EEG-BASED EPILEPTIC SEIZURE PREDICTION USING ENSEMBLE LEARNING

A Project report submitted in fulfillment of the requirements for the award of the degree of

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE AND ENGINEERING

Submitted by

R. Nikhila (320126510116) B. Gowthami (320126510071)

R. G. S. Sai Vanaja (320126510115) P. Mounika (320126510111)

Under the guidance of

Mrs. G. Gowri Pushpa

Assistant Professor

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

ANIL NEERUKONDA INSTITUTE OF TECHNOLOGY AND SCIENCES
(UGC AUTONOMOUS)
(*Permanently Affiliated to AU, Approved by AICTE and Accredited by NBA & NAAC with 'A' Grade*)
Sangivalasa, Bheemili Mandal, Visakhapatnam dist. (AP)

2020-2024

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

ANIL NEERUKONDA INSTITUTE OF TECHNOLOGY AND SCIENCES
(UGC AUTONOMOUS)
(*Affiliated to AU, Approved by AICTE and Accredited by NBA & NAAC with ‘A’ Grade*)
Sangivalasa, Bheemili Mandal, Visakhapatnam dist. (A.P)

CERTIFICATE

This is to certify that the project report entitled “**EEG-BASED EPILEPTIC SEIZURE PREDICTION USING ENSEMBLE LEARNING”** submitted by **Rokkam Nikhila (320126510116), Bonula Gowthami (320126510071), Reddipalli Gruha Satya Sai Vanaja (320126510115), Pidugu Mounika (320126510111)** in partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology** in **Computer Science and Engineering** of Anil Neerukonda Institute of Technology and Sciences, Visakhapatnam is a record of bonafide work carried out under my guidance and supervision.

Project Guide	**Head of the Department**
Mrs. G. Gowri Pushpa, M.Tech, (Ph.D.)	**Dr. M. Ramakrishna Murthy, M.Tech, Ph.D.**
Assistant Professor	Professor
Department of CSE	Department of CSE
ANITS	ANITS

DECLARATION

We, **R. Nikhila (320126510116), B. Gowthami (320126510071), R. G. S. Sai Vanaja (320126510115), P. Mounika (320126510111)**, of final semester B.Tech., in the department of Computer Science and Engineering from ANITS, Visakhapatnam, hereby declare that the project work entitled is **“EEG-Based Epileptic Seizure Prediction using Ensemble Learning”** carried out by us and submitted in partial fulfillment of the requirements for the award of **Bachelor of Technology in Computer Science and Engineering**, under Anil Neerukonda Institute of Technology & Sciences during the academic year 2020-2024 and has not been submitted to any other university for the award of any kind of degree.

R. Nikhila	320126510116
B. Gowthami	320126510071
R. G. S. Sai Vanaja	320126510115
P. Mounika	320126510111

ACKNOWLEDGEMENT

We would like to express our deep gratitude to our project guide **Mrs. G. Gowri Pushpa,** Department of Computer Science and Engineering, ANITS, for her guidance with unsurpassed knowledge and immense encouragement. We are grateful to **Dr. M. Ramakrishna**, Head of the Department, Computer Science and Engineering, for providing us with the required facilities for the completion of the project work.

We are very much thankful to the **Principal and Management, ANITS, Sangivalasa,** for their encouragement and cooperation to carry out this work.

We express our thanks to the Project Coordinator **Mrs. G. Pranitha**, for her continuous support and encouragement. We thank all the **teaching faculty** of the Department of CSE, whose suggestions during reviews helped us in accomplishment of our project.

We would like to thank our parents, friends, and classmates for their encouragement throughout our project period. At last, but not the least, we thank everyone for supporting us directly or indirectly in completing this project successfully.

PROJECT STUDENTS

R. Nikhila (320126510116)

B. Gowthami (320126510071)

R. G. S. Sai Vanaja (320126510115)

P. Mounika (320126510111)

ABSTRACT

Epilepsy, a pervasive neurological disorder marked by unpredictable seizures, poses profound challenges for affected individuals, impacting their daily lives and overall well-being. Identifying seizures ahead of time is paramount for individuals coping with epilepsy to maintain a sense of control over their health. Through our project, we're delving into the realm of cutting-edge technology to enhance seizure prediction methodologies. By fine-tuning these approaches, our objective is to furnish more precise and dependable forecasts, thereby empowering swift responses that mitigate the disruptive repercussions of seizures. Our endeavor is driven by the overarching goal of improving the quality of life for those impacted by epilepsy, ensuring they have the tools and support necessary to navigate their condition with confidence and resilience. This project capitalizes on the intricacies of scalp EEG signals, integrating deep learning models, such as Convolutional Neural Networks (CNNs) and Long Short-Term Memory (LSTM), can automatically learn and extract features from EEG signals through layers of convolution and recurrent operations and ensemble learning methods for classification to create a sophisticated predictive model. Unlike traditional approaches, our system seeks to transcend the limitations of individual models by combining the strengths of diverse algorithms, including SVM, Random Forests, and XGBoost, to form a robust ensemble. Through rigorous training, validation and testing the developed model seeks to enhance the accuracy, specificity and sensitivity of seizure prediction. In practical terms, our innovative seizure prediction system has the potential to revolutionize epilepsy management for medical professionals. This proactive tool enhances patient safety, empowers medical professionals, and fosters improved outcomes and overall well-being in the challenging landscape of epilepsy.

Keywords: Epileptic Seizures, Scalp EEG Signals, Deep Learning, Ensemble Learning

Table of Contents

Chapter 8 Dataset Details

Chapter 9 Codes

Chapter 10 Experimental Analysis and Results

Chapter 11 Installation

Chapter 12 Conclusion and Future Scope

LIST OF FIGURES

List of ABBREVIATIONS

EEG	Electroencephalogram
CNN	Convolutional Neural Network
LSTM	Long Short-Term Memory
SVM	Support Vector Machine
RF	Random Forest
XGBoost	Extreme Gradient Boost
CHB-MIT	Children's Hospital Boston Massachusetts Institute of Technology
PSD	Power Spectral Density
ROC	Receiver Operating Characteristic
AUC	Area Under the Curve
RAM	Random Access Memory
EDF	European Data Format
UML	Unified Modeling Language
IDE	Integrated Development Environment
MATLAB	MATrix LABoratory
CNTK	Microsoft Cognitive Toolkit
AP	Average Precision

Chapter 1 Introduction

1.1 Introduction

Epilepsy, a neurological disorder affecting over 65 million people, is characterized by unpredictable seizures, presenting significant challenges to individuals' daily lives. The abrupt and unpredictable onset of these seizures can seriously upset a person's daily life. They can limit opportunities for typical activities, such as employment and driving, which can reduce a person's overall quality of life. The unpredictable nature of these seizures is a concern for physical safety, but it also has a significant impact on families, carers, and wider society as a whole. It impacts the ease and wellbeing of individuals with epilepsy, families, carers and the healthcare system, costs society and the economy.

