure states. This metric provides a straightforward evaluation of how well the thresholding approach optimizes seizure detection, ensuring that the classification aligns closely with actual seizure events in the data.

Visualization

- SHAP and LIME plots to illustrate the model's reasoning process.
- Performance charts (training/validation loss curves) to track model effectiveness.

5.2.7 User Interaction and Clinical Integration

The key components in the interface that was developed includes:

- Neurologists can upload EEG data and obtain seizure localization results through the user interface.
- Predicts the seizure time intervals to enhance the clinical judgment.
- Explainable AI results to increase trust in decisions made by the model for adoption into medical practice.

This approach combines explainability and deep learning to support medical professionals in managing epilepsy by ensuring precise, comprehensible, and clinically relevant seizure detection.

5.3 Description of Implementation Strategies

5.3.1 Libraries Used

- **TensorFlow/Keras:** For CNN model building.
- **SHAP,LIME:** For visualization of model performance.
- **Matplotlib and Seaborn:** For visualization of model performance.
- **Scikit-learn:** For model evaluation.
- **IDE:** Visual Studio Code, Jupyter Notebook, or PyCharm for development.

5.3.2 Feature Extraction

As part of the feature extraction and analysis process, power spectral density of alpha and delta bands are calculated from the EEG signals. A Fast Fourier Transform(FFT) is used to break down the input signal into its component frequencies. This makes it possible to identify patterns that are associated with seizure events. This method highlighted an important event known as the alpha slowing phenomenon, where during the period before the onset of a seizure, the relative alpha power is seen to be high which then gets converted to delta waves as the seizure occurs. These extracted features improve the CNN model's ability to precisely localise seizure onset and progression by examining inter-frame dependencies.

```python
# Bandpass Filter
def bandpass_filter(data, lowcut, highcut, fs, order=4):
    nyquist = 0.5 * fs
    low, high = lowcut / nyquist, highcut / nyquist
    b, a = butter(order, [low, high], btype='band')
    return lfilter(b, a, data)

# Calculate Power using Welch’s Method
def calculate_power(data, fs=256):
    freqs, psd = welch(data, fs, nperseg=1024)
    return freqs, psd
```

Figure 5.1: Bandpass Filter Implementation and Power Spectral Density (PSD) Calculation (Welch's Method)

5.3.3 Model training

The annotated EEG data is fed into the 2D CNN model to train the model for classifying into seizure and non seizure events. The network uses several convolutional layers, pooling layers, and fully connected layers to identify temporal and spatial patterns in EEG signals. To increase accuracy and avoid overfitting, hyperparameters like learning rate, batch size, and dropout are optimised. In order to classify seizure and non-seizure events using probability scores produced by the sigmoid activation function, the model is trained on annotated data.

```python
# Build the CNN model
model = Sequential([
    Input(shape=(X_shape_1, 1, 1)),
    Conv2D(32, (3, 1), activation='relu'),
    BatchNormalization(),
    MaxPooling2D(pool_size=(2, 1)),
    Dropout(0.3),
    Conv2D(64, (3, 1), activation='relu'),
    BatchNormalization(),
    MaxPooling2D(pool_size=(2, 1)),
    Dropout(0.3),
    Conv2D(128, (3, 1), activation='relu'),
    BatchNormalization(),
    MaxPooling2D(pool_size=(2, 1)),
    Dropout(0.4),
    Flatten(),
    Dense(128, activation='relu'),
    Dropout(0.4),
    Dense(1, activation='sigmoid')
])
```

Figure 5.2: CNN Model

```
# Compile the model
model.compile(optimizer='adam', loss='binary_crossentropy', metrics=['accuracy'])

# CSV Logger
csv_logger = CSVLogger(os.path.join(model_path, 'training_log.csv'))

# Define the checkpoint callback to continue saving the model
checkpoint_callback = ModelCheckpoint(
    filepath=checkpoint_path,
    save_weights_only=False,
    monitor='val_loss',
    mode='min',
    save_best_only=False,
    verbose=1
)

# Train the model
history = model.fit(
    train_dataset,
    epochs=50,
    initial_epoch=0,
    validation_data=val_dataset,
    callbacks=[csv_logger, checkpoint_callback]
)
```

Figure 5.3: Training Configuration, Callbacks and Checkpoints, and Training Execution

5.3.4 Automation of Threshold Fixing

The main issue with seizure time localisation is that the thresholds where the localisation happens is pateint dependent. Hence to automate this we require to follow a process that includes feature extraction as well as statistical methods:

To obtain the initial point from where the distinction of ictal and preictal stages can be made, we use a sliding window method. A sliding window of size 30 is moved with a stride of 1 over the csv file. At each point it calculates the ratio of delta to the alpha value. A general trend of the maximum signal peak occuring in the seizure time range was observed. Based on this, the maximum peak index is first calculated along with the delta-to-alpha ratio at that point. From that point, we now move to preceeding points to find the point where the ratio of delta-to-alpha ratio is greater than or equal to 1.9. This is considered as the temporary seizure onset time.

```python
# Sliding Window
window_size = 30 * 256  # 30 seconds x 256 samples/sec
step_size = 1 * 256  # 1 second step

# Process Each Window
alpha_band = (8, 13)
delta_band = (0.5, 4)

dar_values, time_points = [], []

# Main Window Processing
for start in range(0, len(data) - window_size, step_size):
    end = start + window_size
    window_data = data.iloc[start:end]

    alpha_power_total, delta_power_total = 0, 0

    for col in electrode_cols:
        signal = window_data[col].values
        freqs, psd = calculate_power(signal)

        # Alpha Power
        alpha_idx = np.where((freqs >= alpha_band[0]) & (freqs <= alpha_band[1]))[0]
        alpha_power = np.sum(psd[alpha_idx])

        # Delta Power
        delta_idx = np.where((freqs >= delta_band[0]) & (freqs <= delta_band[1]))[0]
        delta_power = np.sum(psd[delta_idx])

        alpha_power_total += alpha_power
        delta_power_total += delta_power

    # Calculate Delta-to-Alpha Ratio (DAR)
    dar = delta_power_total / alpha_power_total if alpha_power_total != 0 else 0
    dar_values.append(dar)
```

Figure 5.4: Sliding Window Processing and DAR Calculation

```python
# Find the peak (maximum value in the dar_values)
peak_idx = np.argmax(dar_values)  # Find the index of the maximum value
peak_value = dar_values[peak_idx]

# Now check the points behind the peak, calculate the ratio between the peak and each previous point
for j in range(peak_idx - 1, -1, -1):  # Start from the point before the peak and move backward
    ratio = peak_value / dar_values[j] if dar_values[j] != 0 else np.inf  # Avoid division by zero
    if ratio >= 1.9:  # Check if the ratio is 1.9 or greater
        seizure_time = time_points[j]  # Mark the time of seizure onset
        break  # Stop further backtracking as we've found the point of interest
```

Figure 5.5: Peak Detection and Seizure Time Estimation

Based on the possible seizure onset time we have calculated using the sliding window method, 10 one second frames are formed as the ictal time frame and 30 one second frames are formed as the preictal time frames. This helps in studying about the possible patterns before and during the occurence of the seizure that can be correlated with a possible threshold.

```
# Loop to create 30 frames of 1 second each
for frame_index in range(30):
    frame_start_time = start_time - (30 - frame_index)
    frame_end_time = frame_start_time + 1

    # Filter data for the current frame
    frame_data = df[(df['time'] >= frame_start_time) & (df['time'] < frame_end_time)]
```

Figure 5.6: Preictal (30s before seizure onset)

```
# Loop to create 10 frames of 1 second each
for frame_index in range(10):
    frame_start_time = start_time + frame_index
    frame_end_time = frame_start_time + 1

    # Filter data for the current frame
    frame_data = df[(df['time'] >= frame_start_time) & (df['time'] < frame_end_time)]
```

Figure 5.7: Ictal (10s after seizure onset)

For each of the ictal and preictal frame, the relative alpha and relative delta values are now calculated.

```
# Function to calculate relative band power
def calculate_relative_band_power(data, column, band_range, fs=256):
    signal = data[column].values
    freqs, psd = welch(signal, fs=fs)
    band_power = psd[(freqs >= band_range[0]) & (freqs <= band_range[1])].sum()
    total_power = psd.sum()
    return band_power / total_power if total_power > 0 else None
```

Figure 5.8: Relative Band Power Calculation

We use statistical methods like Gaussian Mixture Model (GMM) and Median Absolute Deviation (MAD) to compute dynamic thresholds for detecting significant events in EEG signals. This is done on the preictal delta values which is taken as the input. GMM models the signal distribution in the input data using multiple Gaussian components and defines a threshold based on the mean and standard deviation of the component that is dominant. MAD on the other hand uses the absolute deviation of each datapoint from the median value thereby making it less dependent on the values of the outliers. On combining these two features along with a correction factor, we obtain a more adaptive and reliable threshold that can work well across varying signal characteristics.

```python
def dynamic_mad_threshold(data):
    """MAD-based threshold with adaptive scaling"""
    data_clean = data[~np.isnan(data)]
    if len(data_clean) == 0:
        return np.nan
    median = np.median(data_clean)
    mad_val = mad(data_clean)
    if mad_val == 0:
        return median
    # Adaptive scaling based on data spread
    spread_factor = np.log1p(np.ptp(data_clean))  # Log of range
    scale_factor = 1.0 + 0.3 * spread_factor
    return median + scale_factor * mad_val
```

Figure 5.9: MAD-Based Adaptive Threshold

```python
def refined_gmm_threshold(data, components=2):
    """Improved GMM threshold with robust component selection"""
    data_clean = data[~np.isnan(data)]
    if len(data_clean) < 10:
        return np.percentile(data_clean, 90)

    try:
        gmm = GaussianMixture(n_components=min(components, len(data_clean)//3),
                              random_state=42, n_init=3)
        gmm.fit(data_clean.reshape(-1, 1))
        means = gmm.means_.flatten()
        stds = np.sqrt(gmm.covariances_.flatten())
        weights = gmm.weights_.flatten()

        # Sort components by mean value
        sorted_idx = np.argsort(means)
        means = means[sorted_idx]
        stds = stds[sorted_idx]
        weights = weights[sorted_idx]

        # Find optimal component (balancing weight and position)
        optimal_idx = -1  # Start with highest mean component
        for i in reversed(range(len(means))):
            if weights[i] > 0.2:  # Only consider significant components
                optimal_idx = i
                break

        # Conservative estimate: mean + 0.25*std
        return means[optimal_idx] + 0.25*stds[optimal_idx]
    except:
        return np.percentile(data_clean, 85)
```

Figure 5.10: GMM Threshold

```python
def calculate_distribution_features(data):
    """Calculate key statistical features of the data"""
    data_clean = data[~np.isnan(data)]
    if len(data_clean) < 2:
        return {'skew': 0, 'kurtosis': 3, 'iqr': 0, 'tail_ratio': 1}

    q75, q25 = np.percentile(data_clean, [75, 25])
    iqr_val = q75 - q25
    median = np.median(data_clean)

    return {
        'skew': skew(data_clean),
        'kurtosis': max(kurtosis(data_clean), 1),  # Minimum kurtosis of 1
        'iqr': iqr_val,
        'tail_ratio': (q75 - median)/(median - q25 + 1e-9)  # Avoid division by zero
    }
```

Figure 5.11: Distribution Feature Extraction

```python
def auto_calibrated_ensemble(preictal_values, ictal_values=None):
    """Optimized ensemble method that automatically learns the best threshold"""
    # Calculate base thresholds
    gmm_t = refined_gmm_threshold(preictal_values)
    mad_t = dynamic_mad_threshold(preictal_values)

    # Calculate distribution features
    features = calculate_distribution_features(preictal_values)

    # Optimize the ensemble weighting
    def objective(alpha):
        threshold = alpha*gmm_t + (1-alpha)*mad_t

        # Distribution-based penalty
        p95 = np.percentile(preictal_values[~np.isnan(preictal_values)], 95)
        dist_penalty = abs(p95 - threshold)/np.ptp(preictal_values[~np.isnan(preictal_values)])

        # Performance-based optimization (if ictal data available)
        if ictal_values is not None:
            metrics = evaluate_performance(preictal_values, ictal_values, threshold)
            perf_score = 0.7*metrics['TP Rate'] + 0.3*(1-metrics['FP Rate'])
        else:
            perf_score = 0

        return -(0.6*perf_score + 0.4*(1-dist_penalty))
```

Figure 5.12: Ensemble Threshold with Auto-Calibration

```
# Find optimal alpha between 0 and 1
res = minimize_scalar(objective, bounds=(0, 1), method='bounded')
optimal_alpha = res.x

# Calculate final threshold with learned weighting
base_threshold = optimal_alpha*gmm_t + (1-optimal_alpha)*mad_t

# Apply automatic correction based on historical patterns
auto_correction = 0.05 * np.log1p(features['iqr']) * np.tanh(features['skew'])
final_threshold = base_threshold + auto_correction
```

Figure 5.13: Automatic threshold fixing

Now we obtain the threshold value for the patient. This is then used in the seizure time prediction algorithm which would predict the fine tuned seizure start time as well as end time.

```
# Merge contiguous or close seizure intervals
seizure_frames = df_filtered[df_filtered['predicted_label'] == 1]
merged_seizures = []

if not seizure_frames.empty:
    start_time = seizure_frames['time'].iloc[0]
    for i in range(1, len(seizure_frames)):
        if seizure_frames['time'].iloc[i] - seizure_frames['time'].iloc[i - 1] <= merge_interval_threshold:
            continue
        else:
            end_time = seizure_frames['time'].iloc[i - 1]
            # Only keep intervals longer than minimum seizure duration
            if end_time - start_time >= min_seizure_duration:
                merged_seizures.append([start_time, end_time])
            start_time = seizure_frames['time'].iloc[i]

    # Final interval check for the last sequence
    end_time = seizure_frames['time'].iloc[-1]
    if end_time - start_time >= min_seizure_duration:
        merged_seizures.append([start_time, end_time])

# Save merged seizure intervals as a new CSV for each file
merged_seizures_df = pd.DataFrame(merged_seizures, columns=['Start Time (s)', 'End Time (s)'])

# Save the merged seizure intervals to CSV
csv_file_name = os.path.join(seizure_time_path, file)
merged_seizures_df.to_csv(csv_file_name, index=False)
```

Figure 5.14: Seizure Interval Merging Logic

5.3.5 SHAP:SHapley Additive exPlanations

SHAP works by calculating the importance score for each of the EEG features. It works on the principle of game theory which suggests that the contribution of a feature to the output of a model can be calculated by using shapley values. It thereby enhances the transparency of the model by providing global explanations to the predictions made by the model. Clinicians can understand the decision behind the seizure localisation process and the model's decision-making process by examining these visualisations.

```python
# KernelExplainer works by using the model's predict function and a background dataset
def model_predict(X_batch):
    # Ensure the model gets input in the correct shape (batch size, 23 features, 1, 1)
    X_batch = X_batch.reshape(-1, 23, 1, 1)  # Reshape batch to match model input shape
    return model.predict(X_batch).reshape(-1)  # Return flattened predictions

# Initialize SHAP Kernel Explainer with the model and background data
explainer = shap.KernelExplainer(model_predict, X_background.reshape(X_background.shape[0], -1))
```

Figure 5.15: SHAP Initialization and Model Wrapper

```python
# Process in batches
for i in range(0, num_samples, batch_size):
    batch = X_explain[i:i + batch_size]
    print(f"Processing batch {i // batch_size + 1}: {batch.shape}")
    try:
        shap_values_batch = explainer.shap_values(batch.reshape(batch.shape[0], -1))  # SHAP computation for the batch
        shap_values_full.append(shap_values_batch[0])  # Use first output's SHAP values
    except Exception as e:
        print(f"Error during SHAP value computation for batch {i // batch_size + 1}: {e}")
        raise

# Concatenate all batched SHAP values
shap_values_full = np.concatenate(shap_values_full, axis=0)
```

Figure 5.16: Batch Processing for Memory Efficiency

```python
force_fig = shap.force_plot(
    explainer.expected_value[0],
    shap_values_full[0],  # Flatten SHAP values for visualization
    X_explain[0].reshape(-1),  # Flatten input features for visualization
    feature_names=selected_electrodes
)
```

Figure 5.17: Force Plot

```python
shap.summary_plot(
    shap_values_full,                 # Flatten SHAP values for summary plot
    X_explain.reshape(num_samples, 23),  # Flatten input features for summary plot
    feature_names=selected_electrodes
)
```

Figure 5.18: Summary Plot

5.3.6 LIME:Local Interpretable Model-Agnostic Explanations

In LIME technique, the interpretation for individual seizure predictions are done by perturbing the input EEG data. It produces locally faithful explanations by examining how much the perturbations affect the output of the model. The LIME visualisation highlights the most significant EEG features towards the decision made by the CNN model. Hence it increases the interpretability of the predictions made by the CNN model and thereby increasing the trust of the doctors in the decision made by the model.

```python
# Prediction function
def model_predict(input_data):
    sample = input_data.reshape(-1, X.shape[1], 1, 1)
    prediction = model.predict(sample)
    probability_0 = 1 - prediction
    probability_1 = prediction
    return np.concatenate([probability_0, probability_1], axis=1)
```

Figure 5.19: Model Prediction Wrapper

```python
# Creating LIME explainer
explainer = LimeTabularExplainer(
    training_data=X.reshape(X.shape[0], X.shape[1]),
    mode='classification',
    feature_names=selected_electrodes
)
```

Figure 5.20: LIME Explainer Initialization

```
for i in range(len(X)):
    # Generating explanation
    explanation = explainer.explain_instance(
        data_row=X[i].reshape(-1),
        predict_fn=model_predict,
        top_labels=2,
        num_features=23
    )

    # Explanations
    prediction_class = 1
    local_exp = explanation.local_exp[prediction_class]
    mapped_explanations_d = {explainer.feature_names[idx]: weight for idx, weight in local_exp}
```

Figure 5.21: Per-Sample Explanation Generation

Chapter 6

Results and Discussions

The outcomes of the seizure localisation method discussed show how well it can distinguish between seizure and non-seizure events in EEG data. The CNN model shows high accuracy of 98% which demonstrates its usability as a seizure localisation tool. The explainability tools such as SHAP and LIME emphasises the most contributing features to the output of the CNN model. This thereby sheds light on the decision making process of the model. The feature extraction and its analysis highlighted the importance of alpha and delta waves in seizure localisation process. The seizure detection method was further fine-tuned using automated threshold fixing methods such as autocalibration technique which employs multiple statistical methods. The overall outcomes confirm how well the suggested methodology works to help clinicians detect seizures in a way that is both accurate and understandable.

Loss Curve

An illustration of a machine learning model's error (or loss) across time, usually over training epochs, is called a loss plot. Typically, it has two curves: one for validation loss and one for training loss.

Interpretation for any loss curve: While the validation loss demonstrates how effectively the model generalizes to new data, the training loss reveals how well the model is learning from the training data. As training goes on, both losses should ideally decline and even off. Good learning and generalization are indicated by a lowering training loss accompanied by a consistently low or decreasing validation loss. However, overfitting may be indicated if the validation loss rises while the training loss falls.