To this end, the objective of our project stands as follows: to develop a state-of-the-art EEG-based epileptic seizure prediction system, which provides predictive seizure forecasting in advance. This is essential in order to provide a critical period for possible intervention. By resorting to optimal integration of deep learning and ensemble learning, the efficiency, precision and effectiveness of the seizure forecasting model can be significantly enhanced, thereby considerably improving the lives of the epileptic population.

Empowering individuals with epilepsy to anticipate and prepare for impending seizures is at the core of our research mission. This anticipatory approach addresses the stages of epilepsy: Interictal, referring to the period between seizures as the baseline state; Ictal, signifying the occurrence of a seizure; Postictal, the recovery phase following a seizure; and Preictal, indicating the period leading up to a seizure. Our approach incorporates CNNs for spatial pattern extraction and LSTM architectures for capturing temporal dependencies in EEG signals. This synergistic use of CNNs and LSTMs enhances the system's ability to autonomously extract both spatial and temporal features from EEG data, contributing to a nuanced understanding of the underlying neural patterns.

Furthermore, our approach integrates ensemble learning, a powerful paradigm combining the strengths of multiple algorithms. The ensemble comprises three robust classifiers—Random Forest, Support Vector Machine, and XGBoost—each contributing distinct strengths to the predictive system. Random Forest, recognized for its flexibility and rapid convergence, enhances overall performance. SVM, a robust classifier, plays a pivotal role in effectively classifying EEG data into preictal and interictal states, thereby improving accuracy and reliability. Additionally, XGBoost, a sequential ensemble learning method, introduces efficient parallel processing, ensuring faster computation and further augmenting the predictive capabilities of our refined system. This comprehensive approach, from advanced feature extraction to ensemble classification, positions our system at the forefront of innovation in the field of epileptic seizure prediction. This project extends beyond the development of a technological solution; it addresses a pressing societal need by contributing to early seizure prediction, patient safety, and better healthcare outcomes.

1.2 Motivation of work

Epilepsy is a serious neurological disorder that affects millions of people worldwide. It causes unpredictable seizures that can disrupt daily life for both the person with epilepsy and their loved ones. We're dedicated to creating a system that can predict when these seizures might happen using EEG technology.

Our main goal is to make life easier for people with epilepsy. We want to give people with epilepsy and their carers an opportunity to be prepared for a seizure. They may be able to make lifestyle adjustments, take medication, or ask for help ahead of time. We are hopeful that such preparations can lead to a higher quality of life for them.

We believe that we also have a responsibility to society at large to address the wider impact of epilepsy. The condition can affect more than just a person's health by also impacting their employment, and social interactions. A reliable prediction method will help to cut healthcare burdens, contribute to creating a safer working environment, and in many ways make society and the culture of inclusivity more welcoming to everyone with epilepsy.

Our ultimate goal is to help predict seizures in people diagnosed with epilepsy. By doing so, we can predict when a seizure might happen, and those who are predicted to have a seizure can take proper preparation. We believe that our work can increase the self-confidence and freedom of individuals living with epilepsy.

1.3 Problem Statement

The absence of a reliable and advanced epileptic seizure prediction system creates a pressing challenge for individuals with epilepsy, impacting their safety and overall quality of life. The unpredictable nature of seizures, coupled with the limitations of existing prediction methods, highlights the critical need for an innovative solution. This project aims to bridge this gap by leveraging deep learning techniques, including CNNs and LSTM architectures, in conjunction with ensemble learning methods. The objective is to enhance the accuracy and reliability of seizure prediction, providing a crucial window for timely intervention. The integration of diverse classifiers, such as Support Vector Machine, Random Forest, and XGBoost, promises a comprehensive approach to address the complex dynamics of epileptic seizures. By developing an advanced EEG-based system, this project seeks to empower individuals, caregivers, and healthcare professionals, ultimately improving the management of epilepsy and contributing to enhanced patient safety and well-being.

Chapter 2 Literature Survey

2.1 Introduction

Over 65 million individuals worldwide grapple with epilepsy, a neurological condition characterized by recurrent seizures stemming from the brain's aberrant firing of neurons in a hyper synchronized manner within the cerebral cortex. Epilepsy is one of the most common neural disorders in the world, with unpredictable seizures that can interfere with activities such as driving and working. These seizures often add significant burdens and stress on the patients and their families, manifesting in depression and other health issues. The social impact is also quite large, with caregivers also facing distress and the healthcare system across the board bearing the cost of care. Our project aims to develop an advanced system that leverages EEG data to predict epileptic seizures and provide timely intervention as well as an overall improvement in the life of individuals who suffer from epilepsy. EEG is an established tool in neurology that is used to study and diagnose a variety of brain disorders including epilepsy. Using the recording of the voltage oscillations resulting from the summed field potentials of brain neuronal activity, the EEG produces data that can be analyzed to diagnose and monitor neural disorders. At the same time, the process of EEG reading analysis can be slow and cumbersome and in the hands of a neurologist can be quite time-consuming.

To tackle this problem, we present a solution that utilizes both deep learning and ensemble learning methods to predict seizures with high accuracy and systematic reliability. Our tool aims to contribute to the predictive capacity of early intervention, without which adverse consequences of seizures may continue to impair the quality of life of individuals with epilepsy.

This literature survey presents an overview of recent survey papers of Epileptic Seizure Prediction based on EEG using deep learning and machine learning.

2.2 Prediction of Epileptic Seizures

" A Generalizable Model for Seizure Prediction Based on Deep Learning using CNN-LSTM Architecture " by Mohamad Shahbazi, Hamid Aghajan (2018)

This paper presents a unique method for predicting epileptic seizures using deep learning on EEG information. Their model, which combines CNNs and LSTMs, surpasses earlier approaches by taking into account both frequency and temporal variables, with a sensitivity of 98.21% and a low false prediction rate of 0.13/h on the CHB-MIT dataset. Preprocessing consists of selecting segments and converting them to 2D images using STFT. Each patient's information is used to train personalized models. The CNN-LSTM architecture detects spectral and temporal patterns, and post-processing minimizes incorrect predictions. This study represents a major breakthrough in seizure prediction accuracy, bringing hope for better patient care.

“A Novel Multi-Scale Dilated 3D CNN for Epileptic Seizure Predictions” by Ziyu Wang, Jie Yang and Mohamad Sawan Wang (2021)

Wang, Yang, and Sawan from Westlake University present a unique CNN model for precise epileptic seizure prediction, which is critical for patient safety. Their methodology uses multi-scale dilated convolution to evaluate EEG signals in time, frequency, and channel dimensions. The model captures more complete characteristics using three-dimensional (3D) kernels, resulting in flexibility in receptive fields. When evaluated on the CHB-MIT EEG database, the model outperforms previous approaches by 80.5% accuracy, 85.8% sensitivity, and 75.1% specificity. The suggested approach overcomes the limits of existing machine learning approaches by automating feature extraction from raw EEG data, allowing for real-time applications. The study proves the effectiveness of their strategy through rigorous examination and comparison with other state-of-the-art models, obtaining a decrease in words by 57.6% while preserving important information. Overall, the proposed multi-scale dilated CNN model shows promise for enhancing epileptic seizure prediction and patient safety.

“Ensemble Classification for Epileptic Seizure Prediction” by N. Saranya, Dr. D. Karthika Renuka, R. Geetha Rajakumari (2021)

The research discusses a technique for predicting epileptic fits by blending electroencephalogram (EEG) signals and machine learning algorithms. The model uses Random Forest and Back Propagation Neural Network (BPNN) as classification techniques to effectively detect seizure onset. Dynamic range and functioning of EEG signal are catered for by Finite Impulse Response (FIR) filtering. Feature extraction and noise elimination have been noted as concerns, with successful solutions being offered by FIR filters. In the paper, there was used open-source repositories to acquire EEG data while PCA was employed for dimensionality reduction and cross-validation for modeling validation. Random Forest achieves 95% accuracy on this dataset outperforming the rest of models. On the other hand, BPNN does not perform so well in comparison. This technological development could be integrated into a web application that would monitor epilepsy patients in real time in order to understand their health status thus enabling prompt interventions whenever necessary. Taken as a whole, this paper emphasizes on how effective machine learning is in seizure prediction as well as its possible role in enhancing healthcare outcomes.

“Epileptic Seizure Prediction using EEG Images” by Felix George, Alex Joseph, Bibin Baby, Alex John, Tonny John, Deepak M, Nithin G, and P.S. Sathidevi (2020)

The research describes an automated approach for classifying EEG data into ictal, non-ictal, and pre-ictal categories in order to anticipate epileptic seizures. The model uses ResNET-50, a convolutional neural network (CNN) architecture, to convert 1D EEG input into 2D EEG images for categorization. This unique technique predicts seizures with an accuracy of 94.98%, proving the usefulness of deep residual networks in processing EEG data. The work solves previous approaches' drawbacks by providing a generic strategy that is applicable to all patients and capable of recognizing pre-seizure regions. The proposed method, if integrated into wearable devices, could allow for timely therapies for epilepsy patients, improving their quality of life.

“Epileptic Seizures Prediction Based on Unsupervised Learning for Feature Extraction” by Ruyan Wang, Linhai Wang, Peng He, Yaping Cui, Dapeng Wu (2022).

The paper introduces an unsupervised strategy for forecasting epileptic seizures that uses deep convolutional autoencoders (DCAEs) to learn features from EEG signals. Unlike standard supervised techniques, which require labeled data to extract features, DCAEs learn discriminative features directly from EEG data. The proposed method extracts key information from DCAEs using their hierarchical structure, allowing for more accurate classification of preictal and interictal phases. The approach's usefulness is demonstrated by its evaluation on the CHB-MIT dataset, which yields encouraging results with an accuracy rate of 96.17% and a false alarm rate of only 0.015. These high percentages demonstrate the method's potential to improve epilepsy care by enabling timely intervention and individualized treatment options based on accurate seizure prediction.

“Refine EEG Spectrogram Synthesized by Generative Adversarial Network for Improving the Prediction of Epileptic Seizures” by Tian Yu, Boyuan Cui (2023)

The research describes an approach for improving the prediction of epileptic seizures, which is an important part of controlling epilepsy, a common neurological illness that affects a large proportion of the global population. Traditional seizure prediction algorithms encounter issues due to data paucity and imbalance, which reduces their effectiveness. To overcome these challenges, the paper suggests using Generative Adversarial Networks (GANs) for data augmentation, which would enable the development of synthetic EEG data. This synthesized data is then polished using a unique refining technique. By training a classifier on this enhanced dataset and evaluating it on real EEG data, the study shows a significant improvement in seizure prediction performance, with an average 2.1% rise in Area Under the Curve (AUC) score compared to conventional techniques. The approach shows promise for addressing limited information and imbalance challenges in seizure prediction and other healthcare applications. The findings reveal a technique to improve patient outcomes by predicting seizures more accurately.

“Semi-supervised Deep Learning System for Epileptic Seizures Onset Prediction” by Ahmed M. Abdelhameed and Magdy Bayoumi (2018)

The research describes a new semi-supervised deep learning strategy for predicting epileptic seizure start using electroencephalogram (EEG) data. The system seeks to classify brain states as interictal (normal) or preictal (before to seizure) by integrating unsupervised and supervised techniques. It uses a two-dimensional deep convolutional autoencoder to extract discriminative spatial features from multichannel EEG data, as well as a Bidirectional Long Short-Term Memory (Bi-LSTM) recurrent neural network for temporal classification. Transfer learning is used to initialize patient-specific networks, which improves training efficiency. The experimental results show an average sensitivity of 94.6% and a low false prediction rate of 0.04FP/h across many patients, outperforming existing approaches for seizure prediction accuracy.