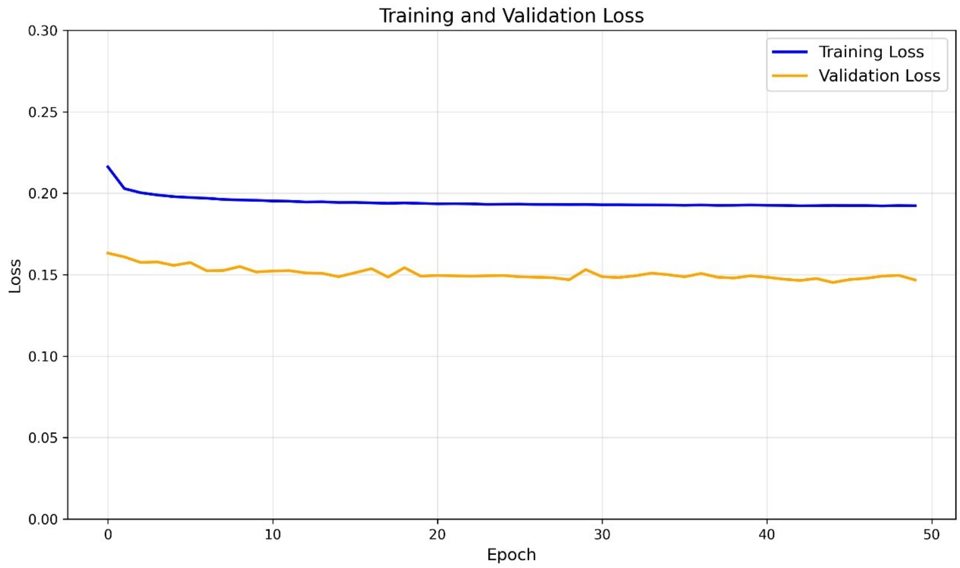


Figure 6.1: Loss plot

Interpretation of the above result: The model's ability to learn from the training data is demonstrated by the plot's training loss, which begins quite high and then gradually plateaus and drops smoothly. It is rare but not necessarily harmful if the validation loss is consistently smaller than the training loss; this could be because of batch normalization, regularization, or dropout strategies that are only used during training. The validation loss shows a general declining trend which suggests that the model can generalise well on unseen data during the training phase. The plot also suggests that the model is devoid of overfitting or underfitting phenomenons.

F1 Score

F1 score is a critical evaluation metric in binary classification tasks. This is of particular importance in datasets where class imbalances can be an issue such as in medical event detections. F1 score is the harmonic mean of precision and recall. It provides insights on the performance of the model and how well the model can balance false positives and false negatives. F1 score serves as a better analysis tool in imbalanced datasets where accuracy can be often deceptive. The model developed had an F1 score of 0.9 which indicates high degrees of consistency as well as efficiency in the detection of seizure events. This suggests that the model guarantees that the seizures predicted are meaningful in addition to the accuracy with which it is predicted. A high F1-score highlights the model's potential for real-world application in clinical settings where prompt and precise seizure detection is essential.

Automated Threshold fixation

The statistical methods such as GMM and MAD were used to calculate the threshold that is required for the seizure onset prediction, the following thresholds were observed:

Table 6.1: Thresholds Estimated Through Statistical Methods for Each Patient

Patient Number	Threshold
Patient 1	0.76
Patient 3	0.73
Patient 5	0.89
Patient 11	0.73
Patient 22	0.78

For patient 5, when the calculated threshold 0.89 was used for seizure time localisation, the following results were observed. The table 6.2 shows the files where seizure was recorded and the actual time of seizure occurence which was denoted in the summary files that were provided with the dataset. The predicted time of seizure occurence is the localisation that was achieved with the threshold that was calculated.

Table 6.2: Comparison of Actual and Predicted Seizure Intervals for Patient CH05

Files	Actual Interval	Predicted Interval
chb05_01.csv	-	-
chb05_06.csv	417 – 532	425 – 531
chb05_13.csv	1086 – 1196	1087 – 1196
chb05_16.csv	2317 – 2413	2321 – 2413
chb05_17.csv	2451 – 2571	2453 – 2469
chb05_22.csv	2348 – 2465	2344 – 2464

For a number of EEG recordings from patient CH05, the table shows a comparison between the actual and expected seizure intervals. The results from GMM and MAD-based estimations are combined in a statistical thresholding procedure to obtain the expected intervals. The method's efficacy in localizing seizure episodes is demonstrated by the strong alignment between the predicted and actual seizure intervals throughout the majority of files. As an illustration of excellent precision, the projected interval (1087–1196) in 'chb05_13.csv' closely resembles the actual interval (1086–1196). Minor differences can be ascribed to noise in the signal or threshold tuning effects. For example, in 'chb05_06.csv', the projected interval begins 8 seconds later than the actual.

Overall, the statistical approach shows good reliability in accurately detecting the onset and offset of seizures with little mistake, indicating that it may help neurologists locate seizures precisely.

SHAP Visualisation

By giving each feature an importance value for a specific prediction, SHAP, a popular post-hoc interpretability strategy, aids in explaining the results of any machine learning model. SHAP offers both local and global explanations and is grounded in game theory. SHAP values, which indicate the relative contributions of each attribute to the final conclusion, provide a local explanation of the prediction for a particular occurrence. To find general trends in feature relevance, SHAP can be combined globally across all instances.

Interpretation of any SHAP plot: Bar plots or force plots are commonly used to visualize SHAP results. The SHAP value, which shows how a feature affects the model output, is indicated on the x-axis of these charts. Red (positive SHAP values) features push the forecast higher (toward seizure, for example), whereas blue (negative SHAP values) features push it lower (toward non-seizure). The strength of the feature's contribution to that particular prediction is shown by the SHAP value's magnitude.

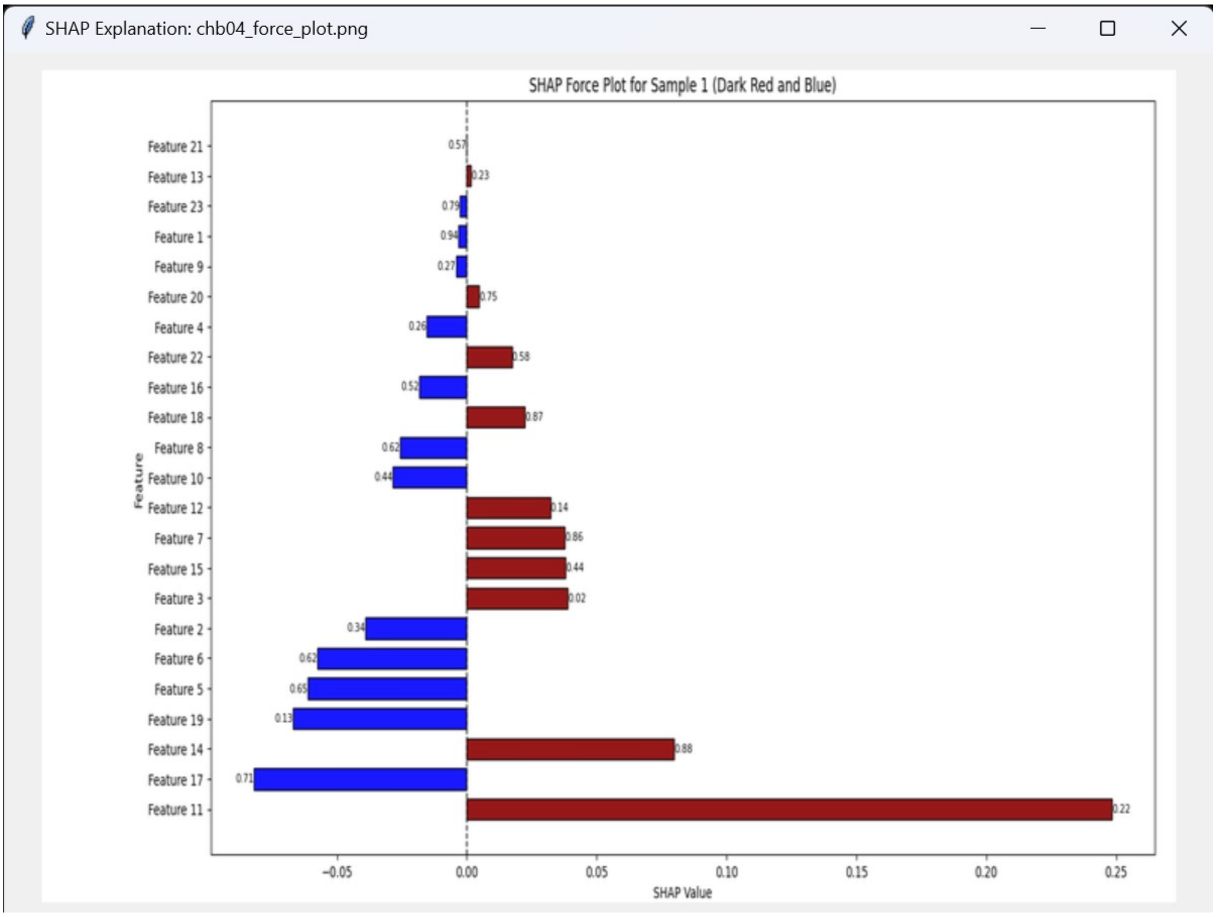


Figure 6.2: SHAP Result

Interpretation of the above SHAP visualisation: The SHAP force map for sample 1 in the provided figure indicates that Feature 11 is a major driver for seizure prediction, contributing the most to raising the prediction value (with a SHAP value of around 0.22). Significant benefits are also provided by other features, such as Features 14, 15, and 7. However, the prediction is pushed away from seizure classification by Features 17 and 5, which have a detrimental impact. To make the final prediction, the model weighs these conflicting factors. By defining the distinct roles of each feature, SHAP offers transparency and facilitates a better understanding of the model's decision-making process. Furthermore, we may derive global insights into which factors regularly affect predictions across the dataset by aggregating SHAP values across numerous such samples.

LIME Visualisation

Any machine learning model's predictions can be explained using the LIME technique, which approximates the model locally using an interpretable model. LIME concentrates on local explanations, or the reasons behind a single instance's particular prediction, as opposed to global explanation techniques, which explain the behavior of the model as a whole. Because of this, LIME is particularly helpful in delicate or high-stakes fields like healthcare, where it is essential to comprehend individual forecasts.

While SHAP provides local interpretability as well, LIME goes one step further by employing a different method: LIME trains a more straightforward, interpretable model (like a decision tree or linear model) that locally replicates the black-box model by perturbing the input data surrounding the sample and observing the resulting predictions. Because of this methodological difference, it is advantageous to use both SHAP and LIME; SHAP gives consistency and universal comprehension, but LIME delivers rule-based interpretations for specific circumstances that are easy to grasp and human-readable.

Interpretation of any LIME plot: Every bar shows a situation (feature range) and how it affects the forecast. The bar's length indicates the amount that condition contributed. Red bars show negative contributions, which push the prediction away from the expected class, and green bars show positive contributions, which push the forecast toward the predicted class. The x-axis displays each feature condition's weight or influence.

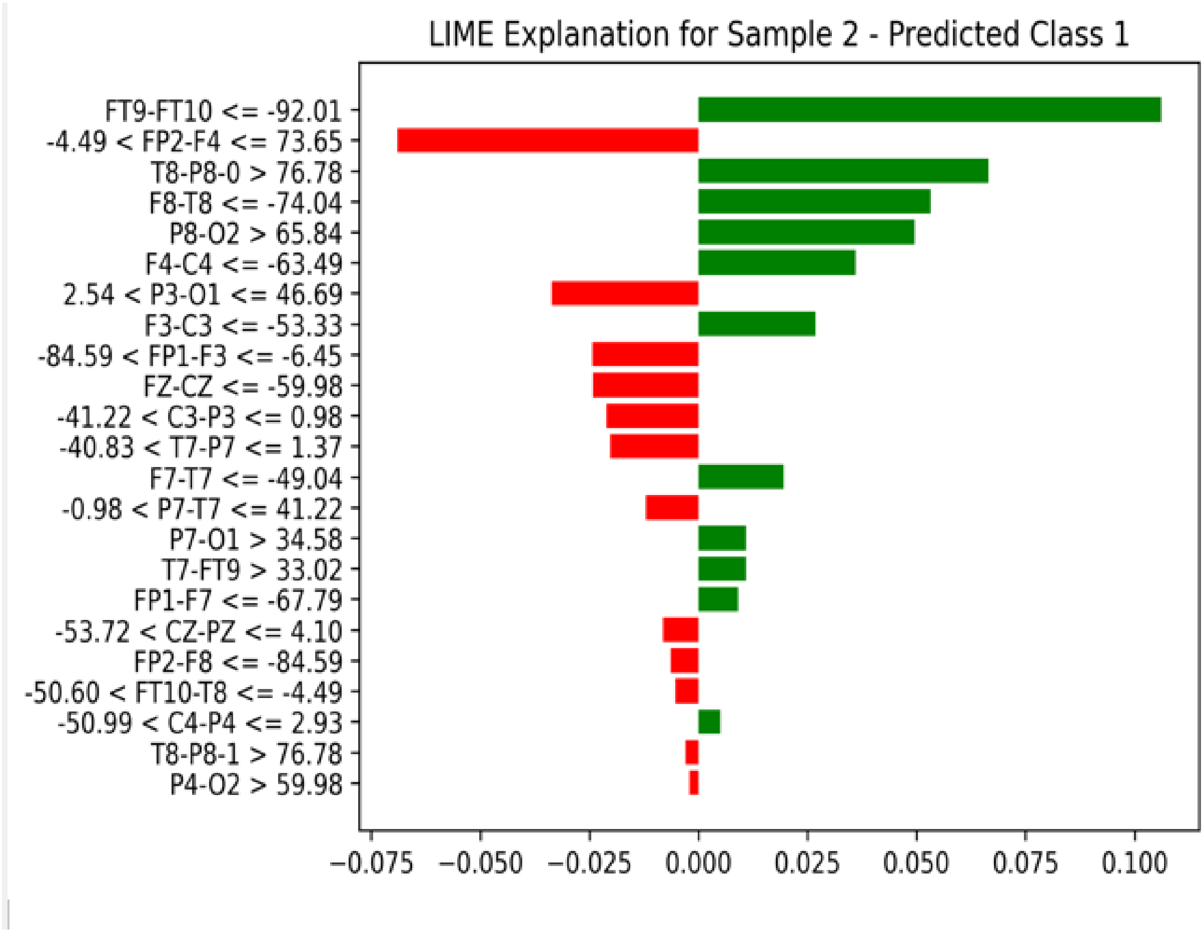


Figure 6.3: LIME Result

Interpretation of the above LIME Result: We are considering the patient 1, whose file chb01_03.csv contains a seizure. The seizure starts at 3001.191406. Lime explanations can be made by perturbing the data in this region. The forecast for Sample 2, which was categorized as Class 1 (seizure), is explained by this LIME plot. With a significant positive contribution (green bar), FT9-FT10 ¡= -92.01 is the most significant criterion, greatly pushing the forecast in the direction of Class 1.

Positive contributions are also made by conditions such as F8-T8 <= -74.04 and T8-P8 > 76.78.

P3-O1 <= 2.54 and FP2-F4 <= 73.65, on the other hand, are adversely contributing (red bars), contradicting the seizure prediction but not sufficiently so to reverse it.

Certain factors have a negligible impact, suggesting that they have little influence over the local choice.

All things considered, the explanation demonstrates that several channel pairs with large potential differences (such as FT9-FT10) were crucial in identifying the data as a seizure, whereas a few others attempted to disprove the prediction but were less successful.

Understanding why a particular EEG recording was marked as a seizure requires this type of interpretability, which is why LIME is a useful supplement to SHAP in attaining both comprehensive and case-specific interpretability.

The GUI Interface for the neurologists:

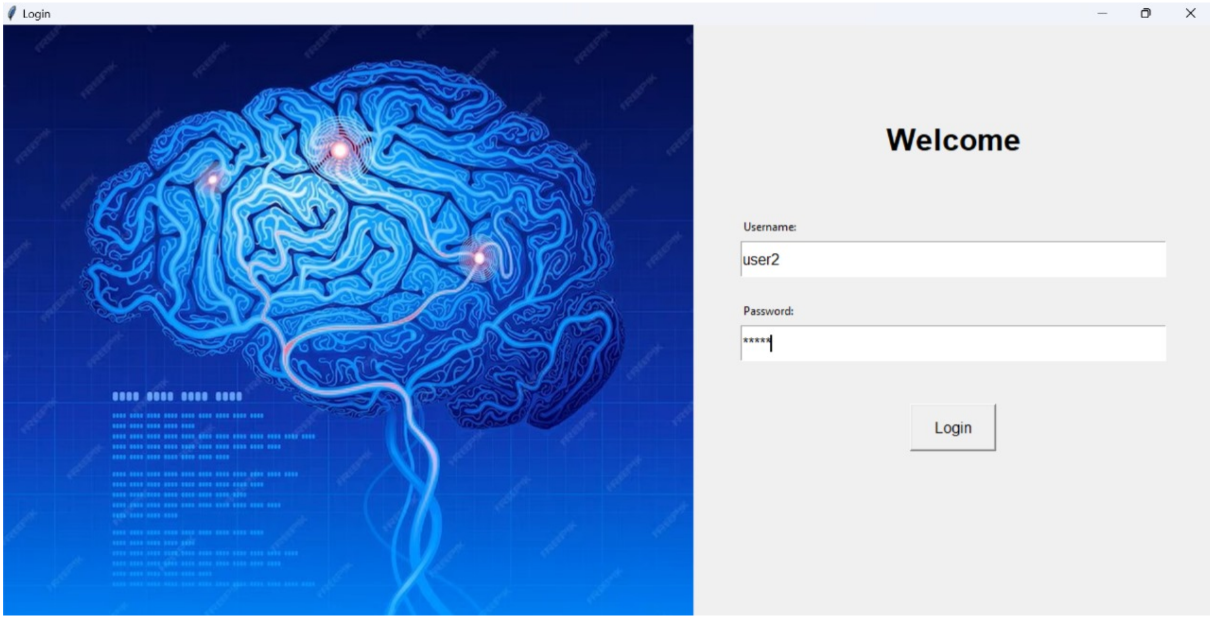


Figure 6.4: Login Page

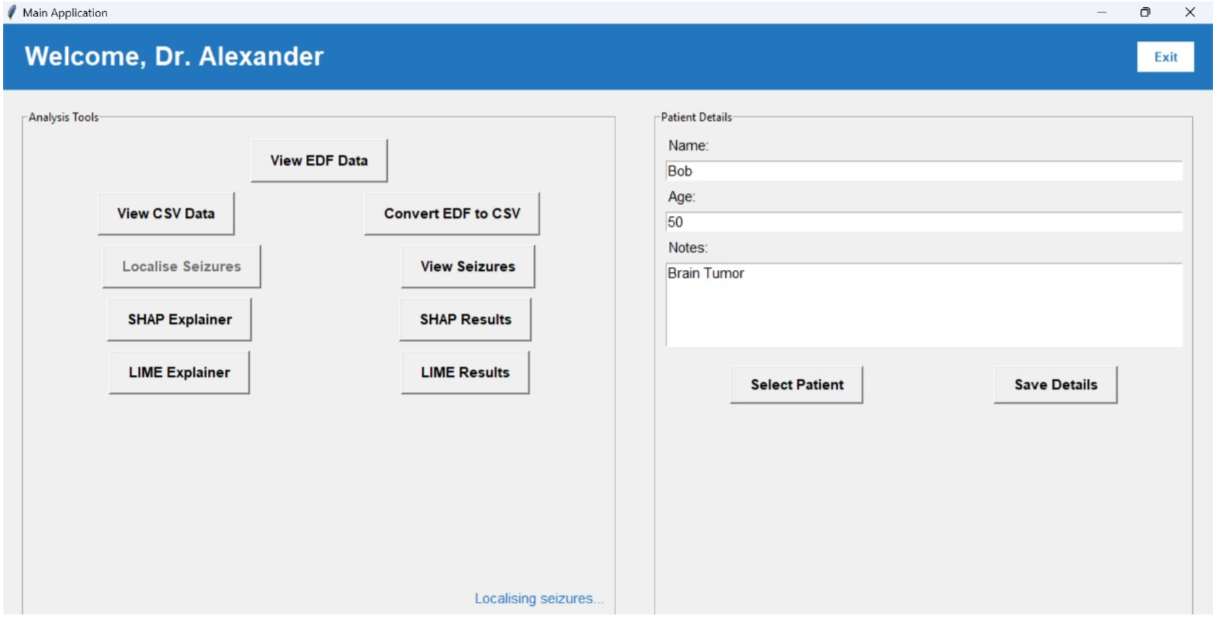


Figure 6.5: Home Page

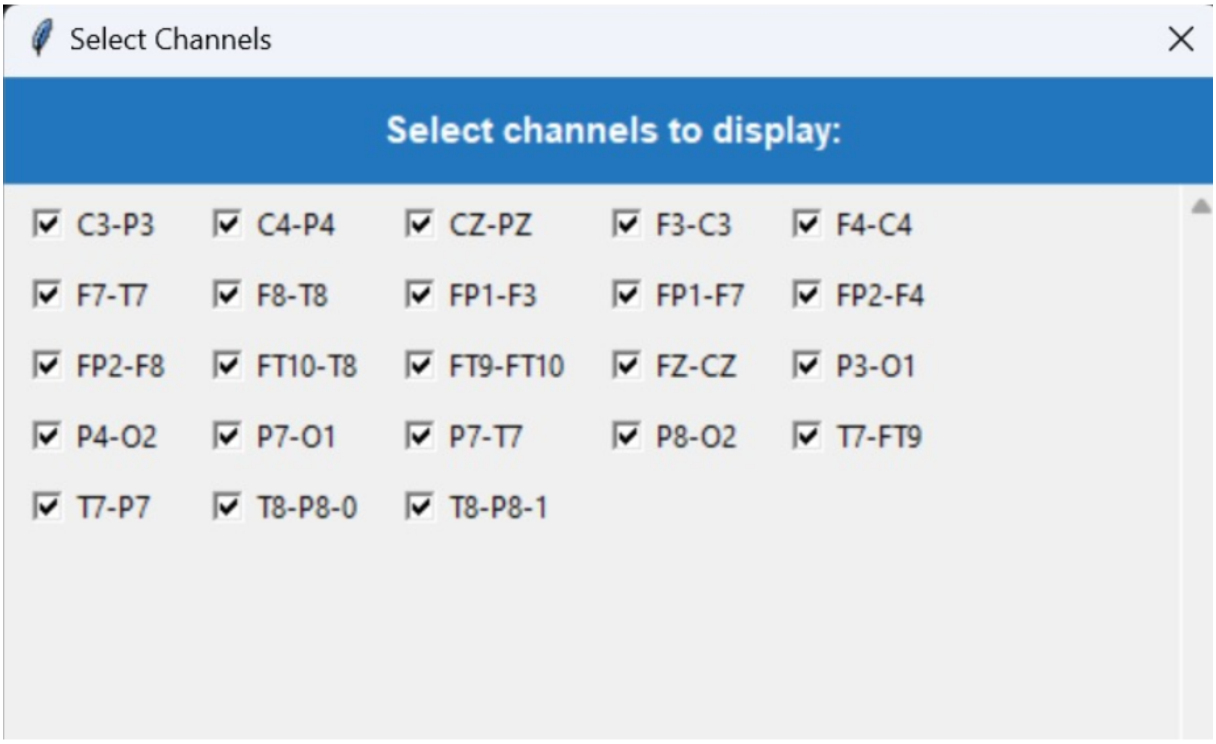


Figure 6.6: View EDF Files Button

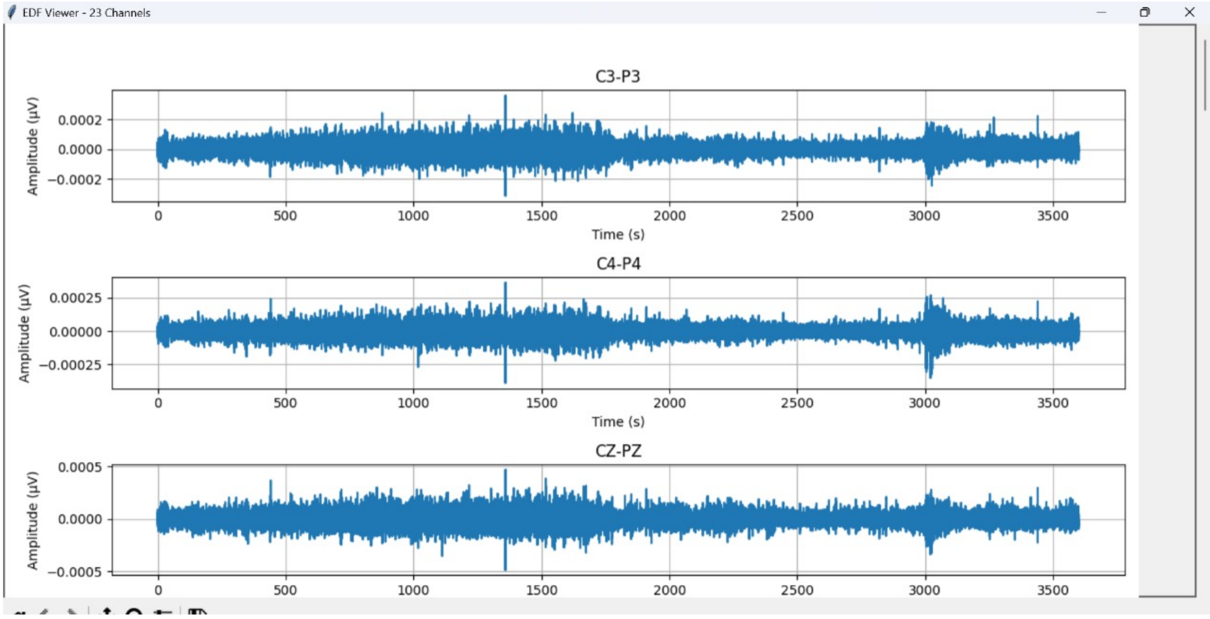
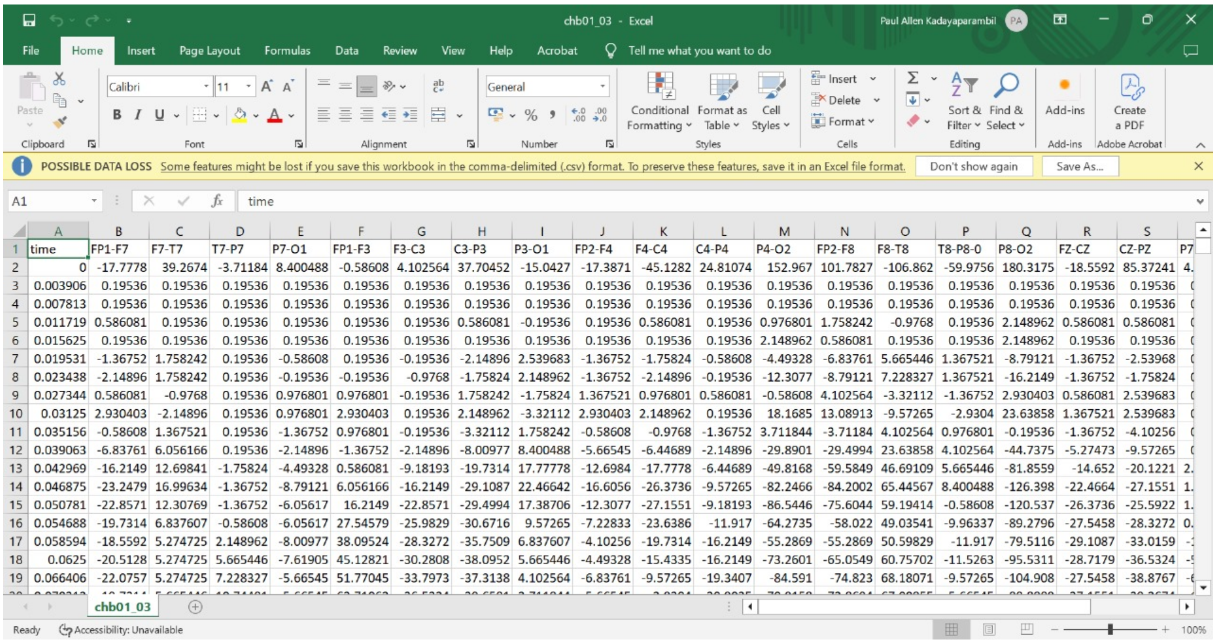


Figure 6.7: EDF Data Visualised for the selected electrodes



time	FP1-F7	F7-T7	T7-P7	P7-O1	FP1-F3	F3-C3	C3-P3	P3-O1	FP2-F4	F4-C4	C4-P4	P4-O2	FP2-F8	F8-T8	T8-P8-0	P8-O2	FZ-CZ	CZ-PZ
0	-17.7778	39.2674	-3.71184	8.400488	-0.58608	4.102564	37.70452	-15.0427	-17.3871	-45.1282	24.81074	152.967	101.7827	-106.862	-59.9756	180.3175	-18.5592	85.37241
0.003906	0.19536	0.19536	0.19536	0.19536	0.19536	0.19536	0.19536	0.19536	0.19536	0.19536	0.19536	0.19536	0.19536	0.19536	0.19536	0.19536	0.19536	0.19536
0.007813	0.19536	0.19536	0.19536	0.19536	0.19536	0.19536	0.19536	0.19536	0.19536	0.19536	0.19536	0.19536	0.19536	0.19536	0.19536	0.19536	0.19536	0.19536
0.011719	0.586081	0.19536	0.19536	0.19536	0.19536	0.19536	0.586081	-0.19536	0.19536	0.586081	0.19536	0.976801	1.758242	-0.9768	0.19536	2.148962	0.586081	0.586081
0.015625	0.19536	0.19536	0.19536	0.19536	0.19536	0.19536	0.19536	0.19536	0.19536	0.19536	0.19536	2.148962	0.586081	0.19536	0.19536	2.148962	0.19536	0.19536
0.019531	-1.36752	1.758242	0.19536	-0.58608	0.19536	-0.19536	-2.14896	2.539683	-1.36752	-1.75824	-0.58608	-4.49328	-6.83761	5.665446	1.367521	-8.79121	-1.36752	-2.53968
0.023438	-2.14896	1.758242	0.19536	-0.19536	-0.19536	-0.9768	-1.75824	2.148962	-1.36752	-2.14896	-0.19536	-12.3077	-8.79121	7.228327	1.367521	-16.2149	-1.36752	-1.75824
0.027344	0.586081	-0.9768	0.19536	0.976801	0.976801	-0.19536	1.758242	-1.75824	1.367521	0.976801	0.586081	-0.58608	4.102564	-3.32112	-1.36752	2.930403	0.586081	2.539683
0.03125	2.930403	-2.14896	0.19536	0.976801	2.930403	0.19536	2.148962	-3.32112	2.930403	2.148962	0.19536	18.1685	13.08913	-9.57265	-2.9304	23.63858	1.367521	2.539683
0.035156	-0.58608	1.367521	0.19536	-1.36752	0.976801	-0.19536	-3.32112	1.758242	-0.58608	-0.9768	-1.36752	3.711844	-3.71184	4.102564	0.976801	-0.19536	-1.36752	-4.10256
0.039063	-6.83761	6.056166	0.19536	-2.14896	-1.36752	-2.14896	-8.00977	8.400488	-5.66545	-6.44689	-2.14896	-29.8901	-29.4994	23.63858	4.102564	-44.7375	-5.27473	-9.57265
0.042969	-16.2149	12.69841	-1.75824	-4.49328	0.586081	-9.18193	-19.7314	17.77778	-12.6984	-17.7778	-6.44689	-49.8168	-59.5849	46.69109	5.665446	-81.8559	-14.652	-20.1221
0.046875	-23.2479	16.99634	-1.36752	-8.79121	6.056166	-16.2149	-29.1087	22.46642	-16.6056	-26.3736	-9.57265	-82.2466	-84.2002	65.44567	8.400488	-126.398	-22.4664	-27.1551
0.050781	-22.8571	12.30769	-1.36752	-6.05617	16.2149	-22.8571	-29.4994	17.38706	-12.3077	-27.1551	-9.18193	-86.5446	-75.6044	59.19414	-0.58608	-120.537	-26.3736	-25.5922
0.054688	-19.7314	6.837607	-0.58608	-6.05617	27.54579	-25.9829	-30.6716	9.57265	-7.22833	-23.6386	-11.917	-64.2735	-58.022	49.03541	-9.96337	-89.2796	-27.5458	-28.3272
0.058594	-18.5592	5.274725	2.148962	-8.00977	38.09524	-28.3272	-35.7509	6.837607	-4.10256	-19.7314	-16.2149	-55.2869	-55.2869	50.59829	-11.917	-79.5116	-29.1087	-33.0159
0.0625	-20.5128	5.274725	5.665446	-7.61905	45.12821	-30.2808	-38.0952	5.665446	-4.49328	-15.4335	-16.2149	-73.2601	-65.0549	60.75702	-11.5263	-95.5311	-28.7179	-36.5324
0.066406	-22.0757	5.274725	7.228327	-5.66545	51.77045	-33.7973	-37.3138	4.102564	-6.83761	-9.57265	-19.3407	-84.591	-74.823	68.18071	-9.57265	-104.908	-27.5458	-38.8767