“Predicting Epileptic Seizures using Ensemble Method” by Prosper Chiemezuo Noble-Nnakenyi, Kehinde Adebola Olatunji, Oluwatoyin Bunmi Abiola (2022)

The study described in this paper presents an ensemble model for forecasting epileptic seizures that employs deep learning techniques. To improve prediction accuracy, the proposed model incorporates long short-term memory (LSTM), convolutional neural network (CNN), and sparse autoencoder (SAE). The study overcomes the limits of current seizure prediction models by utilizing ensemble learning, which capitalizes on the strengths of individual models while reducing their drawbacks.

The proposed methodology's key components include data collecting from EEG repositories, data preprocessing, feature extraction by signal mapping, and model fusion via majority voting. The ensemble model demonstrated good accuracy, sensitivity, and specificity in forecasting epileptic episodes.

The results show that the ensemble model outperformed the individual models, with an average accuracy of 97.4%. This high level of accuracy shows that the proposed approach may be useful in real-world applications for early seizure prediction.

“Epileptic Seizure Prediction: A Semi-Dilated Convolutional Neural Network Architecture” by Ramy Hussein, Soojin Lee, Rabab Ward and

Martin J. McKeown (2021)

Semi-Dilated Convolutional Network (SDCN) is a newly described convolutional neural network architecture aimed at accurately predicting seizures using EEG data. Discriminative features are extracted from EEG scalograms using a new convolutional module called ‘semi-dilated convolution’ which is equipped in this design. A Sigmoid output and fully connected layers are resulted via parallel paths of 3x3 and 5x5 semi-dilated convolution blocks. The SDCN achieved sensitivity scores of 88.45% and 89.52%, respectively, on the American Epilepsy Society and Melbourne University EEG datasets, surpassing current state-of-the-art methods. Cost function used is binary cross-entropy, the optimizer used here is Adam while learning rate was set to be 0.001. In general, The SDCN proved to be superior for seizure prediction task giving a hint that semi dilation convolutions can effectively be employed for feature extraction as well as Classification on EEGs

“Optimizing Seizure Prediction from Reduced Scalp EEG Channels Based on Spectral Features and MAML” by Anibal Romney; Vidya Manian (2021)

The paper proposes a novel technique to seizure prediction in epilepsy by combining Model Agnostic Meta-Learning (MAML) with Deep Neural Networks (DNN) using patient-specific electrode channels. The goal is to reduce the number of EEG scalp electrode channels required for effective computational training of time-series signals. The study uses the CHB-MIT Dataset to optimize and choose the number of channels for each individual, with feature extraction performed using Ensemble Empirical Mode Decomposition (EEMD) and Sequential Feature Selection (SFS). The MAML model has a remarkable average sensitivity and specificity score of 91% and 90%, respectively, across 23 individuals. This method shows promise for real-time seizure prediction with a few EEG scalp electrodes, potentially increasing the quality of life for epileptic sufferers.

Chapter 3 Proposed Methodology

3.1 Introduction

The proposed methodology for the EEG-based epileptic seizure prediction system is designed to comprehensively address the challenges associated with predicting seizures accurately and reliably. The starting point involves thoroughly collecting data from twenty-three patients, who were subjected to various EEG recordings using the international 10-20 System. The dataset was divided into training, validation, and testing sets, which ensured that there was a balance between the normal and seizure activities as well as maintaining the sequence of events. Then, feature scaling is done to normalize feature values and reshape data in order to make them fit with Convolutional Neural Network (CNN) and Long Short-Term Memory (LSTM) models.

Feature extraction forms the core of the methodology mixing both CNN and LSTM paradigms for capturing spatial-temporal patterns in EEG signals. Time domain and frequency domain features are generated including mean, variance, skewness, kurtosis, power spectral density (PSD). These components provide an abundant description of EEG characteristics. These features are concatenated together to form a complete set of model input features.

Model training employs diverse base models, including XGBoost, Support Vector Machine (SVM), and Random Forest (RF). The individual models are integrated into an ensemble classifier, utilizing a Majority Voting Strategy to combine their predictions. The ensemble classifier aims to enhance the robustness and generalization of the predictive model. Finally, an in-depth analysis of prediction results is conducted to evaluate the system's performance and effectiveness in seizure prediction. The system's performance is evaluated using accuracy, sensitivity, and specificity, with a focus on the binary outcomes—positive and negative predictions—to empower individuals with timely and reliable information about potential seizure occurrences.

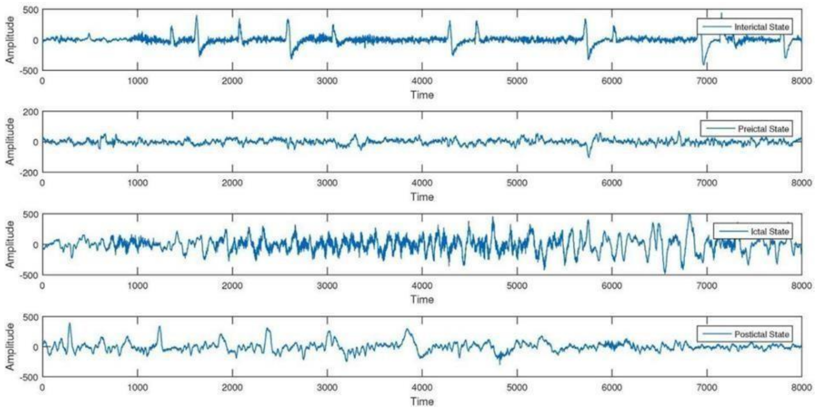


Figure 3.1: Four states of epileptic seizures from a 30s long segment of scalp EEG signal

3.2 Objectives

The objectives of the proposed system are as follows:

- **To collect and preprocess chest EEG data:** Gather a comprehensive dataset of EEG recordings from individuals with epilepsy, ensuring diversity in patient demographics and recording conditions. Identify epileptic periods from individual patients, extract ictal data, and preictal data equally from '.edf files, combine all the data from 24 patients into a single dataset (.csv), and retain data for only the 23 channels which are important for further analysis.

- **To develop a robust deep learning model for predicting seizures:** Design and train deep learning architectures, including Convolutional Neural Networks (CNNs) and Long Short-Term Memory (LSTM) networks, for feature extraction from EEG signals. Integrate diverse classifiers such as XGBoost, Support Vector Machine (SVM), and Random Forest for ensemble learning to enhance prediction robustness.