Figure 6.8: Input CSV

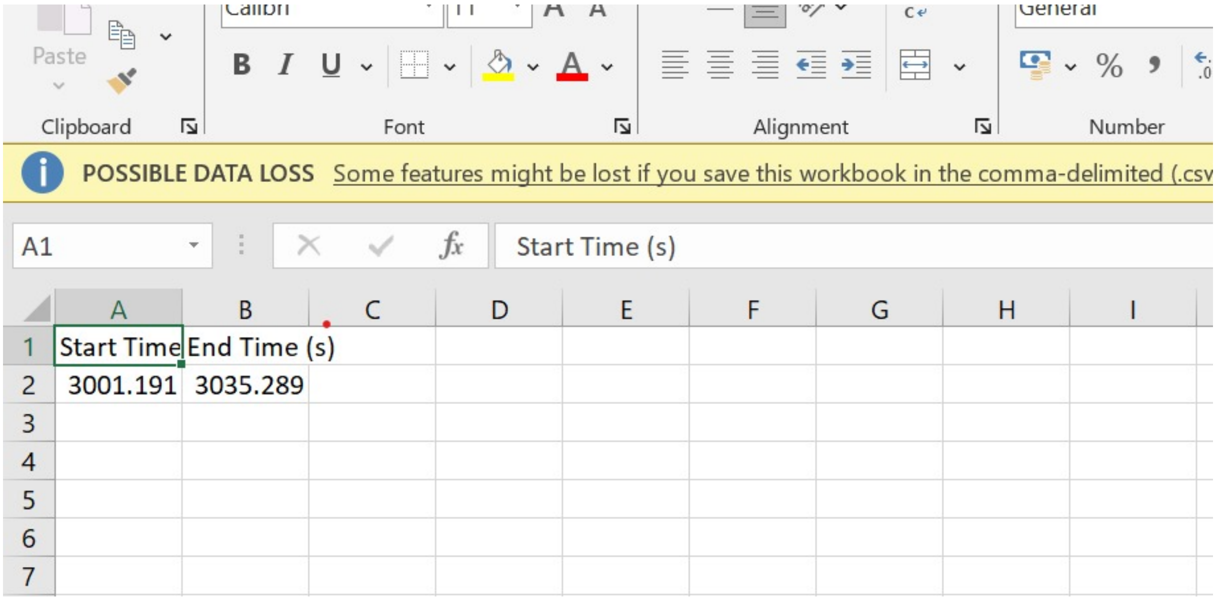


Figure 6.9: Time Prediction in one seizure file of patient ch01

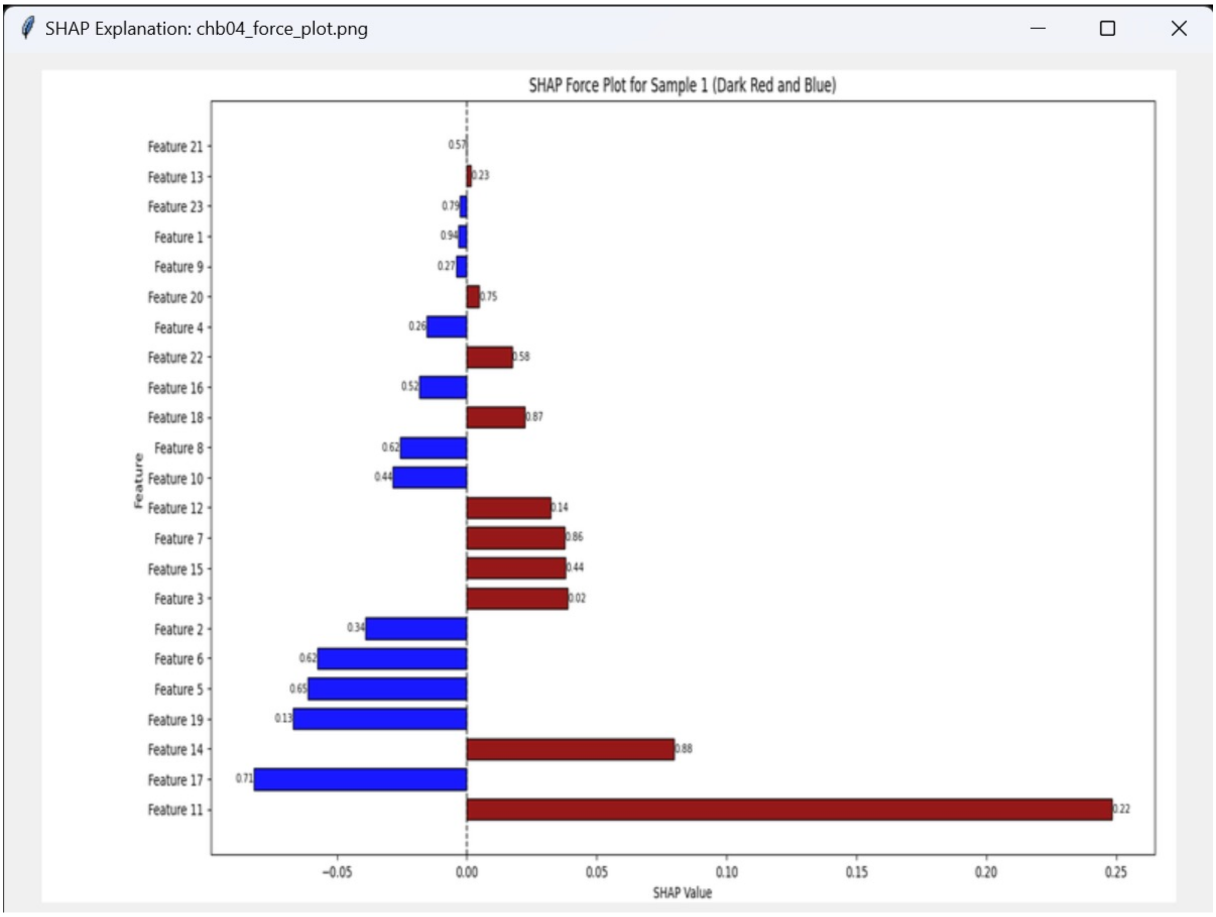


Figure 6.10: SHAP Result for patient ch01's seizure 1

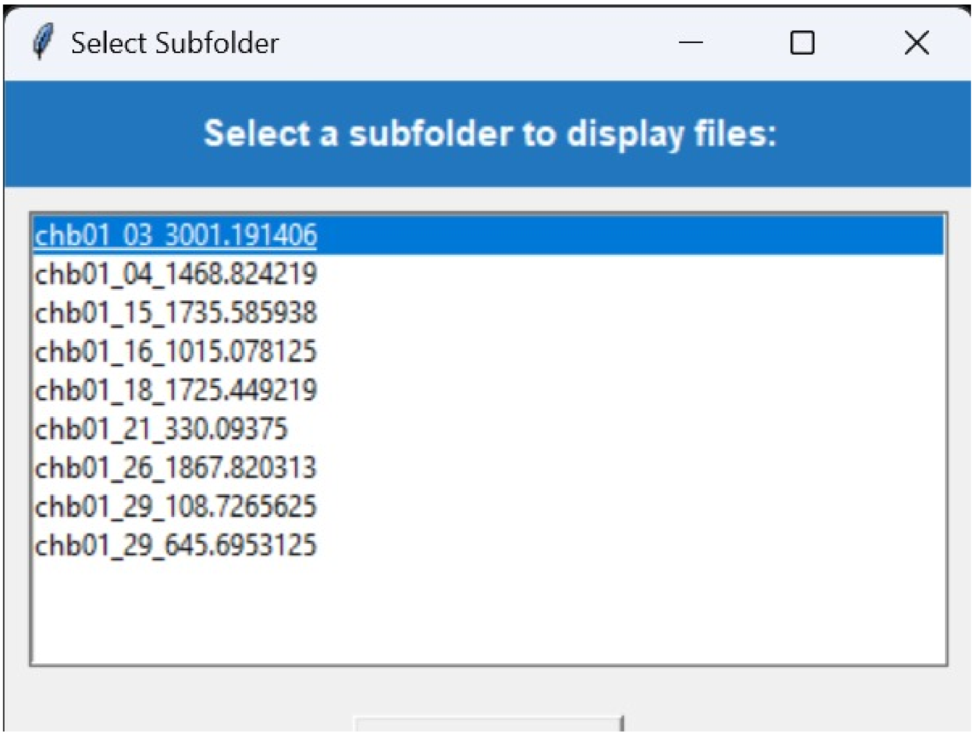


Figure 6.11: Subfolder Selection for lime visualisation

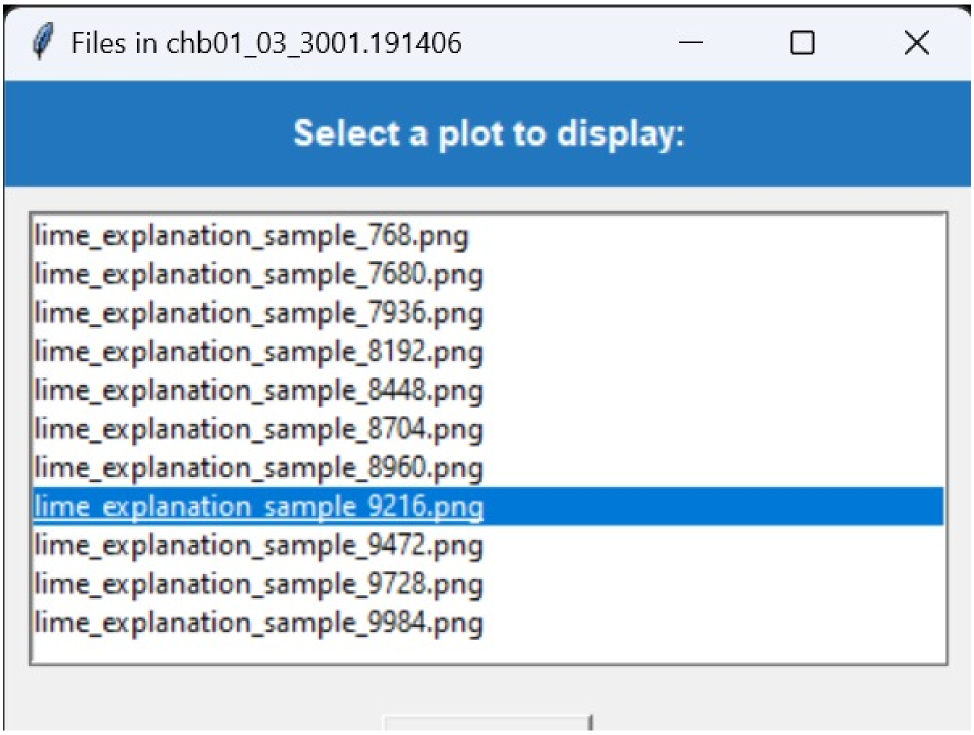


Figure 6.12: Select Plot of lime to visualise

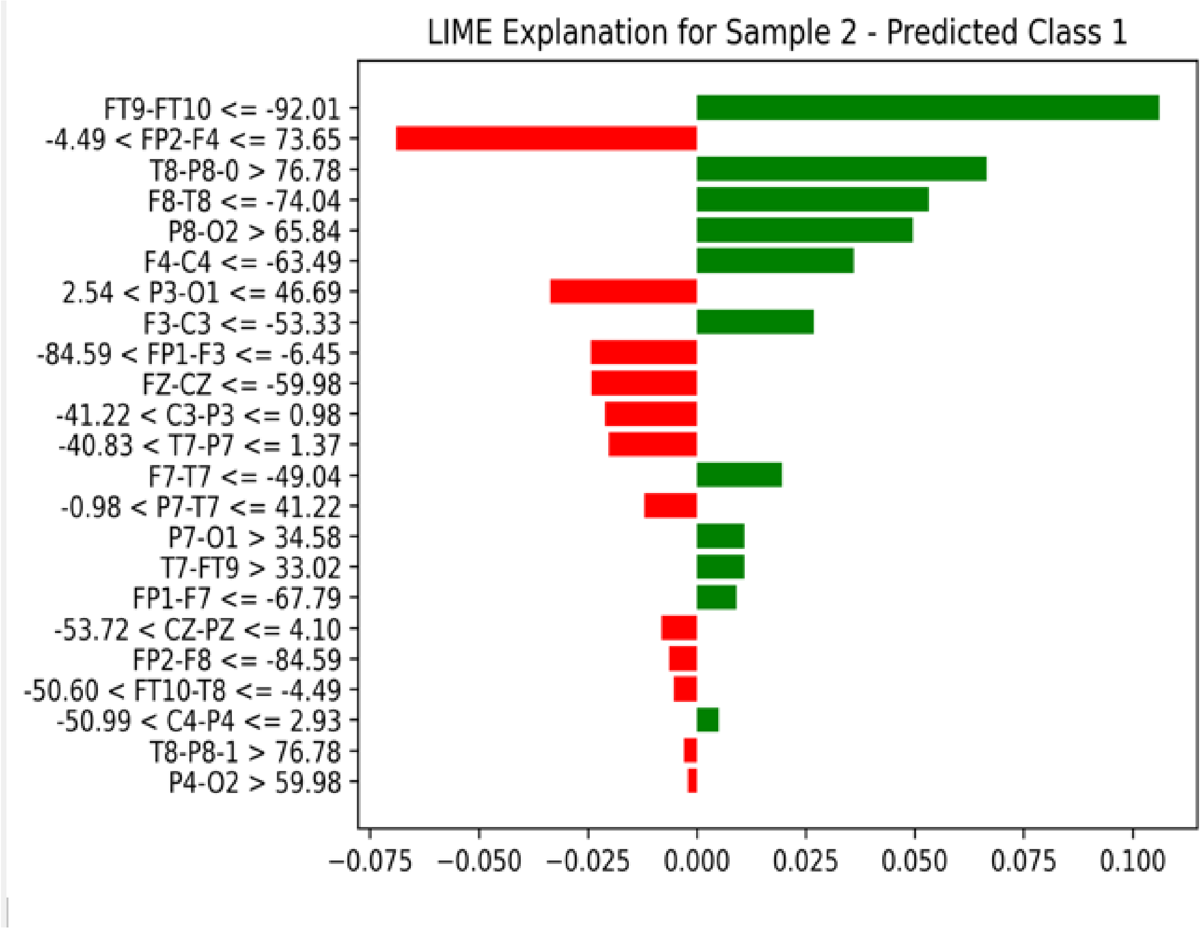


Figure 6.13: LIME Result

Chapter 7

Conclusions & Future Scope

With an emphasis on incorporating the interpretability strategies like SHAP and LIME the project effectively created an automated system for seizure localisation using a CNN-based model. For deep learning applications the system processes EEG data in the '.edf' format and converts it into a structured CSV file. After being trained on preprocessed and annotated data the machine learning model can accurately predict the seizure occurrences. By using explainable AI techniques, the model's predictions will be comprehensible, allowing for a deeper understanding of the decision-making process. By providing neurologists with an intuitive interface to visualise results and make informed decisions the system advances automated seizure detection.

The future scope of the project includes enhancing the interpretability features of the XAI models to improve the clarity and reliability of the insights generated by SHAP and LIME for better clinical decision making. By expanding the system to handle a greater range of EEG datasets such as those with different electrode configurations and sampling rates, its usefulness could be further increased. The addition of these real-time seizure detection features is another possible enhancement that would make the system to react rapidly while the EEG monitoring is underway. To help the neurologists create thorough and useful clinical reports the system could also be improved with advanced user interface features like automatic reporting tools.

References

[1] K. Garg *et al.*, "Epileptic seizure detection using machine learning and deep learning methods," *IEEE Xplore*, 2023, [Online]. Available: https://ieeexplore.ieee.org/document/9865313.

[2] A. Johnson *et al.*, "Epileptic seizure detection and prediction using deep learning techniques," in *IEEE Conference Publication*, 2022, [Online]. Available: https://ieeexplore.ieee.org/document/9740802.

[3] P. Chen *et al.*, "Epileptic seizure detection in eeg signals using machine learning and deep learning techniques," *IEEE Journals & Magazine*, 2024, [Online]. Available: https://ieeexplore.ieee.org/document/10547246.

[4] Y. Zhang *et al.*, "Efficient training of deep learning models with limited computational resources," *International Journal of Computer Science*, 2023.

[5] M. T. Ribeiro, S. Singh, and C. Guestrin, """why should i trust you?": Explaining the predictions of any classifier," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '16)*. New York, NY, USA: Association for Computing Machinery, 2016, pp. 1135–1144.

Appendix A: Presentation

Epileptic Seizure Localization from EEG Signals

FINAL PRESENTATION

Guide:
Dr. Sminu Izudheen
Professor
Dept. of CSE
RSET

Niyatha V S
Paul Allen
Powell Moothedan
Rose Jacob

Contents

Introduction

- Epilepsy is a neurological condition characterized by erratic signals from groups of neurons in the brain, leading to seizures, assessed using EEG.
- Examining the 10 to 15 hours of EEG data through visual analysis is time-consuming
- Our software tool streamlines this process for neurologists
- Enables automated localisation of seizure
- Provides interpretation to the decision made by the CNN model

Introduction

- To get a better understanding:

  1 hour EEG data = 1 * 60 * 60 * 256

  =9,21,600 data points = 3600 sheets

  40 hours EEG data = 9,21,600 * 40

  = 3,68,64,000 data points = 1,44,000 sheets

Problem Definition

- **Manual Analysis of EEG Signals**: Epileptic seizures are diagnosed using EEG recordings, but analyzing long hours of EEG signals is time-consuming and prone to human error.
- **Automated Detection**: Using Convolutional Neural Networks (CNNs) to automate seizure localization offers a solution to speed up diagnosis.
- **Explainability in AI**: Providing understandable explanations for the model's decisions is essential for clinicians to trust the automated system.

Novelty and Innovativeness

- Seizure Localization + Detection in One Pipeline
  - Combines both identification and spatial localization — bridging diagnostic and surgical planning needs.
- Integration of Deep Learning with Post-Hoc Explainability
  - Utilizes 2D CNNs for accurate seizure classification, enhanced with SHAP and LIME, offering interpretable justifications at both global and local levels — crucial for clinical adoption.
- Time-Resolved Interval Analysis for Seizure Prediction
  - Moves beyond point-wise detection to interval-based prediction, aligning more naturally with how seizures evolve over time, and improving early warning capabilities.

Novelty and Innovativeness

- Use of Statistical Thresholding to Define Seizure Onset
  - A novel approach using thresholds derived from non-seizure baseline EEG distributions, enabling personalized and robust detection adaptable to individual brain signal dynamics.
- Focus on Pediatric EEG (CHB-MIT/Bonn Dataset)
  - Tailors the system to pediatric epileptic profiles, a relatively underexplored area where seizure patterns and brain dynamics differ significantly from adults.
- Designed to Aid, Not Replace Clinicians
  - The system supports neurologists by highlighting patterns and channel contributions rather than offering black-box predictions.

Scope of Implementation

- Diagnostic Assistance
  - Assists neurologists in post-hoc review of EEG recordings by highlighting seizure intervals and most contributing channels, reducing manual effort and cognitive load.
- Support for Surgical Planning
  - Aids in seizure localization by identifying channels with high SHAP values and LIME values during ictal periods, providing insights into probable epileptogenic zones.

Scope of Implementation

- Educational Tool
  - Offers visual, interpretable outputs to help understand EEG patterns associated with seizures.
- Hospital-based Research Deployment
  - Useful in clinical research settings to retrospectively study EEG records and correlate machine learning outputs with patient outcomes.
- Interdisciplinary Collaboration Enabler
  - Bridges the gap between data science and neurology by offering interpretable and clinically aligned insights, encouraging collaborative improvements.

Literature Survey

PAPER	**ADVANTAGES**	**DISADVANTAGES**
Schirrmeister RT, Gemein LA, Eggensperger K, Hutter F, Ball T. XAI4EEG: Spectral and Spatio-Temporal Explanation of Deep Learning-Based Seizure Detection in EEG Time Series	**+** Provides consistent and accurate attributions **+** Model-agnostic; works with any machine learning model **+** Offers both local and global interpretability	**-** Computationally expensive for large datasets or complex models **-** Difficult to scale for real-time applications **-** Can be challenging to interpret in high-dimensional data

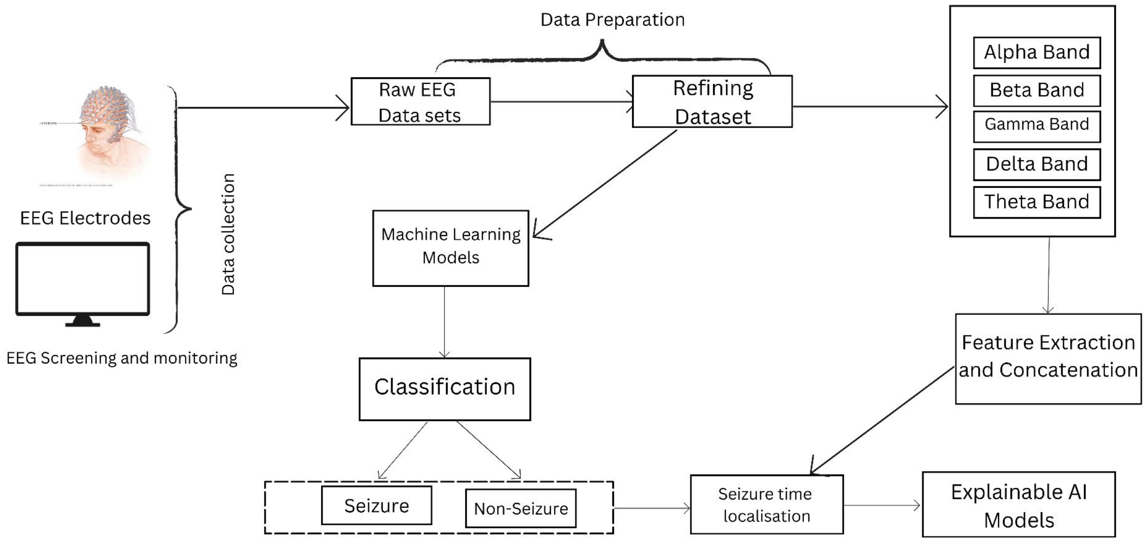
Literature Survey

PAPER	ADVANTAGES	DISADVANTAGES
Ribeiro MT, Singh S, Guestrin C. "Why Should I Trust You?": Explaining the Predictions of Any Classifier. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '16). New York, NY: ACM; 2016. p. 1135–44. doi:10.1145/2939672	**+** Model-agnostic; can explain any machine learning model **+** Fast and efficient for local explanations **+** Provides simple and interpretable explanations	**-** Sensitive to sampling; results may vary across different runs **-** Struggles with explaining highly complex or non-linear models **-** Limited global interpretability; focuses mainly on local explanations

Literature Survey

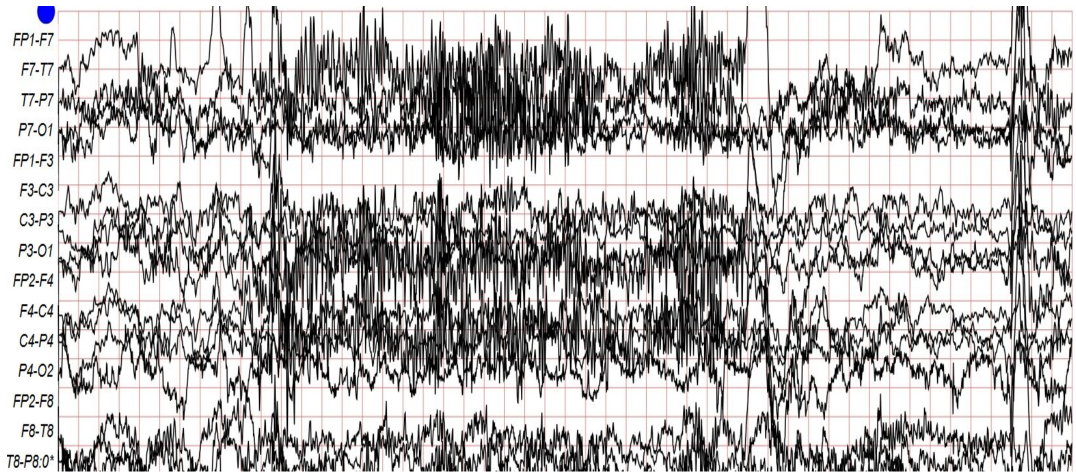
PAPER	ADVANTAGES	DISADVANTAGES
Inbaraj XA, Jeng JH. Mask-GradCAM: Object Identification and Localization of Visual Presentation for Deep Convolutional Network	**+** Highly effective for visualizing deep learning models, particularly CNNs **+** Produces class-specific heatmaps to show important regions in input data **+** Intuitive for image-related tasks and visual data	**-** Primarily designed for convolutional neural networks (CNNs); limited generalization to other model types **-** Resolution of heatmaps can be coarse **-** Difficult to apply to non-visual tasks like text or tabular data

Architecture Diagram

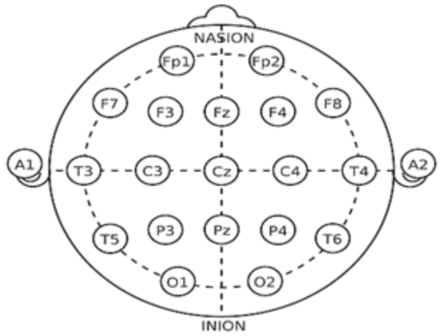


Data Preprocessing

- The .edf files were converted to .csv files where the columns represent the electrode potential values and the rows represent the time samples at a sampling rate of 256 Hz
- During preprocessing stage, the csv files are annotated as 1 where a seizure event was recorded in the summary files provided and 0 where non seizure events were recorded.



Input .edf file



10 - 20 standard electrode system

Annotated patient .csv files

Electrodes as per the 10 - 20 international standard

Seizure label as 0 or 1

time	FP1-F7	F7-T7	T7-P7	P7-O1	FP1-F3	F3-C3	C3-P3	P3-O1	FP2-F4	F4-C4	C4-P4	P4-O2	FP2-F8	F8-T8	T8-P8-0	P8-O2	FZ-CZ	CZ-PZ	P7-T7	T7-FT9	FT9-FT10	FT10-T8	T8-P8-1	SEIZURE_LABEL
0	25.9829	5.27473	52.5519	19.7314	26.7643	12.6984	84.591	29.4994	27.1551	49.8168	7.61905	0.19536	53.7241	72.0879	9.96337	31.0623	44.3468	71.6972	52.9426	18.5592	59.9756	75.0951	9.96337	0
0.00391	0.19536	0.19536	0.19536	0.19536	0.19536	0.19536	0.19536	0.19536	0.19536	0.19536	0.19536	0.19536	0.19536	0.19536	0.19536	0.19536	0.19536	0.19536	0.19536	0.19536	0.19536	0.19536	0.19536	0
0.00781	0.19536	0.19536	0.19536	0.19536	0.19536	0.19536	0.19536	0.19536	0.19536	0.19536	0.19536	0.19536	0.19536	0.19536	0.19536	0.19536	0.19536	0.19536	0.19536	0.19536	0.19536	0.19536	0.19536	0
0.01172	0.19536	0.19536	0.19536	0.19536	0.19536	0.19536	0.19536	0.19536	0.19536	0.19536	0.19536	0.19536	0.19536	0.19536	0.19536	0.19536	0.19536	0.19536	0.19536	0.58608	0.58608	-0.19536	0.19536	0
0.01563	0.19536	0.19536	0.19536	-0.19536	0.19536	0.19536	0.19536	-0.19536	0.19536	0.19536	0.19536	-0.19536	0.19536	0.19536	0.19536	-0.19536	0.19536	-0.19536	0.19536	0.58608	-0.9768	0.9768	0.19536	0
0.01953	0.58608	1.36752	-1.36752	-0.58608	0.58608	0.19536	-0.19536	-0.58608	0.19536	0.19536	0.19536	-0.58608	0.19536	-0.19536	0.19536	-0.58608	-0.58608	0.19536	1.75824	-2.9304	-0.9768	2.53968	0.19536	0
0.02344	0.58608	0.58608	-1.36752	0.58608	0.9768	0.19536	-0.58608	0.58608	-0.19536	0.19536	0.19536	0.9768	0.19536	-0.19536	0.19536	0.9768	-0.19536	0.9768	1.75824	-4.49328	2.14896	0.9768	0.19536	0
0.02734	-0.19536	-3.71184	3.32112	1.36752	-0.19536	-0.19536	-0.19536	1.75824	-0.19536	0.19536	-0.58608	1.75824	0.19536	0.58608	-0.58608	1.75824	0.19536	0.9768	-2.9304	4.884	1.75824	-1.75824	-0.58608	0
0.03125	-0.19536	-1.75824	3.32112	-0.9768	-0.58608	0.19536	0.9768	-0.58608	0.19536	0.58608	0.58608	-0.9768	0.58608	0.58608	0.19536	-0.9768	0.19536	-0.19536	-2.9304	8.79121	-2.14896	-2.14896	0.19536	0
0.03516	0.58608	3.71184	-3.32112	-1.75824	0.58608	0.19536	0.19536	-1.75824	0.19536	0.58608	1.36752	-2.53968	0.58608	-0.19536	1.75824	-2.14896	-0.58608	-0.19536	3.71184	-3.71184	-1.75824	1.75824	1.75824	0
0.03906	0.9768	4.10256	-6.44689	-0.19536	1.75824	-0.58608	-1.75824	-1.36752	-0.19536	0.19536	0.19536	-1.75824	-0.19536	-0.58608	0.19536	-0.9768	-2.9304	1.36752	6.83761	-15.4335	-1.75824	11.917	0.19536	0
0.04297	1.36752	2.14896	-7.22833	-6.83761	4.49328	-0.58608	-5.66545	-8.00977	-0.58608	-1.36752	1.36752	-9.57265	1.36752	-4.10256	0.19536	-8.40049	-10.3541	0.19536	7.61905	-18.5592	-14.2613	26.7643	0.19536	0
0.04688	6.05617	2.14896	-11.917	-6.05617	8.00977	-2.53968	-9.96337	-4.10256	-3.71184	0.58608	1.36752	-4.884	1.36752	-6.44689	1.36752	-4.49328	-12.3077	6.44689	12.3077	-29.1087	0.9768	14.652	1.36752	0
0.05078	3.32112	-14.2613	-4.49328	-0.19536	6.05617	-7.22833	-14.652	0.19536	-6.44689	2.53968	-1.36752	-0.19536	-1.75824	-0.19536	-3.71184	-0.19536	-16.2149	13.0891	4.884	-12.6984	-2.14896	20.1221	-3.71184	0
0.05469	0.58608	-32.2344	10.7448	-4.49328	1.36752	-9.18193	-10.7448	-5.27473	-7.22833	8.00977	0.58608	-6.83761	2.53968	0.9768	-2.53968	-6.83761	-20.5128	16.2149	-10.3541	15.8242	-5.66545	17.7778	-2.53968	0
0.05859	2.9304	-2.53968	-14.652	-11.1355	2.9304	-9.18193	-8.00977	-9.96337	-6.83761	15.4335	6.44689	-11.1355	4.49328	4.49328	6.05617	-12.6984	-18.5592	21.685	15.0427	-13.0891	4.10256	8.40049	6.05617	0
0.0625	2.53968	-13.8706	-13.8706	-0.9768	-0.19536	-13.0891	-7.22833	-4.49328	-10.7448	19.7314	6.44689	-8.00977	4.49328	4.49328	7.61905	-9.96337	-17.3871	28.3272	14.2613	-4.10256	-1.36752	16.9963	7.61905	0
0.06641	0.19536	4.10256	-27.5458	-2.9304	-2.53968	-11.1355	-3.32112	-7.61905	-11.1355	24.8107	9.18193	-9.96337	9.18193	6.05617	10.3541	-13.8706	-16.9963	32.6252	27.9365	-33.0159	9.96337	26.3736	10.3541	0
0.07031	3.32112	12.6984	-35.7509	-2.53968	-3.71184	-8.79121	-0.9768	-5.66545	-12.3077	29.4994	15.4335	-13.4799	9.57265	8.79121	16.2149	-17.7778	-11.5263	33.4066	36.1416	-24.42	7.22833	14.2613	16.2149	0

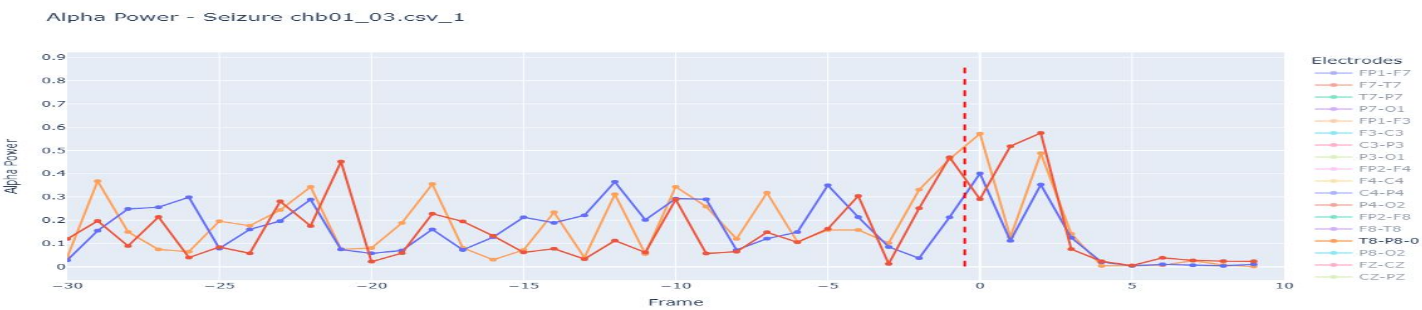
Time samples at sampling rate of 256 Hz which means each 1 second is divided into 256 samples

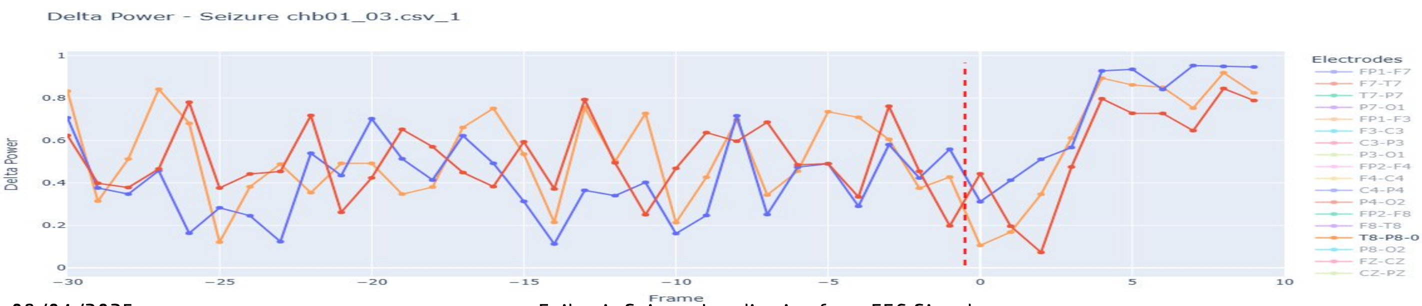
Electrode potential value

Feature Extraction

- Considers 30 one second long preictal frames and one second ictal frames
- Relative power is calculated by integrating the Power Spectral Density within the specified frequency bands and normalizing it by the total power across the entire frequency range
- Analysis of particularly the power spectral density of alpha and delta waves revealed that the relative power in the alpha band was higher during the preictal phase and the alpha waves convert to delta waves during the ictal phase, also known as the alpha slowing phenomenon
- This approach will help in localising the origin of the seizure spikes by considering interframe dependencies

Feature Extraction





Patient Specific CNN Model

Convolutional Blocks (Feature Extraction)

- Convolutional Blocks (Feature Extraction)
  - Conv2D (32 filters, 3x3)
  - Extracts low-level spatial features like edges or activations from EEG frames.
- BatchNormalization
  - Stabilizes learning and speeds up convergence.
- MaxPooling2D
  - Reduces spatial dimensions, preserves dominant features.
- Dropout
  - Prevents overfitting by randomly deactivating neurons during training.

Patient Specific CNN Model

Deeper Feature Extraction Layers

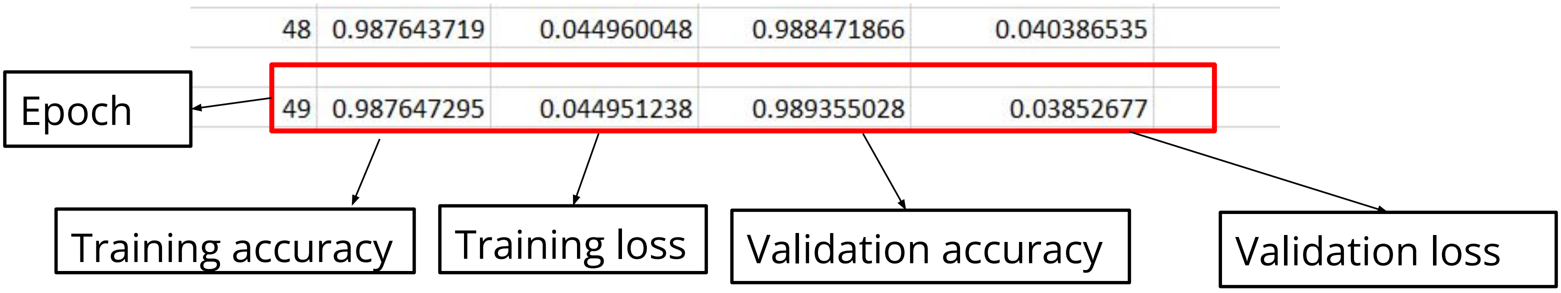
- Conv2D (64 filters)
  - Learns more complex patterns in reduced spatial dimensions.
- BatchNormalization + MaxPooling + Dropout
  - Repeated combination to normalize, downsample, and regularize the data.
- Conv2D (128 filters)
  - Final convolutional block for rich, abstract feature learning.
- BatchNormalization + MaxPooling2D + Dropout
  - Final feature compression and regularization layer.

Patient Specific CNN Model

Dense Layers (Classification Head)

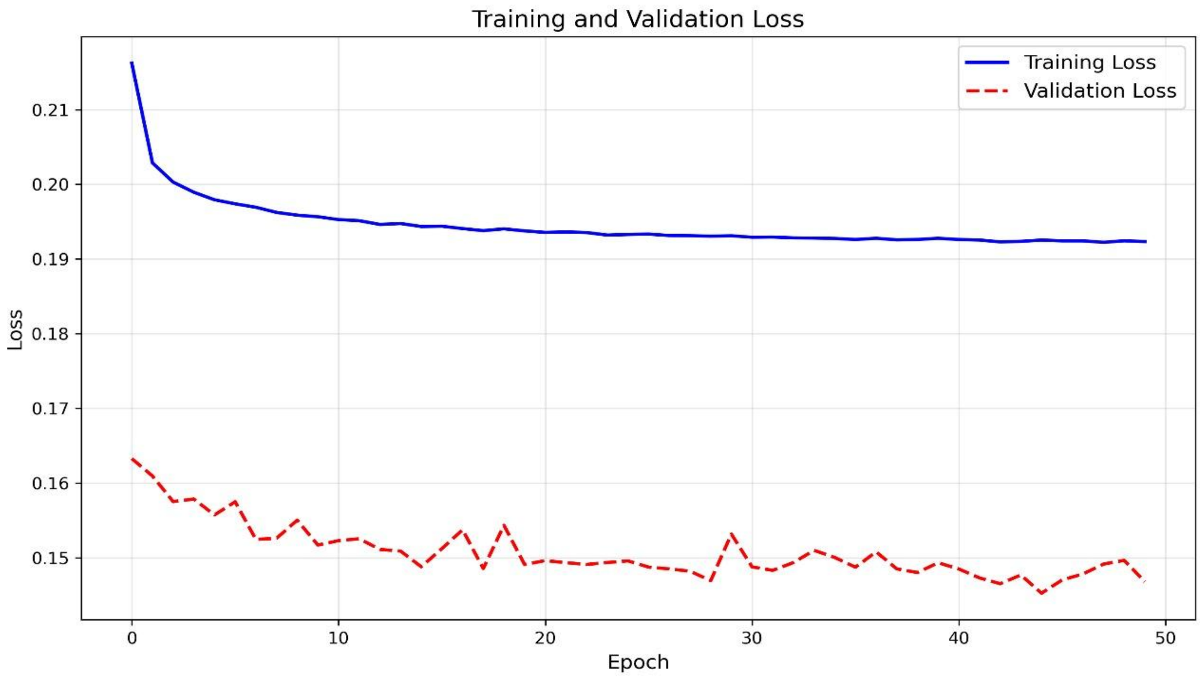
- Flatten
    - Converts 2D feature maps to 1D vector for dense layers.
- Dense (128 units)
    - Fully connected layer for learning nonlinear combinations of features.
- Dropout
    - Further regularization.
- Dense (1 unit, output layer)
    - Final prediction – seizure (1) or non-seizure (0).

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 21, 1, 32)	128
batch_normalization (BatchNormalization)	(None, 21, 1, 32)	128
max_pooling2d (MaxPooling2D)	(None, 10, 1, 32)	0
dropout (Dropout)	(None, 10, 1, 32)	0
conv2d_1 (Conv2D)	(None, 8, 1, 64)	6,208
batch_normalization_1 (BatchNormalization)	(None, 8, 1, 64)	256
max_pooling2d_1 (MaxPooling2D)	(None, 4, 1, 64)	0
dropout_1 (Dropout)	(None, 4, 1, 64)	0
conv2d_2 (Conv2D)	(None, 2, 1, 128)	24,704
batch_normalization_2 (BatchNormalization)	(None, 2, 1, 128)	512
max_pooling2d_2 (MaxPooling2D)	(None, 1, 1, 128)	0
dropout_2 (Dropout)	(None, 1, 1, 128)	0
flatten (Flatten)	(None, 128)	0
dense (Dense)	(None, 128)	16,512
dropout_3 (Dropout)	(None, 128)	0
dense_1 (Dense)	(None, 1)	129



48	0.987643719	0.044960048	0.988471866	0.040386535
49	0.987647295	0.044951238	0.989355028	0.03852677

Loss curve



Automation of Threshold Fixing for Time Prediction Algorithm

- The CNN model threshold (like 0.5) is used for binary classification of frames, but is not optimal for temporal localization of seizures.
- Seizure dynamics vary between patients: onset pattern, duration, and signal strength differ.
- A global threshold may miss weak seizures or falsely detect strong noise.
- Hence, custom thresholds per patient improve prediction reliability.
- Manual threshold tuning is time-consuming and prone to bias.
- EEG patterns vary by age, condition, and electrode placement.
- Automation ensures adaptive, robust, and personalized seizure time prediction without human supervision.
- Our approach uses a data-driven method for threshold calibration.

Automation of Threshold Fixing for Time Prediction Algorithm

- Apply a sliding window of size 30 and stride size 1 across the EEG to compute the Delta-Alpha Ratio (D-A-R) at each second.
- It had been observed that the D-A-R peaks coincide with seizure periods.
- It had also been observed that the D-A-R from peak to actual seizure onset ≥ 1.9.
- This gives a possible seizure onset time window.
- From the predicted seizure onset, we consider:
    - 30 preictal frames (1 second each, before onset).
    - 10 ictal frames (1 second each, after onset).
- These 40 frames are used to extract delta values for statistical modeling.

Automation of Threshold Fixing for Time Prediction Algorithm

The following statistical methods are applied to these values:

- MAD (Median Absolute Deviation):
  - A robust dispersion measure.

    MAD Threshold = median + (scale factor × MAD)

    where, MAD = median(|Xi − median(X)|)

Automation of Threshold Fixing for Time Prediction Algorithm

- GMM (Gaussian Mixture Model):
  - A GMM with two components is fitted on the delta values.
  - The minimum of the GMM means is taken as a candidate threshold, assuming it represents the preictal (non-seizure) cluster.

$$p(X) = \sum_{i=1}^{K} w_i \mathcal{N}(X \mid \mu_i, \sigma_i^2)$$

p(X) - probability density
K - number of Gaussian components
wi - weight of i th component
μ - center of the ith gaussian
σ - spread of the component

Automation of Threshold Fixing for Time Prediction Algorithm

- IQR (Interquartile Range):
  - Captures spread by computing Q1 and Q3.

    IQR = Q3 − Q1, where Q3 is 75th percentile and Q1 is 25th percentile
- Skewness:
  - A measure of signal distribution asymmetry that affects the threshold's appropriate shift.

$$\text{Skewness} = \frac{\frac{1}{N}\sum_{i=1}^{N}(X_i-\mu)^3}{\left(\frac{1}{N}\sum_{i=1}^{N}(X_i-\mu)^2\right)^{3/2}}$$

Xi is the EEG value
μ is the mean
N is the number of samples.

Automation of Threshold Fixing for Time Prediction Algorithm

- The base threshold is adaptively tuned by adding an auto-correction factor that is calculated using these features:

  Correction = 0.05 · log(1 + IQR) · tanh(Skewness)
- The final threshold is calculated as,

  Final Threshold = α · TGMM + (1 − α) · TMAD + Correction

  where, α is the weight parameter balancing GMM and MAD thresholds

Automation of Threshold Fixing for Time Prediction Algorithm

- After identifying a statistical threshold using preictal Delta-Alpha-Ratio (DAR) data through GMM, MAD, IQR, and kurtosis ensemble methods, this calibrated threshold is used to refine the CNN model's output.
- The CNN model predicts seizure likelihood for each EEG frame.
- Instead of using an arbitrary fixed threshold (e.g., 0.5), the calibrated, patient-specific threshold is applied to convert continuous prediction scores into binary labels (0 or 1).