- **To evaluate the performance of the model:** Assess the predictive performance of the developed model using metrics such as accuracy, sensitivity, and specificity. Conduct thorough cross-validation and testing to ensure the model's reliability and generalization to unseen data.

- **To validate the model in real-world settings:** Conduct thorough cross-validation and testing to ensure the model's reliability and generalization to unseen data. Gather feedback from healthcare professionals and individuals with epilepsy to assess the model's usability and effectiveness in practical applications.

- **To make the model accessible:** Develop user-friendly interfaces or applications to make the seizure prediction model accessible to individuals with epilepsy, caregivers, and healthcare professionals. Provide documentation and support resources to facilitate the implementation and utilization of the model in various healthcare settings.

3.3 System Architecture

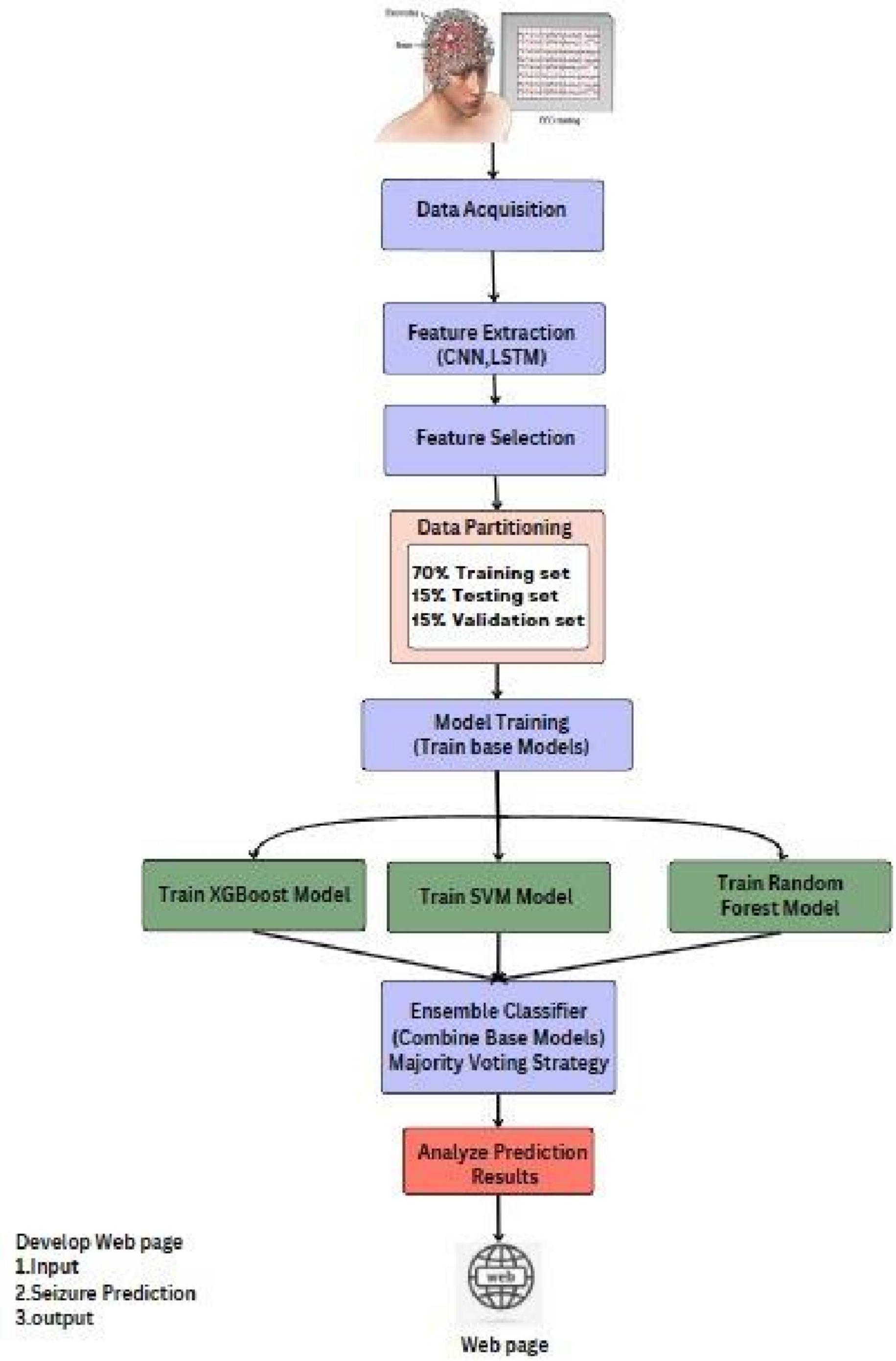


Figure 3.2: Architecture of proposed model

3.3.1 Dataset Acquisition

The EEG data which we used in our work was obtained from the CHB-MIT Database. The CHB-MIT dataset is one of the most widely used openly accessible EEG datasets for seizure identification and prediction. The CHB-MIT dataset is made up of 23 instances organized from continuous scalp EEG recordings of 22 juvenile patients. 23 EEG channels are recorded in most of the files, and all signals are captured at a resolution of 16 bits at 256 samples per second. The dataset contains annotations describing the beginning and conclusion of each seizure. Preictal stage is the term used by the prediction task to describe the temporal interval that precedes each onset.

IEEE Dataport now hosts the pre-processed dataset. In the past, studies only used a small number of the patients' EEG data from the original database. The authors of this research summarized the dataset using 68 entire minutes of epileptic seizure durations from all of the patients, as well as 68 whole minutes of preictal time.

We utilized a subset of the preprocessed CHB-MIT dataset for our EEG-Based Epileptic Seizure Prediction project, specifically selecting 20,000 rows out of the original 2 lakh rows. The preprocessing steps were conducted by Deepa B and Ramesh K (2021), involving the extraction of epileptic periods, filtering redundant electrode data, and labeling ictal and preictal states. The final preprocessed dataset, comprising 23 essential channels, provided the basis for our analysis. Notably, data cleaning and transformation were intentionally omitted to provide flexibility to researchers in selecting appropriate methods for training and testing models. This subset of the preprocessed dataset encompasses data from all 24 patients involved in the study.

The following subsections describe the procedure used to pre-process the CHB-MIT scalp EEG database.

- **Step1:**

  CHB MIT scalp EEG database provides data at physionet in 'edf' European data format. The data is supported with information regarding epileptic periods. The voltage levels from EEG electrodes are obtained from 'edf' files.

- **Step2**:

  Information about the ictal state and equal preictal state is gathered from 'edf' files. Two files are maintained separate so that researchers can use data as needed. This also helps with labeling.

- **Step3:**

  Duplicate and inaccurate electrode data are produced by a thorough analysis of the dataset. The 96 data channels of the EEG are comprised of 23 mandatory channels. We're going to keep these 23 channels.

- **Step4:**

  The preictal and non-preictal state data are labeled with 0 and 1, respectively, in the last stage. The final pre-processed data is obtained by merging the two datasets. Five different files contain the data: two raw files for non-preictal and preictal data, two processed files with 23 important channels according to the 10-20 EEG placement system, and a final file with 136 minutes of combined non-preictal and preictal data with outcomes indicated as '0' for preictal and '1' for non-preictal (Deepa B and Ramesh K, 2021).

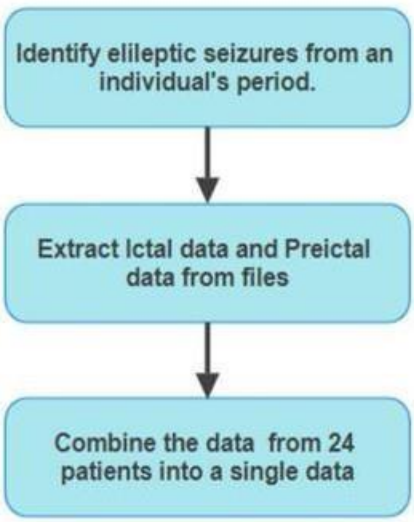


Figure 3.3: Steps in Preprocessing

3.3.2 Feature Extraction using CNN-LSTM Models

Feature extraction is a pivotal step in the data preprocessing pipeline, focusing on deriving informative representations from raw data using a combination of Convolutional Neural Network (CNN) and Long Short-Term Memory (LSTM) models. These models are designed to capture complex spatial and temporal patterns within the electroencephalogram (EEG) signals.

3.3.2.1 CNN Feature Extraction:

- The CNN component specializes in identifying spatial patterns within the

EEG data.

- Convolutional layers with filters detect hierarchical features, recognizing spatial relationships across electrode channels.

- Pooling layers condense information, emphasizing essential spatial characteristics while reducing dimensionality.

3.3.2.2 LSTM Feature Extraction:

- The LSTM component excels at capturing temporal dependencies in EEG signals.

- Long Short-Term Memory units maintain memory of previous states, enabling the model to understand the sequential nature of EEG data.

- LSTM layers learn temporal patterns over extended time intervals.

3.3.2.3 Combining CNN and LSTM:

- Outputs from the CNN and LSTM models are concatenated to create a comprehensive feature set.
- This combined feature set captures both spatial and temporal aspects of EEG signals, providing a rich representation for subsequent classification tasks.
- Feature extraction using CNN-LSTM models enhances the interpretability of EEG data, allowing the model to automatically learn discriminative features. The drawn-out functions work as high-level depictions that can properly separate in between typical and also seizure tasks, mapping out the structure for precise category.

The result is a feature-rich depiction of EEG information including both spatial as well as temporal attributes. This depiction is essential for constructing designs that can determine complex patterns within EEG signals, adding to the performance of seizure discovery and also category.

3.3.3 Feature Selection

Judiciousness belongs to attribute choice in our job to boost our forecast designs' interpretability together with efficiency. We developed a solid attribute choice treatment by making use of the abilities of artificial intelligence strategies especially Random Forest Feature Importance. We assessed each function's importance in our information utilizing an arbitrary woodland classifier. After that relying on their significance positions the SelectFromModel method was made use of to maintain just one of the most helpful attributes. This approach reduced the likelihood of overfitting as well as enhanced design generalization along with lowering the dimensionality of our function area. We ensured that our designs were learnt one of the most essential plus appreciable attributes by meticulously selecting them which eventually boosted the forecasted precision together with general efficiency of the version.

3.3.4 Model Training

In this stage, we teach our computer models to identify patterns in brain activity and anticipate seizures. We're using three types of models: XGBoost, Support Vector Machine (SVM), and Random Forest (RF). These models are like the brain of our prediction system. We'll explain how we're training them and why we chose them.

3.3.4.1 Model Selection

Any predictive system's effectiveness and dependability depend on the careful selection of its underlying models. Here, we explain the reasoning behind our selection of the models—XGBoost, Support Vector Machine (SVM), and Random Forest (RF)—and discuss their features, capabilities, and applicability to our epileptic seizure prediction system.

1. XGBoost

XGBoost, brief for eXtreme Slope Improving attracts attention as a crucial selection for our anticipating structure owing to its exceptional abilities in handling varied datasets along with catching detailed connections. At its significance XGBoost uses a set knowing structure mostly driven by increasing strategies allowing it to build extremely accurate anticipating versions by progressively combining weak students. This hidden formula skillfully lessens loss features while additionally punishing version intricacy, consequently accomplishing an optimum equilibrium in between prejudice along with difference. In addition, XGBoost's intrinsic durability versus overfitting as well as its capability to take care of missing out on information even more improve its appearance. Additionally, its scalability plus performance makes it appropriate for assessing large datasets effortlessly straightening with the demands of our seizure forecast system.

2. Support Vector Machine

Support Vector Machine (SVM) holds a prominent setting in the domain name of monitored knowing popular for its capability to manage high-dimensional attribute areas as well as give durable category. SVM's efficiency comes from its adherence to the concept of architectural danger reduction guaranteeing its capability to generalize successfully to hidden information circumstances. By building an ideal hyperplane that takes full advantage of the margin in between various courses, SVM lusters in determining complex patterns within information also despite sound as well as high dimensionality. Furthermore, SVM's versatility expands to its ability to take care of non-linear connections with bit techniques, allowing it to outline intricate choice limits properly. Within the context of our seizure prediction system, SVM's resilience, scalability, and adeptness in managing high-dimensional feature spaces make it an indispensable tool for precise classification and prediction tasks.