Automation of Threshold Fixing for Time Prediction Algorithm

- This ensures:
  - Alignment between DAR-based seizure zones and CNN predictions
  - Adaptability across patients with varying EEG dynamics
  - Minimized false alarms and improved temporal accuracy
- Once predictions are binarized using this threshold:
  - Consecutive or closely spaced seizure-labeled frames are merged
  - Short noisy predictions (e.g., <10 seconds) are discarded
- The final output is a list of precise seizure intervals

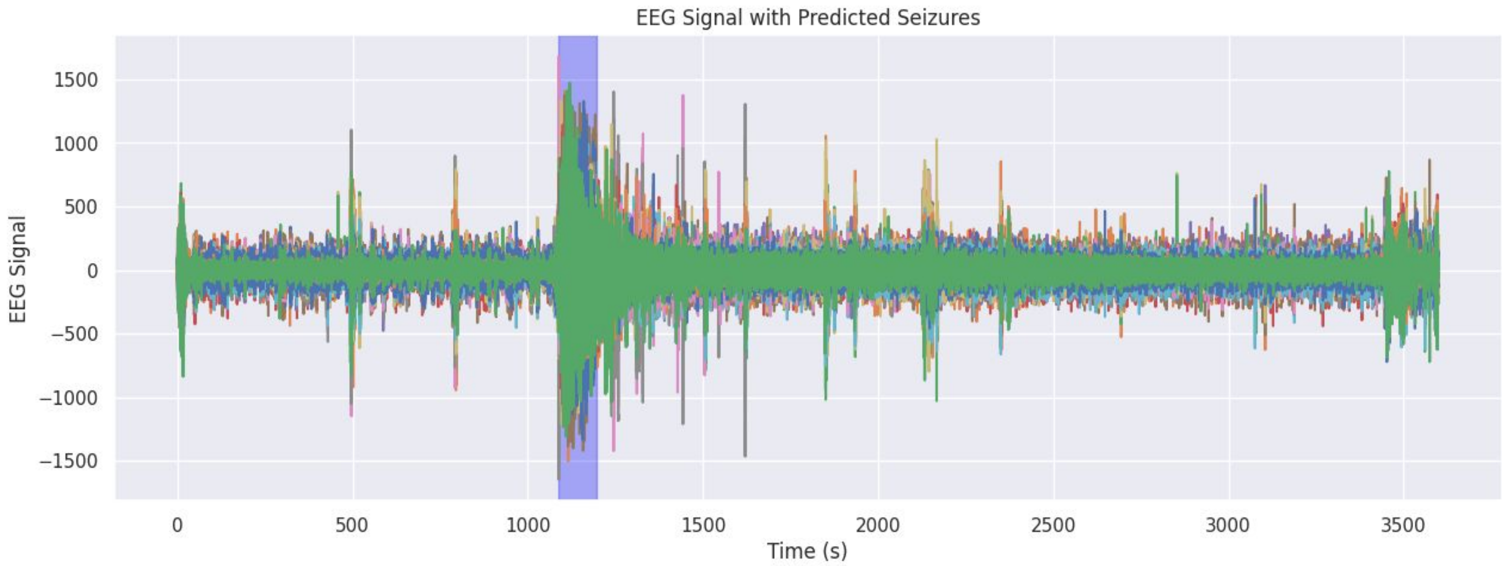
Patient number	Threshold calculated using the statistical approach
Patient 1	0.76
Patient 3	0.73
Patient 5	0.89
Patient 11	0.73
Patient 22	0.78

CH01		
FILES	ACTUAL	PREDICTED
chb01_01.csv		
chb01_03.csv	2996 - 3036	3001.191406 - 3035.289063
chb01_04.csv	1467 - 1494	1468.824219 - 1492.207031
chb01_15.csv	1732 - 1772	1735.585938 - 1777.367188
chb01_16.csv	1015 - 1066	1015.078125 - 1065.722656
chb01_18.csv	1720 - 1810	1725.449219 - 1808.660156
chb01_21.csv	327 - 420	330.09375 - 4190.2226563
chb01_26.csv	1862 - 1963	1867.820313 - 1962.410156
chb01_27.csv		
chb01_29.csv		108.7265625 - 121.703125 & 645.6953125 - 655.984375

Time prediction made by using the threshold that was calculated using statistical methods which was 0.76 for patient 1

CH05		
FILES	ACTUAL	PREDICTED
chb05_01.csv		
chb05_06.csv	417 - 532	425 - 531
chb05_13.csv	1086 - 1196	1087 - 1196
chb05_16.csv	2317 - 2413	2321 - 2413
chb05_17.csv	2451 - 2571	2453 - 2469
chb05_22.csv	2348 - 2465	2344 - 2464

Time prediction made by using the threshold that was calculated using statistical methods which was 0.89 for patient 5



Plot with the predicted time highlighted in blue

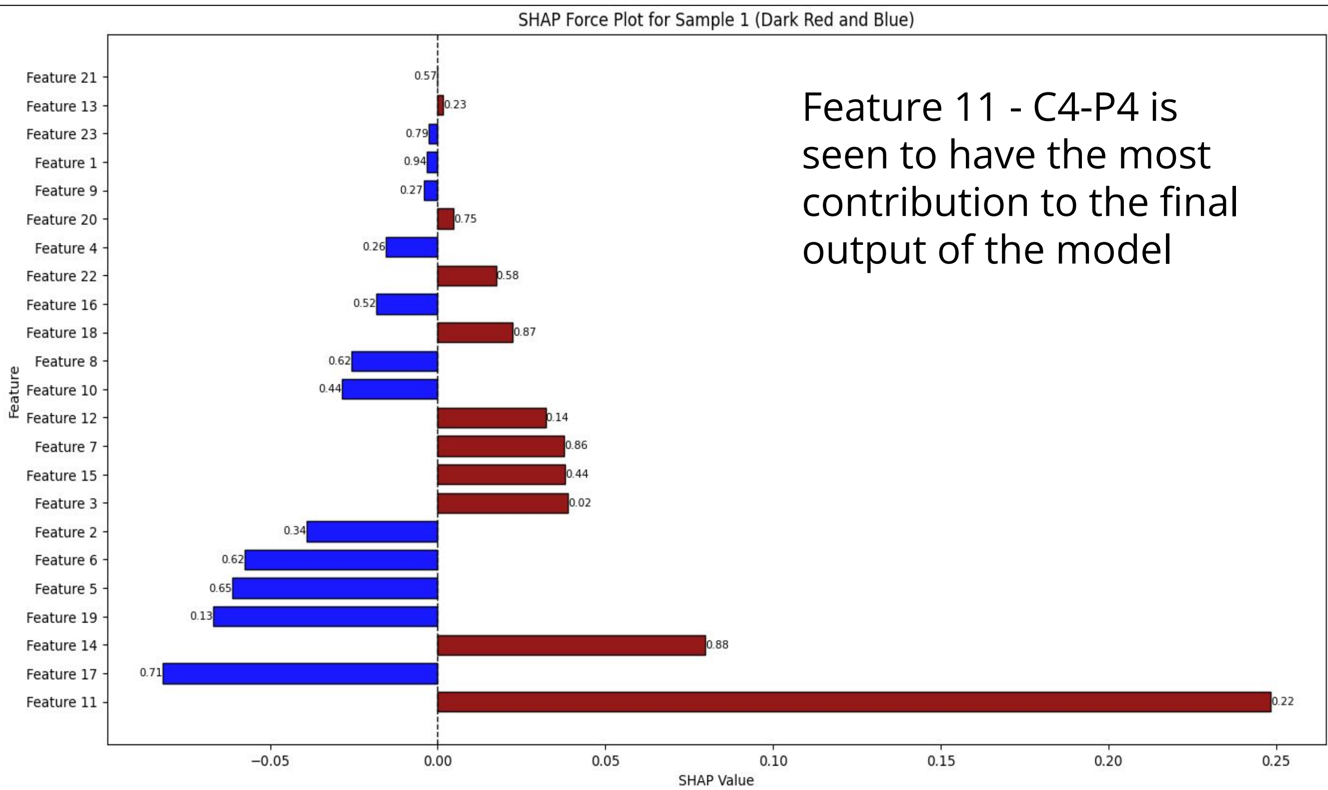
Explainable AI Model - SHAP

- SHAP (SHapley Additive exPlanations) is a model-agnostic interpretability technique based on cooperative game theory. It assigns each feature an importance value for a particular prediction.
- It provides global and local insights into model behavior while maintaining consistency and fairness in feature attribution.
- SHAP calculates the marginal contribution of each feature by evaluating all possible combinations (feature coalitions), ensuring each feature’s impact is fairly represented.
- By aggregating SHAP values over many samples, we obtain a global view of which features consistently influence model predictions.
- Helps identify most influential EEG biomarkers or frequency bands in seizure prediction.
- SHAP KernelExplainer was used in this study as a model-agnostic approach to interpret the CNN's predictions.

Explainable AI Model - SHAP

$$\phi_i = \sum_{S \subseteq N \setminus \{i\}} \frac{|S|!(|N| - |S| - 1)!}{|N|!} [f(S \cup \{i\}) - f(S)]$$

- Interpreting the Plot:
  - X-axis (SHAP value) shows how much that feature pushes the prediction higher or lower.
  - Color gradient (e.g., red to blue) represents the original feature value.
  - A wider spread → higher impact; clustering near zero → minimal influence.
- SHAP allows doctors to see global trends across patients.



Feature 11 - C4-P4 is seen to have the most contribution to the final output of the model

Explainable AI Model - LIME

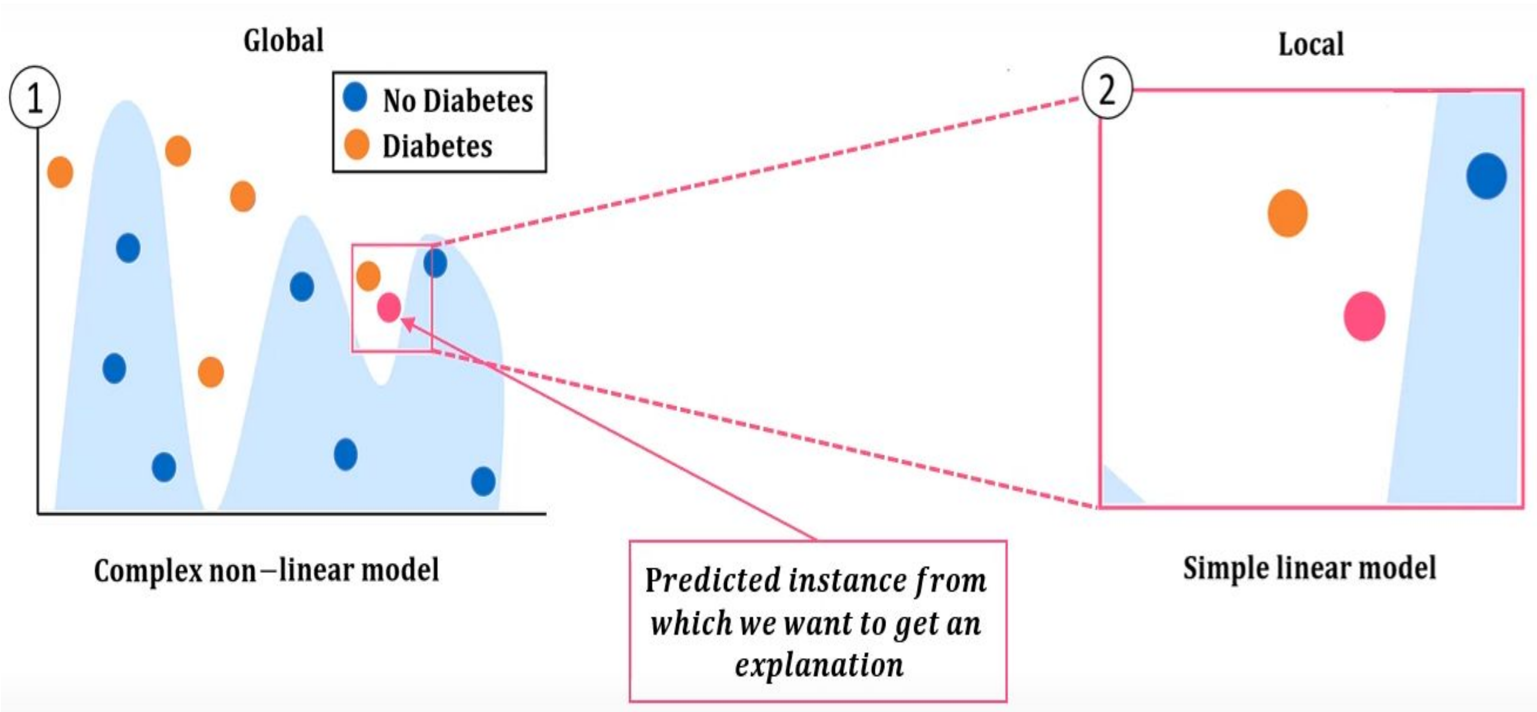
- SHAP gives an overview of how each channel behaves overall.
- LIME allows case-by-case breakdowns, mimicking a doctor analyzing an individual patient's snapshot.
- Explains the prediction of an individual sample by approximating the model locally.
- Perturbs the input sample slightly (e.g., mask or modify EEG channels).
- Observes model predictions for these perturbed versions.
- Fit a simple, interpretable model (e.g., linear regression) on the perturbed data and predictions.

Explainable AI Model - LIME

- This surrogate model approximates the black-box model in the vicinity of the original input.
- Explains what features (channels) most influenced the prediction for this specific time window.
- Bar plot output:
  - Features (EEG channels) are shown with weights.
  - Positive weight → feature supports the model’s prediction (e.g., seizure).
  - Negative weight → feature pushes against the prediction.

Explainable AI Model - LIME

- Magnitude of weight shows how strongly the feature impacted the output.
- Highlights which channels are most significant at a given moment.
- Helps validate whether the model aligns with known seizure patterns.
- Doctors can visualize temporal-spatial patterns of seizures and verify if model reasoning makes clinical sense.



Input Data

I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X
P3-O1	FP2-F4	F4-C4	C4-P4	P4-O2	FP2-F8	F8-T8	T8-P8-0	P8-O2	F7-C7	C7-P7	P7-T7	T7-FT9	FT9-FT10	FT10-T8	T8-P8-1
300.6593	179.9267	80.29304	-67.0085	161.1722	166.2515	37.3138	59.58486	91.23321	11.13553	77.558	0.586081	54.11477	26.76435	55.28694	59.58486
0.19536	0.19536	0.19536	0.19536	0.19536	0.19536	0.19536	0.19536	0.19536	0.19536	0.19536	0.19536	0.19536	0.19536	0.19536	0.19536
0.586081	0.19536	0.19536	0.19536	0.976801	0.19536	0.19536	0.19536	0.976801	0.19536	0.19536	0.19536	0.19536	1.758242	-0.9768	0.19536
2.148962	-0.9768	-1.75824	-0.19536	3.711844	-1.36752	-0.19536	0.586081	1.758242	0.19536	-0.58608	1.758242	-2.53968	2.930403	-0.19536	0.586081
-2.53968	0.586081	-2.9304	2.930403	-5.27473	-0.19536	1.758242	-0.19536	-5.66545	0.19536	0.976801	0.19536	-2.53968	-12.6984	16.60562	-0.19536

$$x = [300.6593, 179.9267, 80.29304, \ldots, 55.28694, 59.58486]$$

Perturbate

1. **Sample 1:**

   $$x_1 = [301.1593, 180.4267, 80.79304, \ldots, 55.78694, 60.08486]$$

2. **Sample 2:**

   $$x_2 = [300.1593, 179.4267, 79.79304, \ldots, 54.78694, 59.08486]$$

3. **Sample 3:**

   $$x_3 = [300.6593, 180.4267, 80.79304, \ldots, 55.28694, 60.58486]$$

4. **Sample 4:**

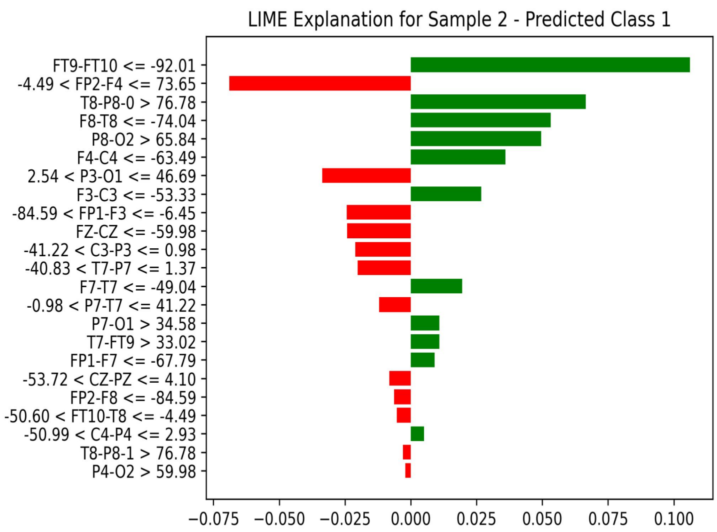
   $$x_4 = [301.6593, 179.9267, 81.29304, \ldots, 55.78694, 59.58486]$$

5. **Sample 5:**

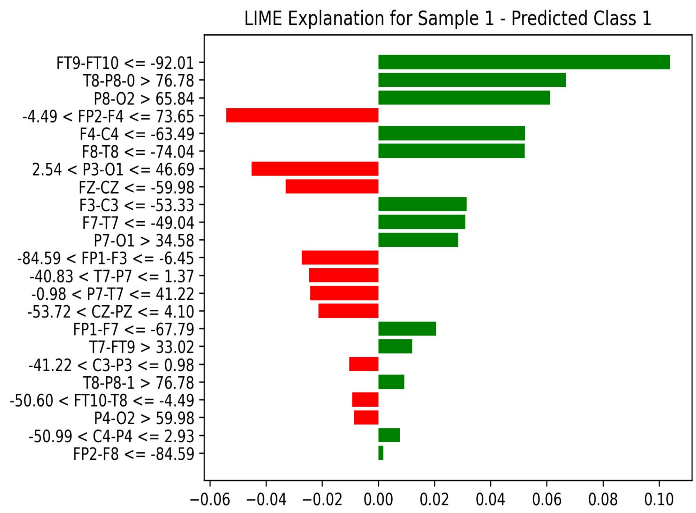
   $$x_5 = [299.6593, 180.9267, 79.29304, \ldots, 54.28694, 60.08486]$$

Perturbate

$$X = \begin{bmatrix} 300.6593 & 179.9267 & 80.29304 & \dots & 55.28694 & 59.58486 \\ 301.1593 & 180.4267 & 80.79304 & \dots & 55.78694 & 60.08486 \\ 300.1593 & 179.4267 & 79.79304 & \dots & 54.78694 & 59.08486 \\ 300.6593 & 180.4267 & 80.79304 & \dots & 55.28694 & 60.58486 \\ 301.6593 & 179.9267 & 81.29304 & \dots & 55.78694 & 59.58486 \\ 299.6593 & 180.9267 & 79.29304 & \dots & 54.28694 & 60.08486 \end{bmatrix} \quad y = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 1 \\ 0 \end{bmatrix}$$

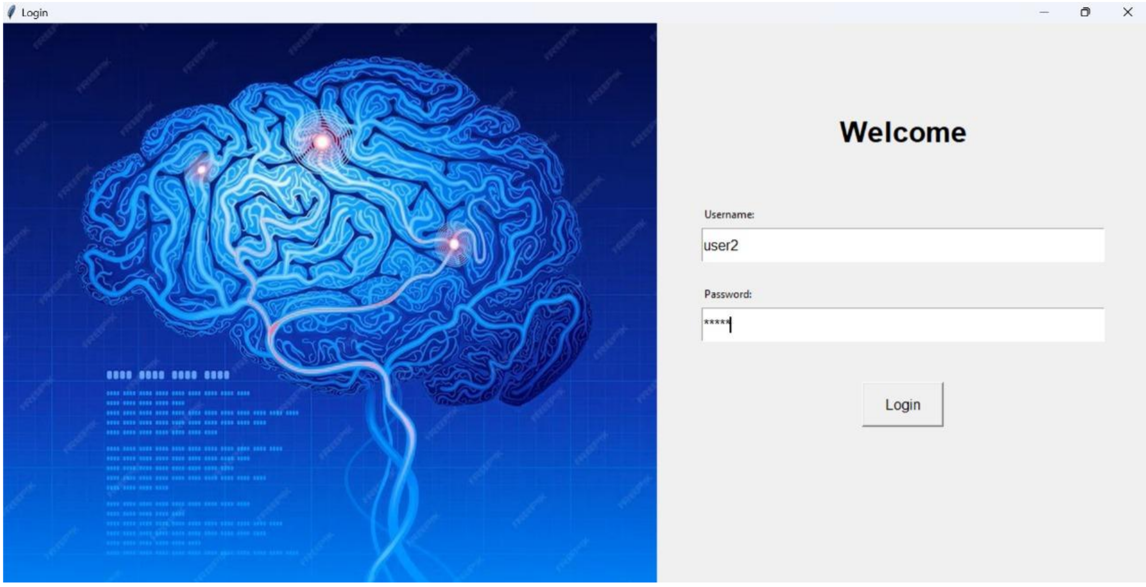


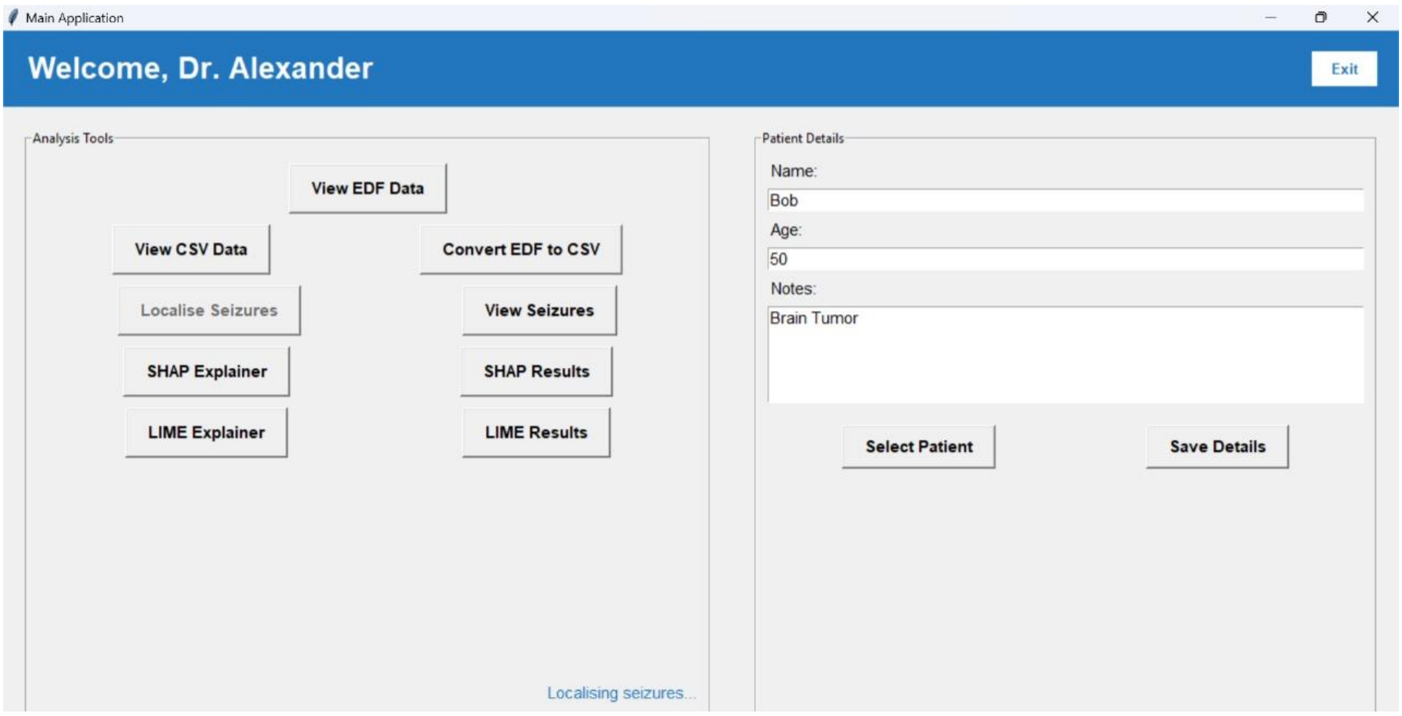
- The feature “FT9-FT10 <= -92.01” has the most positive impact on the prediction for Class 1.
- The feature “-4.49 < FP2-F4 <=73.65” has the most negative impact on the prediction for Class 1.



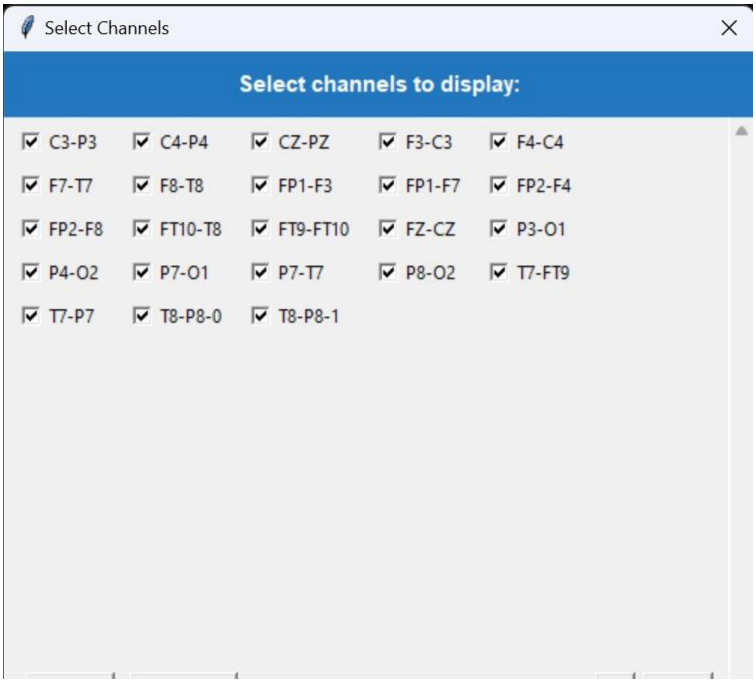
- The feature “FT9-FT10 <= -92.01” has the most positive impact on the prediction for Class 1.
- The feature “-4.49 < FP2-F4 <=73.65” has the most negative impact on the prediction for Class 1.

GUI



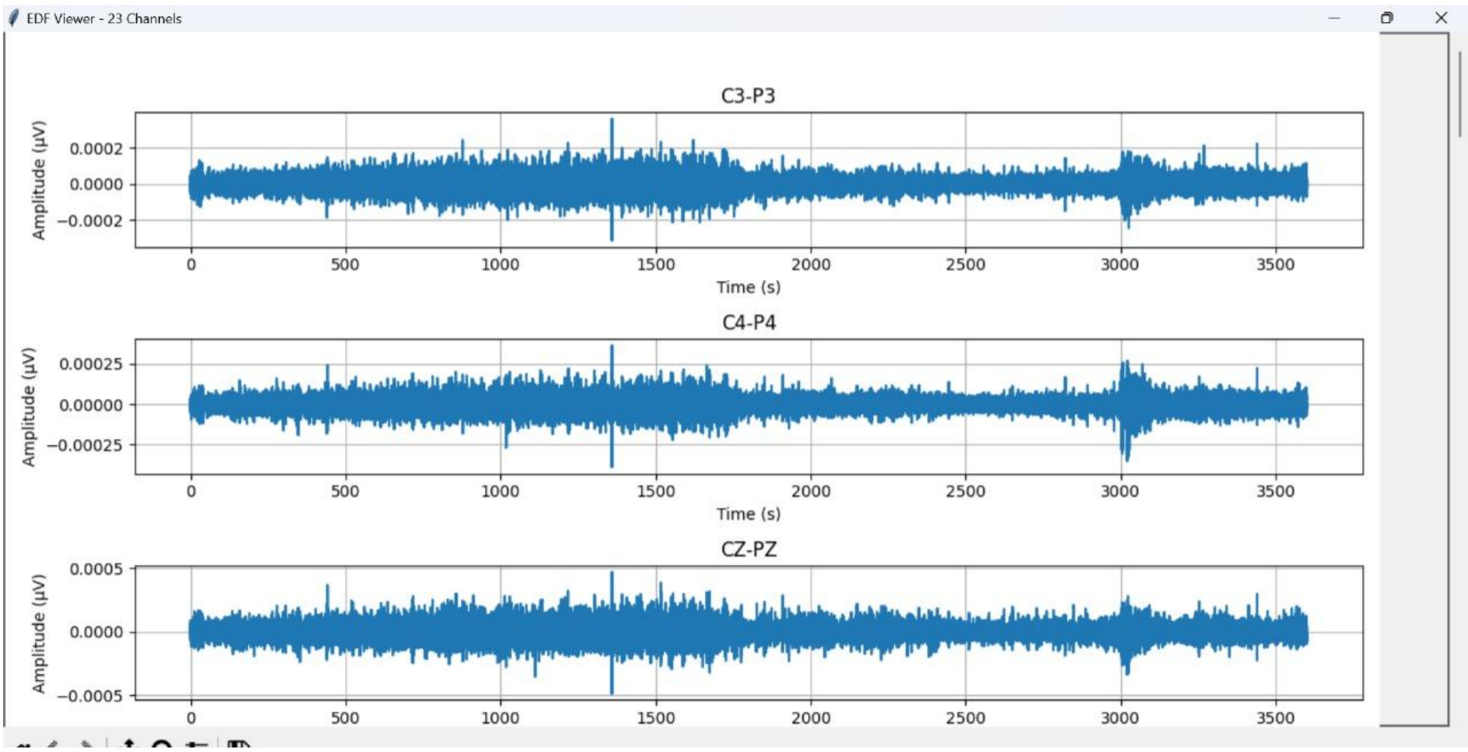


Home page

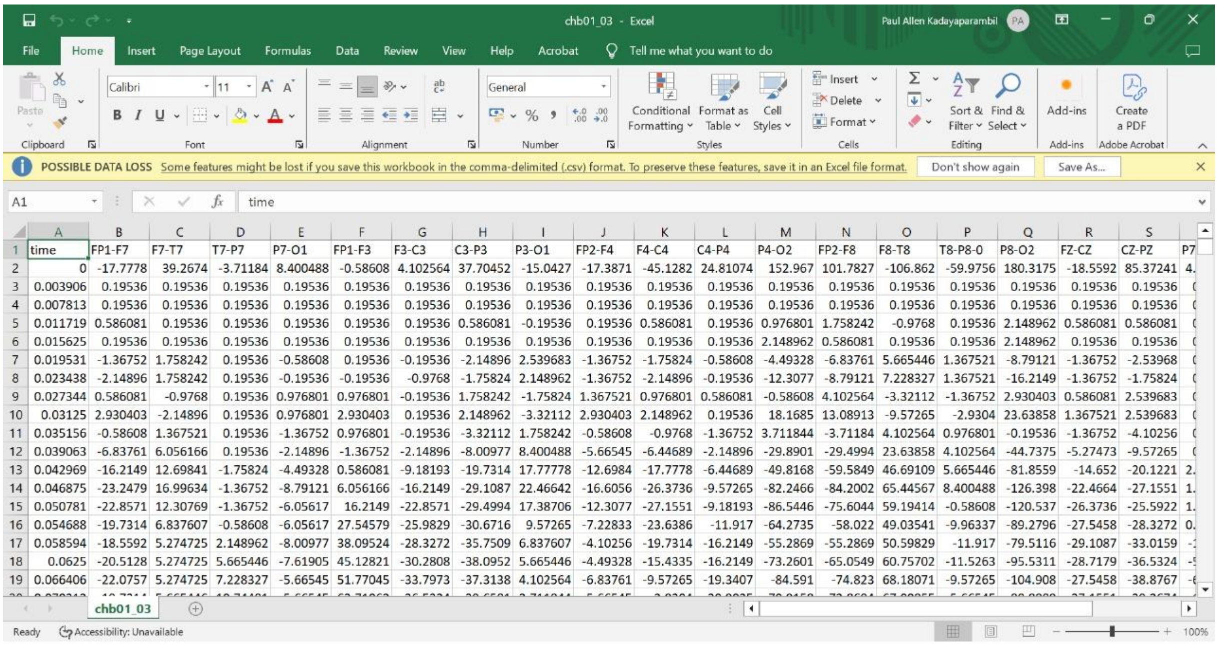


View EDF Files Button:

Option to select the required channels



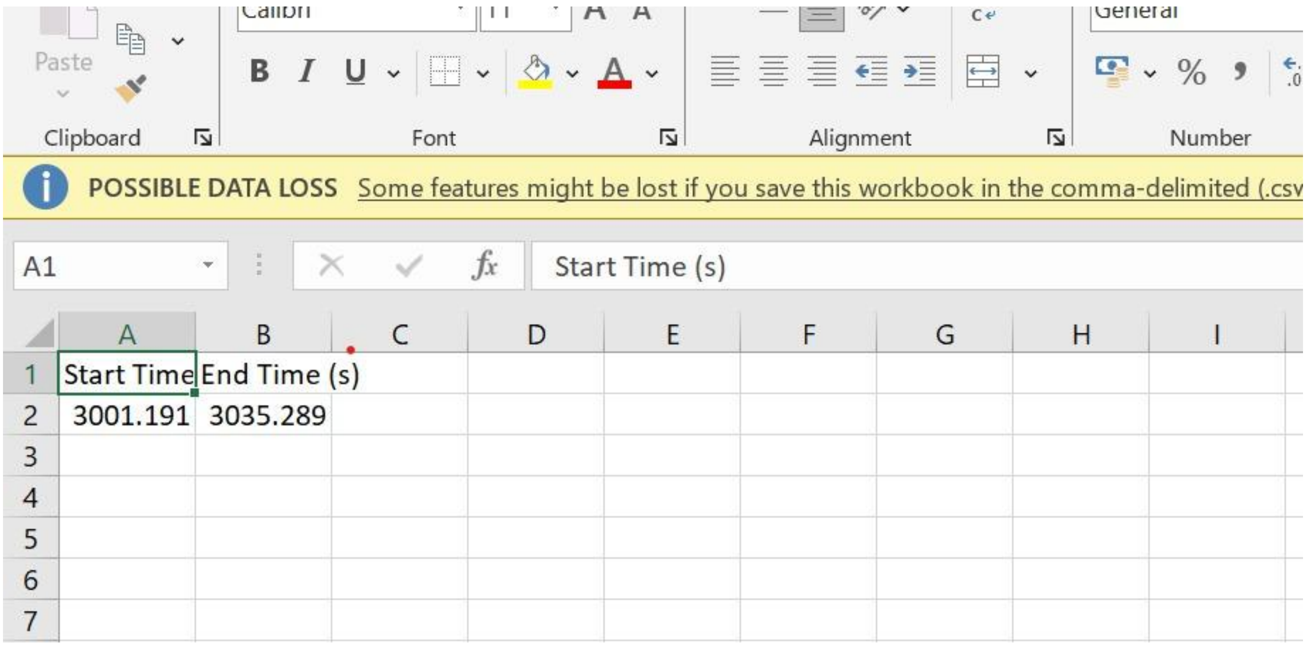
EDF Data Visualised for the selected electrodes



	A	B	C	D	E	F	G	H	I	J
1	time	FP1-F7	F7-T7	T7-P7	P7-O1	FP1-F3	F3-C3	C3-P3	P3-O1	FP2-F4
2	0	-17.7778	39.2674	-3.71184	8.400488	-0.58608	4.102564	37.70452	-15.0427	-17.3871
3	0.003906	0.19536	0.19536	0.19536	0.19536	0.19536	0.19536	0.19536	0.19536	0.19536
4	0.007813	0.19536	0.19536	0.19536	0.19536	0.19536	0.19536	0.19536	0.19536	0.19536
5	0.011719	0.586081	0.19536	0.19536	0.19536	0.19536	0.19536	0.586081	-0.19536	0.19536
6	0.015625	0.19536	0.19536	0.19536	0.19536	0.19536	0.19536	0.19536	0.19536	0.19536
7	0.019531	-1.36752	1.758242	0.19536	-0.58608	0.19536	-0.19536	-2.14896	2.539683	-1.36752
8	0.023438	-2.14896	1.758242	0.19536	-0.19536	-0.19536	-0.9768	-1.75824	2.148962	-1.36752
9	0.027344	0.586081	-0.9768	0.19536	0.976801	0.976801	-0.19536	1.758242	-1.75824	1.367521
10	0.03125	2.930403	-2.14896	0.19536	0.976801	2.930403	0.19536	2.148962	-3.32112	2.930403
11	0.035156	-0.58608	1.367521	0.19536	-1.36752	0.976801	-0.19536	-3.32112	1.758242	-0.58608
12	0.039063	-6.83761	6.056166	0.19536	-2.14896	-1.36752	-2.14896	-8.00977	8.400488	-5.66545
13	0.042969	-16.2149	12.69841	-1.75824	-4.49328	0.586081	-9.18193	-19.7314	17.77778	-12.6984
14	0.046875	-23.2479	16.99634	-1.36752	-8.79121	6.056166	-16.2149	-29.1087	22.46642	-16.6056
15	0.050781	-22.8571	12.30769	-1.36752	-6.05617	16.2149	-22.8571	-29.4994	17.38706	-12.3077
16	0.054688	-19.7314	6.837607	-0.58608	-6.05617	27.54579	-25.9829	-30.6716	9.57265	-7.22833
17	0.058594	-18.5592	5.274725	2.148962	-8.00977	38.09524	-28.3272	-35.7509	6.837607	-4.10256
18	0.0625	-20.5128	5.274725	5.665446	-7.61905	45.12821	-30.2808	-38.0952	5.665446	-4.49328
19	0.066406	-22.0757	5.274725	7.228327	-5.66545	51.77045	-33.7973	-37.3138	4.102564	-6.83761

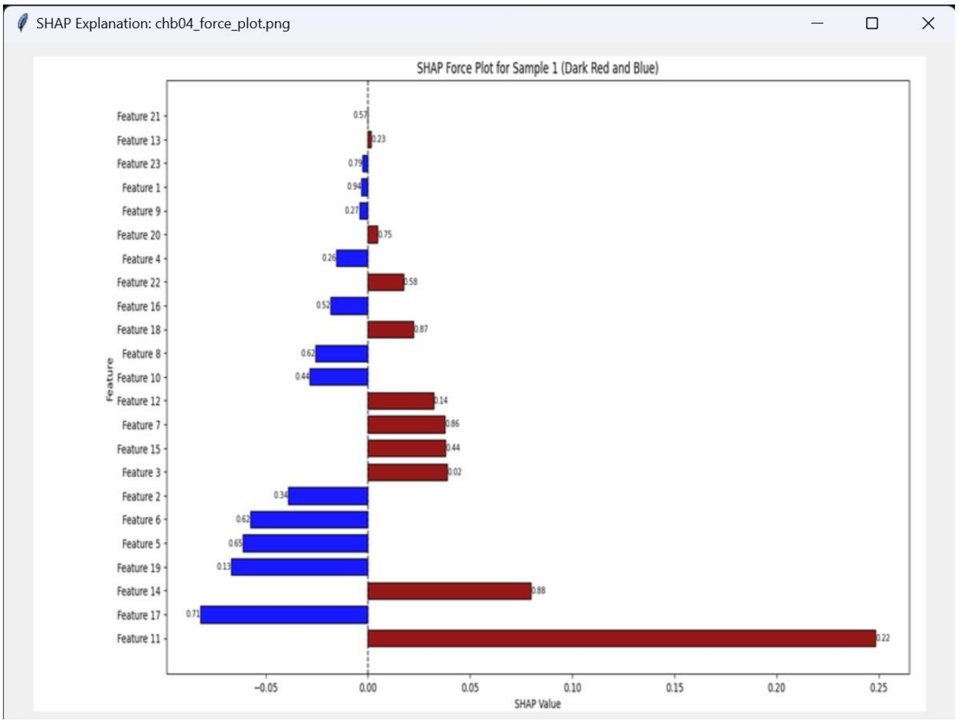
	K	L	M	N	O	P	Q	R	S
1	F4-C4	C4-P4	P4-O2	FP2-F8	F8-T8	T8-P8-0	P8-O2	FZ-CZ	CZ-PZ
2	-45.1282	24.81074	152.967	101.7827	-106.862	-59.9756	180.3175	-18.5592	85.37241
3	0.19536	0.19536	0.19536	0.19536	0.19536	0.19536	0.19536	0.19536	0.19536
4	0.19536	0.19536	0.19536	0.19536	0.19536	0.19536	0.19536	0.19536	0.19536
5	0.586081	0.19536	0.976801	1.758242	-0.9768	0.19536	2.148962	0.586081	0.586081
6	0.19536	0.19536	2.148962	0.586081	0.19536	0.19536	2.148962	0.19536	0.19536
7	-1.75824	-0.58608	-4.49328	-6.83761	5.665446	1.367521	-8.79121	-1.36752	-2.53968
8	-2.14896	-0.19536	-12.3077	-8.79121	7.228327	1.367521	-16.2149	-1.36752	-1.75824
9	0.976801	0.586081	-0.58608	4.102564	-3.32112	-1.36752	2.930403	0.586081	2.539683
10	2.148962	0.19536	18.1685	13.08913	-9.57265	-2.9304	23.63858	1.367521	2.539683
11	-0.9768	-1.36752	3.711844	-3.71184	4.102564	0.976801	-0.19536	-1.36752	-4.10256
12	-6.44689	-2.14896	-29.8901	-29.4994	23.63858	4.102564	-44.7375	-5.27473	-9.57265
13	-17.7778	-6.44689	-49.8168	-59.5849	46.69109	5.665446	-81.8559	-14.652	-20.1221
14	-26.3736	-9.57265	-82.2466	-84.2002	65.44567	8.400488	-126.398	-22.4664	-27.1551
15	-27.1551	-9.18193	-86.5446	-75.6044	59.19414	-0.58608	-120.537	-26.3736	-25.5922
16	-23.6386	-11.917	-64.2735	-58.022	49.03541	-9.96337	-89.2796	-27.5458	-28.3272
17	-19.7314	-16.2149	-55.2869	-55.2869	50.59829	-11.917	-79.5116	-29.1087	-33.0159
18	-15.4335	-16.2149	-73.2601	-65.0549	60.75702	-11.5263	-95.5311	-28.7179	-36.5324
19	-9.57265	-19.3407	-84.591	-74.823	68.18071	-9.57265	-104.908	-27.5458	-38.8767