3. Random Forest

Random Forest (RF) arises as a keystone in anticipating modeling jobs using unmatched convenience as well as scalability. Based in set understanding concepts, RF accumulates choice trees to produce a durable anticipating structure. Each choice tree within RF is educated on a part of the information and also their forecasts are combined with ballot or balancing systems. This varied set not just improves anticipating precision yet additionally safeguards versus overfitting and also difference. In addition, RF's natural capacity to deal with non-linear partnerships as well as attribute communications clothing it with premium discriminative power allowing accurate demarcation of intricate choice borders. In addition, RF's parallelizability expedites version training along with reason, straightening easily with real-time forecast demands. Within the domain name of our seizure forecast system RF attracts attention for its strength, interpretability, along with versatility throughout varied modeling circumstances.

In recap each design brings its distinct toughness as well as abilities to the leading jointly boosting the anticipating expertise of our system plus leading the way for boosted person end results plus lifestyle.

3.3.4.2 Model Training Procedure

For each selected base model (XGBoost, SVM, and Random Forest):

- Initialize the model with default parameters or parameters based on prior experimentation.
- Train the model on the training dataset.
- Use appropriate evaluation metrics (e.g., accuracy, F1-score, etc.) to monitor the model's performance during training.
- Repeat the training process for each base model.

3.3.5 Ensemble Classifier

In this phase, we combine the predictions generated by the base models, namely XGBoost, Support Vector Machine (SVM), and Random Forest (RF), using a Majority Voting Strategy to construct an ensemble classifier. Ensemble methods leverage the diversity among individual models to enhance prediction robustness and generalization, thereby improving overall performance.

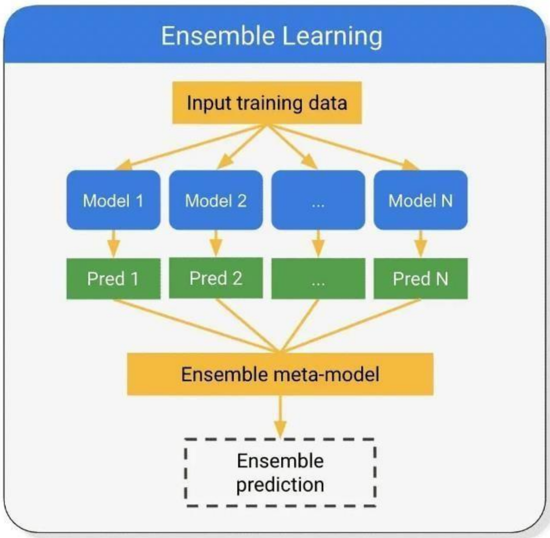


Figure 3.4: How does Ensemble Learning Work?

3.3.5.1 Majority Voting Strategy

The Majority Voting Strategy aggregates the predictions from multiple base models and assigns the final class label based on the majority prediction. The ensemble classifier predicts the class that receives the greatest number of votes, with each base model prediction having equal weight. By reducing the possibility of individual model biases and errors, this method produces forecasts that are more accurate and dependable.

3.3.5.2 Implementation

To apply the set classifier making use of the Majority Voting Method we comply with these actions:

1. **Create Predictions:** Utilize the educated base designs (XGBoost, SVM, RF) to produce forecasts on the recognition or screening information collection.

2. **Integrate Predictions:** Aggregate the forecasts from each base design right into a solitary set forecast matrix where each row matches an information circumstance as well as each column stands for the anticipated course tag from a base version.

3. **Voting Mechanism:** For each information circumstances establish the bulk course tag amongst the forecasts produced by the base versions. Designate this bulk course tag as the last forecast for the equivalent information circumstances in the set classifier's outcome.

4. **Evaluation:** Evaluate the performance of the ensemble classifier using appropriate evaluation metrics such as accuracy, sensitivity, specificity, and area under the ROC curve (AUC). Compare the ensemble classifier's performance against individual base models to assess its effectiveness in enhancing prediction robustness and generalization.

3.3.5.3 Benefits of Ensemble Classification

- **Improved Robustness:** By combining predictions from diverse models, the ensemble classifier mitigates the risk of individual model biases and errors, leading to more robust predictions.

- **Enhanced Generalization:** Ensemble methods exploit the complementary strengths of different models, allowing for better generalization to unseen data instances.

- **Increased Accuracy:** The ensemble classifier leverages the wisdom of crowds, often achieving higher accuracy than any individual base model alone.

3.3.6 Analyze Prediction Results

We analyze the prediction results of our ensemble classifier on both the validation set and testing set. We evaluate various metrics to assess the performance of the model and gain insights into its behavior.

3.3.6.1 Ensemble Classifier Performance

Validation Set:

- **Accuracy:** The ensemble classifier achieves an accuracy of approximately 94.77% on the validation set, indicating that it correctly classifies the majority of instances.
- **Precision and Recall:** The precision and recall for both classes (0 and 1) are high, indicating that the model effectively identifies both non-seizure (0) and seizure (1) instances. The precision and recall values are balanced, contributing to the model's overall effectiveness.
- **F1-Score:** The F1-score, which considers both precision and recall, is

approximately 95% for both classes. This balanced F1-score indicates robust performance across both classes.

- **Confusion Matrix:** The confusion matrix reveals that the model correctly predicts the majority of instances, with a relatively low number of false positives and false negatives.

Testing Set:

- **Accuracy:** The ensemble classifier maintains a high accuracy of approximately 94.73% on the testing set, consistent with its performance on the validation set.

- **Precision and Recall:** Similar to the validation set, the precision and recall values for both classes are high on the testing set, indicating the model's ability to effectively discriminate between seizure and non-seizure instances.

- **F1-Score:** The F1-score remains balanced on the testing set, reflecting the model's consistent performance in terms of precision and recall.

3.3.6.2 Additional Metrics and Insights

- **Specificity:** The specificity of the model is approximately 97.26% on the testing set, indicating its ability to correctly identify non-seizure instances (true negatives) with high accuracy.

- **False Positive Rate (FPR):** The FPR is relatively low, indicating a minimal rate of false alarms or misclassifications of non-seizure instances as seizure instances.

- **False Negative Rate (FNR):** The FNR is also low, indicating a small proportion of seizure instances being missed or misclassified as non-seizure instances.

- **AUC-ROC Score:** The Area Under the Receiver Operating Characteristic (ROC) Curve is approximately 94.68%, indicating good discriminative power and model performance across different threshold values.