Visualise CSV file button to visualise the EDF files in csv format

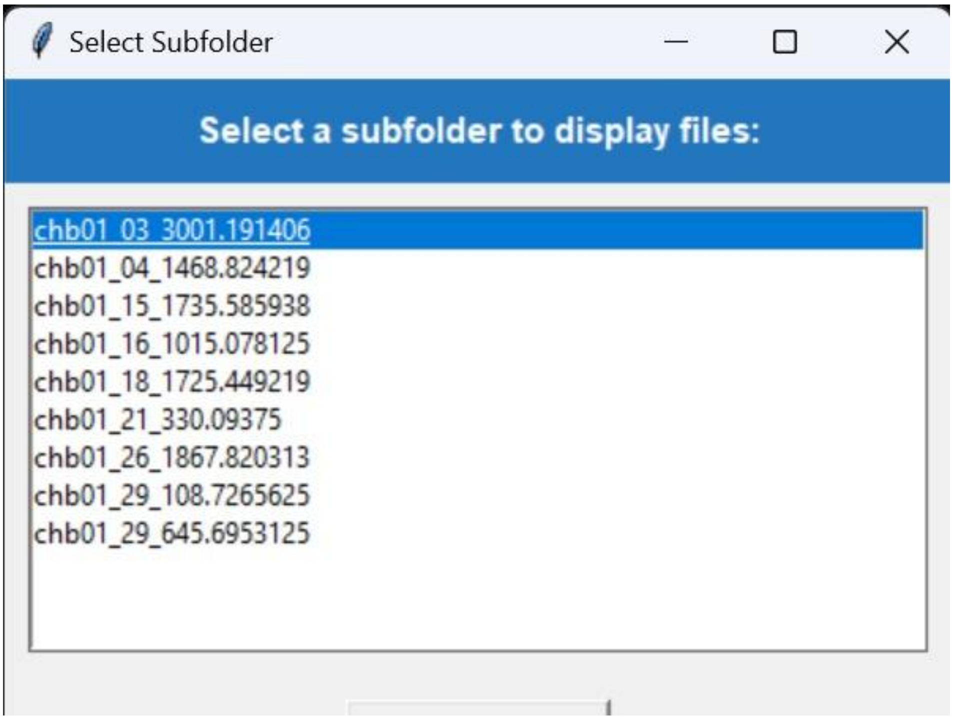


	A	B
1	Start Time	End Time (s)
2	3001.191	3035.289

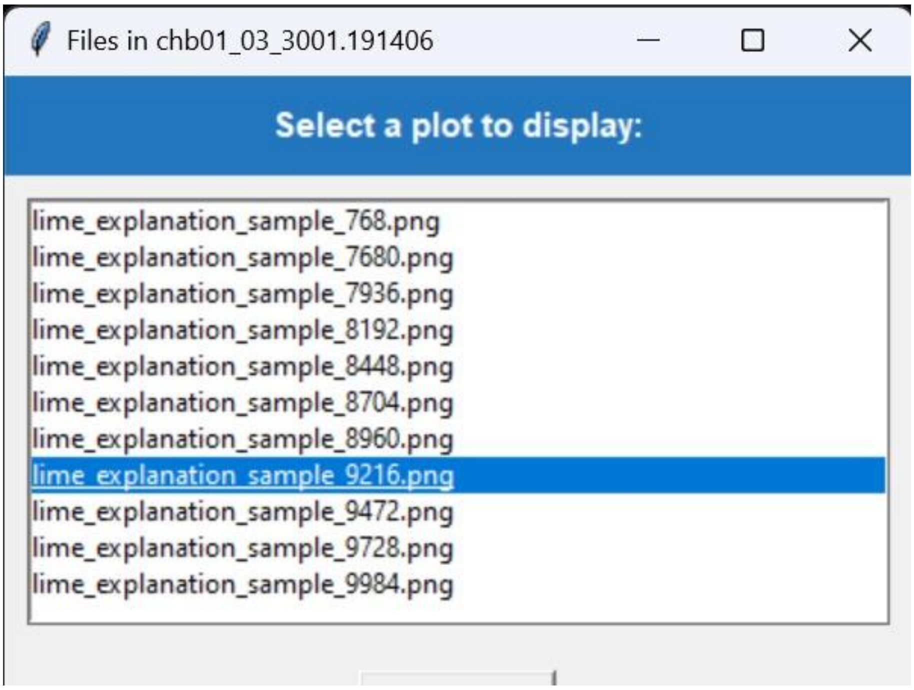
Results of time prediction saved in csv files



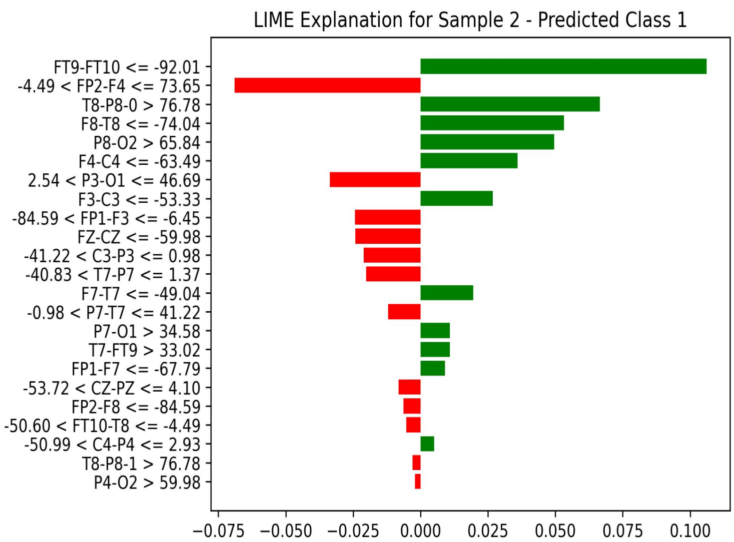
SHAP result for selected seizure of the patient



Selection of subfolder for displaying the LIME result



Selection of sample for displaying the LIME result



LIME results visualised

Results

- The developed 2D CNN model has an accuracy of 0.98
- Alpha Band Slowing Observed
  - Clear reduction in alpha band power during ictal intervals, especially in ictal regions—consistent with known neurophysiological signatures of seizures.
  - Demonstrates model's alignment with clinical observations.
- Statistical Thresholding for Seizure Interval Detection
  - Automates the identification of seizure time windows across patients.
  - Reduces subjectivity and manual tuning; ensures data-driven, patient-specific thresholding.

Results

- SHAP (SHapley Additive exPlanations)
  - Used for global feature attribution, indicating which EEG channels contribute most to seizure prediction.
  - SHAP values show channel-wise importance over time
- LIME (Local Interpretable Model-agnostic Explanations)
  - Provides local, instance-specific explanations by approximating the model with a linear surrogate.
  - Visualizations show which features influenced individual predictions, with class-wise contribution strength.
- Helps clinicians validate whether model reasoning aligns with physiological patterns for a given EEG snippet.

Future Scope

- Integration of Diverse Datasets
  - Extend evaluation across large-scale public EEG datasets like TUH, EPILEPSIAE, and iEEG.org to validate generalizability across demographics and recording setups.
  - Incorporate multimodal data (e.g., ECG, video EEG) for richer seizure context and prediction accuracy.
- Real-Time Seizure Alert System
  - Transform the model into an edge-deployable version for real-time monitoring and seizure alerting.
  - Integrate with wearable EEG devices or hospital systems for continuous patient supervision.

Future Scope

- Extension to Other Neurological Conditions
  - Adapt framework to detect and explain abnormal brain activity in conditions like sleep disorders, stroke, or Alzheimer's disease.
  - Customizable for other signal-based biomedical applications (e.g., EMG for movement disorders).
- Personalized Seizure Forecasting
  - Incorporate temporal patterns, circadian rhythms, and patient-specific biomarkers to forecast seizures rather than just detect them.
  - Potential integration with patient diaries and medication schedules for tailored predictions.

Conclusion

- **Work Completed:** We have successfully completed data preprocessing, feature extraction, and CNN model training. Additionally, we have implemented SHAP and LIME for explainability and also have done work on threshold fixing and GUI.

- **Future scope:** The future scope involves generalising the model on various datasets and include real time localisation if possible.

References

1. M. Xu, P. Wei, S. Yan, X. Gong, and Y. Zhao, "Research Progress on Intelligent Diagnosis of Epilepsy Based on EEG Signals": This paper reviews the advancements in intelligent diagnosis of epilepsy using EEG signals. It focuses on how machine learning and neural network techniques have been applied to improve seizure detection and diagnosis accuracy.
2. G. Shi, T. Wang, S. Xie, et al., "Deep Learning-Based Methods for Seizure Detection Using EEG Signals: A Comprehensive Survey": This comprehensive survey covers various deep learning approaches used for seizure detection via EEG signals. It evaluates the effectiveness of models like CNNs and RNNs in identifying epileptic events.
3. A. L. Rodriguez, M. H. Taha, and J. B. Porras, "EEG Seizure Prediction Based on Enhanced Deep Learning Models and Information Theoretic Approaches": The paper discusses seizure prediction methods using EEG signals, enhanced by deep learning models and information theory techniques to predict seizure onset.

References

4. Schirrmeister, R.T., Gemein, L.A., Eggensperger, K., Hutter, F. and Ball, T., "XAI4EEG: Spectral and Spatio-Temporal Explanation of Deep Learning-Based Seizure Detection in EEG Time Series": This research introduces XAI techniques for explaining how deep learning models detect seizures using spectral and spatio-temporal EEG features, providing greater transparency in model decision-making.
5. X. Alphonse Inbaraj and Jyh-Horng Jeng, "Mask-GradCAM: Object Identification and Localization of Visual Presentation for Deep Convolutional Network": This paper introduces Mask-GradCAM, a technique that enhances object identification and localization in deep convolutional networks, useful for improving interpretability in vision models.
6. Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin, "Why Should I Trust You?": Explaining the Predictions of Any Classifier: This influential paper introduces the LIME (Local Interpretable Model-agnostic Explanations) framework, which provides human-interpretable explanations for the predictions made by black-box machine learning models.

Thank You

Appendix B: Vision, Mission, Programme Outcomes and Course Outcomes

Vision, Mission, Programme Outcomes and Course Outcomes

Institute Vision
To evolve into a premier technological institution, moulding eminent professionals with creative minds, innovative ideas and sound practical skill, and to shape a future where technology works for the enrichment of mankind.

Institute Mission
To impart state-of-the-art knowledge to individuals in various technological disciplines and to inculcate in them a high degree of social consciousness and human values, thereby enabling them to face the challenges of life with courage and conviction.

Department Vision

To become a centre of excellence in Computer Science and Engineering, moulding professionals catering to the research and professional needs of national and international organizations.

Department Mission

To inspire and nurture students, with up-to-date knowledge in Computer Science and Engineering, ethics, team spirit, leadership abilities, innovation and creativity to come out with solutions meeting societal needs.

Programme Outcomes (PO)
Engineering Graduates will be able to:

1. Engineering Knowledge: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.

2. Problem analysis: Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.

3. Design/development of solutions: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.

4. Conduct investigations of complex problems: Use research-based knowledge including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.

5. Modern Tool Usage: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.

6. The engineer and society: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal, and cultural issues and the consequent responsibilities relevant to the professional engineering practice.

7. Environment and sustainability: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.

8. Ethics: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

9. Individual and Team work: Function effectively as an individual, and as a member or leader in teams, and in multidisciplinary settings.

10. Communication: Communicate effectively with the engineering community and with society at large. Be able to comprehend and write effective reports documentation. Make effective presentations, and give and receive clear instructions.

11. Project management and finance: Demonstrate knowledge and understanding of engineering and management principles and apply these to one's own work, as a member and leader in a team. Manage projects in multidisciplinary environments.

12. Life-long learning: Recognize the need for, and have the preparation and ability to engage in independent and lifelong learning in the broadest context of technological change.

Programme Specific Outcomes (PSO)

A graduate of the Computer Science and Engineering Program will demonstrate:

PSO1: Computer Science Specific Skills
The ability to identify, analyze and design solutions for complex engineering problems in multidisciplinary areas by understanding the core principles and concepts of computer science and thereby engage in national grand challenges.

PSO2: Programming and Software Development Skills
The ability to acquire programming efficiency by designing algorithms and applying standard practices in software project development to deliver quality software products meeting the demands of the industry.

PSO3: Professional Skills
The ability to apply the fundamentals of computer science in competitive research and to develop innovative products to meet the societal needs thereby evolving as an eminent researcher and entrepreneur.

Course Outcomes (CO)

Course Outcome 1: Model and solve real world problems by applying knowledge across domains (Cognitive knowledge level: Apply).

Course Outcome 2: Develop products, processes or technologies for sustainable and socially relevant applications (Cognitive knowledge level: Apply).

Course Outcome 3: Function effectively as an individual and as a leader in diverse teams and to comprehend and execute designated tasks (Cognitive knowledge level: Apply).

Course Outcome 4: Plan and execute tasks utilizing available resources within timelines, following ethical and professional norms (Cognitive knowledge level: Apply).

Course Outcome 5: Identify technology/research gaps and propose innovative/creative solutions (Cognitive knowledge level: Analyze).

Course Outcome 6: Organize and communicate technical and scientific findings effectively in written and oral forms (Cognitive knowledge level: Apply).

Appendix C:
CO-PO-PSO Mapping

CO-PO AND CO-PSO MAPPING

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	2	2	2	1	2	2	2	1	1	1	1	2	3		
CO2	2	2	2		1	3	3	1	1		1	1		2	
CO3									3	2	2	1			3
CO4					2			3	2	2	3	2			3
CO5	2	3	3	1	2							1	3		
CO6					2			2	2	3	1	1			3

3/2/1: high/medium/low

JUSTIFICATIONS FOR CO-PO MAPPING

Mapping	Level	Justification
101003/CS822U.1-PO1	M	Knowledge in the area of technology for project development using various tools results in better modeling.
101003/CS822U.1-PO2	M	Knowledge acquired in the selected area of project development can be used to identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions.
101003/CS822U.1-PO3	M	Can use the acquired knowledge in designing solutions to complex problems.
101003/CS822U.1-PO4	M	Can use the acquired knowledge in designing solutions to complex problems.
101003/CS822U.1-PO5	H	Students are able to interpret, improve, and redefine technical aspects for design of experiments, analysis, and interpretation of data, and synthesis of the information to provide valid conclusions.
101003/CS822U.1-PO6	M	Students are able to interpret, improve, and redefine technical aspects by applying contextual knowledge to assess societal, health, and consequential responsibilities relevant to professional engineering practices.
101003/CS822U.1-PO7	M	Project development based on societal and environmental context solution identification is the need for sustainable development.
101003/CS822U.1-PO8	L	Project development should be based on professional ethics and responsibilities.

101003/CS822U.1-PO9	L	Project development using a systematic approach based on well-defined principles will result in teamwork.
101003/CS822U.1-PO10	M	Project brings technological changes in society.
101003/CS822U.1-PO11	H	Acquiring knowledge for project development gathers skills in design, analysis, development, and implementation of algorithms.
101003/CS822U.1-PO12	H	Knowledge for project development contributes engineering skills in computing and information gatherings.
101003/CS822U.2-PO1	H	Knowledge acquired for project development will also include systematic planning, developing, testing, and implementation in computer science solutions in various domains.
101003/CS822U.2-PO2	H	Project design and development using a systematic approach brings knowledge in mathematics and engineering fundamentals.
101003/CS822U.2-PO3	H	Identifying, formulating, and analyzing the project results in a systematic approach.
101003/CS822U.2-PO5	H	Systematic approach is the tip for solving complex problems in various domains.
101003/CS822U.2-PO6	H	Systematic approach in the technical and design aspects provides valid conclusions.
101003/CS822U.2-PO7	H	Systematic approach in the technical and design aspects demonstrates the knowledge of sustainable development.
101003/CS822U.2-PO8	M	Identification and justification of technical aspects of project development demonstrates the need for sustainable development.
101003/CS822U.2-PO9	H	Apply professional ethics and responsibilities in engineering practice of development.
101003/CS822U.2-PO11	H	Systematic approach also includes effective reporting and documentation, which gives clear instructions.

101003/CS822U.2-PO12	M	Project development using a systematic approach based on well-defined principles will result in better teamwork.
101003/CS822U.3-PO9	H	Project development as a team brings the ability to engage in independent and lifelong learning.
101003/CS822U.3-PO10	H	Identification, formulation, and justification in technical aspects will be based on acquiring skills in design and development of algorithms.
101003/CS822U.3-PO11	H	Identification, formulation, and justification in technical aspects provides the betterment of life in various domains.
101003/CS822U.3-PO12	H	Students are able to interpret, improve, and redefine technical aspects with mathematics, science, and engineering fundamentals for the solutions of complex problems.
101003/CS822U.4-PO5	H	Students are able to interpret, improve, and redefine technical aspects with identification, formulation, and analysis of complex problems.
101003/CS822U.4-PO8	H	Students are able to interpret, improve, and redefine technical aspects to meet the specified needs with appropriate consideration for public health and safety, and the cultural, societal, and environmental considerations.
101003/CS822U.4-PO9	H	Students are able to interpret, improve, and redefine technical aspects for design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
101003/CS822U.4-PO10	H	Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools for better products.
101003/CS822U.4-PO11	M	Students are able to interpret, improve, and redefine technical aspects by applying contextual knowledge to assess societal, health, and consequential responsibilities relevant to professional engineering practices.

101003/CS822U.4-PO12	H	Students are able to interpret, improve, and redefine technical aspects for demonstrating the knowledge of, and need for sustainable development.
101003/CS822U.5-PO1	H	Students are able to interpret, improve, and redefine technical aspects, apply ethical principles, and commit to professional ethics and responsibilities and norms of the engineering practice.
101003/CS822U.5-PO2	M	Students are able to interpret, improve, and redefine technical aspects, communicate effectively on complex engineering activities with the engineering community and society at large, such as being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.
101003/CS822U.5-PO3	H	Students are able to interpret, improve, and redefine technical aspects to demonstrate knowledge and understanding of the engineering and management principle in multidisciplinary environments.
101003/CS822U.5-PO4	H	Students are able to interpret, improve, and redefine technical aspects, recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.
101003/CS822U.5-PO5	M	Students are able to interpret, improve, and redefine technical aspects in acquiring skills to design, analyze, and develop algorithms and implement those using high-level programming languages.
101003/CS822U.5-PO12	M	Students are able to interpret, improve, and redefine technical aspects and contribute their engineering skills in computing and information engineering domains like network design and administration, database design, and knowledge engineering.

101003/CS822U.6-PO5	M	Students are able to interpret, improve, and redefine technical aspects and develop strong skills in systematic planning, developing, testing, implementing, and providing IT solutions for different domains, which helps in the betterment of life.
101003/CS822U.6-PO8	H	Students will be able to associate with a team as an effective team player for the development of technical projects by applying the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
101003/CS822U.6-PO9	H	Students will be able to associate with a team as an effective team player to identify, formulate, review research literature, and analyze complex engineering problems.
101003/CS822U.6-PO10	M	Students will be able to associate with a team as an effective team player for designing solutions to complex engineering problems and design system components.
101003/CS822U.6-PO11	M	Students will be able to associate with a team as an effective team player, use research-based knowledge and research methods including design of experiments, analysis, and interpretation of data.
101003/CS822U.6-PO12	H	Students will be able to associate with a team as an effective team player, applying ethical principles and committing to professional ethics and responsibilities and norms of the engineering practice.
101003/CS822U.1-PSO1	H	Students are able to develop Computer Science Specific Skills by modeling and solving problems.
101003/CS822U.2-PSO2	M	Developing products, processes or technologies for sustainable and socially relevant applications can promote Programming and Software Development Skills.
101003/CS822U.3-PSO3	H	Working in a team can result in the effective development of Professional Skills.

101003/CS822U.4-PSO3	H	Planning and scheduling can result in the effective development of Professional Skills.
101003/CS822U.5-PSO1	H	Students are able to develop Computer Science Specific Skills by creating innovative solutions to problems.
101003/CS822U.6-PSO3	H	Organizing and communicating technical and scientific findings can help in the effective development of Professional Skills..