- **Average Precision (AP):** The AP score, which measures the average precision-recall tradeoff, is approximately 93.19%, reflecting the model's precision across different recall levels.

Measures	Computation
Accuracy-The number of correct predictions from all predictions made.	$\frac{TP+TN}{TP+TN+FP+FN}$
Sensitivity- True positive rate (TPR) of a test.	$\frac{TP}{TP+FN}$
Specificity- True negative rate (TNR) of a test	$\frac{TN}{TN+FP}$
False positive rate (FPR)	$\frac{FP}{FP+TN}$
False positive rate per hour (FPR/h).	(FPR/h) is calculated in the horizon time in the ictal transition of the epileptic EEG signal
Receiver operating characteristic (ROC).	ROC is a plot of TPR (sensitivity) against FPR (1-specificity)
Where TP =True Positive, TN =True Negative, FN = False Negative, FP = False Positive	

The analysis of prediction results demonstrates the robust performance of the ensemble classifier in predicting epileptic seizures. The model exhibits high accuracy, precision, recall, and balanced F1-scores on both the validation and testing sets. The low false positive and false negative rates further indicate the model's effectiveness in distinguishing between seizure and non-seizure instances.

Chapter 4 Requirement Analysis

4.1 Introduction

Requirement Engineering is a fundamental process that plays a vital role in software development. It includes the recognition meaning, and also administration of the system needs essential to satisfy the customer's assumptions. These needs make up of functions, features plus restrictions that the system should fulfill.

The procedure of Requirement Engineering makes up of 2 vital tasks, specifically demand elicitation and also evaluation. Demand elicitation includes the collection as well as paperwork of the customer's needs utilizing numerous strategies such as studies, meetings, monitoring and also workshops. The primary purpose is to generate a comprehensive plus precise system requirements that the customer can recognize as well as accept.

When the needs have actually been collected the following action includes evaluating as well as refining them to develop an evaluation design that designers can translate plus usage to make as well as carry out the system. This phase requires determining any kind of disparities, unpredictability or spaces in the demands coupled with resolving them via conversations with the customer as well as various other stakeholders.

A need is a declaration that explains the anticipated capability of the suggested system either clearly or unconditionally. Demands can be classified as practical, which explains what the system have to do, or non-functional, which defines exactly how well the system ought to execute.

Requirements can be divided into two major categories:

1. Functional Requirements.
2. Non-Functional Requirements.

4.2 Functional Requirements

Functional requirements are specific actions and behaviors that a software system mustperform to meet users' needs. They define the system's capabilities and features and areexpressed in terms of input, processing, and output. To ensure accuracy, they must be defined clearly and validated. The functional requirements for the proposed system areas follows:

- **Data Partitioning:** Our dataset is characterized by its balanced composition, comprising 10,000 rows for each class under consideration. The dataset is partitioned into distinct subsets: a training set encompassing 70% of the data, and separate validation and testing sets each comprising 15% of the data.

- **Data Preparation:** In data preparation, the CHB MIT scalp EEG database in 'edf' format provides voltage levels from EEG electrodes, supported with information on epileptic periods. Separate files for ictal and preictal states aid in labeling and flexibility for researchers. Only 23 essential channels retained. Final preprocessing involves labeling ictal and preictal states as 1 and 0, respectively.

- **Feature Extraction:** Feature extraction in EEG-based seizure prediction utilizes CNN and LSTM models to capture spatial and temporal patterns. CNNs identify spatial patterns, while LSTMs capture temporal dependencies. Outputs from these models are combined to create a comprehensive feature set. Additionally, time-domain and frequency-domain features like mean, variance, PSD, and relative power are computed.

- **Classification**: The system must be able to accurately predict the onset of seizure into one of the two classes: Preictal and Interictal.

- **Result Visualization**: The system must visualize the classification results in anintuitive and user-friendly manner.

4.3 Non-Functional Requirements

Non-functional requirements are the criteria that define the system's performance, quality, and behavior, rather than its specific functionality. These requirements describethe system's characteristics, such as its reliability, security, usability, performance, scalability, and maintainability, and are essential for ensuring that the system meets the user’s expectations and needs. The non-functional requirements for the proposed systemare as follows:

- **Accuracy:** The model should demonstrate high accuracy in distinguishing between seizures and non-seizures.

- **Performance:** The model should efficiently and quickly classify the seizures.

- **Robustness:** The system should demonstrate resilience to variations in EEG signals and environmental factors, ensuring reliable seizure prediction across diverse conditions.

- **Scalability:** The system should be scalable to accommodate larger datasets and potential integration with real-time monitoring systems, ensuring its applicability in broader clinical settings and future expansions.

4.4 Technical Requirements

The technical requirements for this project are mentioned below:

1. Hardware Requirements
2. Software Requirements

4.4.1 Hardware Requirements

- **Processor:** The processor needs to be fast enough to handle the training of a deeplearning model with a large dataset. A high-end processor, such as Intel Core i7or i9 or an equivalent AMD processor, is recommended.

- **Graphics Processing Unit (GPU):** A powerful GPU with high memory capacityis required to accelerate the training process of deep learning models. NVIDIA GPUs are commonly used for deep learning tasks.

- **RAM:** Deep learning models require a significant amount of memory to hold theweights and biases of the model during training. At least 8 GB of RAM is recommended.

- **Storage:** The dataset and model checkpoints can take up a large amount of diskspace. It is recommended to have at least 500 GB of storage available.

4.4.2 Software Requirements

- **Python Programming Language:** Python is an interpreted, high-level, general-purpose programming language. It is widely used in data science, machine learning, and artificial intelligence. It is necessary for the implementation of theproject.

- **TensorFlow or Keras Deep Learning Libraries:** TensorFlow and Keras are two popular deep learning libraries. They provide a high-level API for buildingand training deep learning models.

- **Scikit-Learn Library:** Scikit-Learn is a popular machine learning library. It provides a wide range of machine learning algorithms and tools. Stratified Shuffle Split method is a cross-validation method that is used for evaluation of machine learning models. Scikit-Learn library is necessary for the implementation of this method and also for accuracy calculation metrics.

- **MNE Library:** MNE-Python is an open-source Python module for processing, analysis, and visualization of functional neuroimaging data (EEG, MEG, sEEG, ECoG, and fNIRS).

- **Numpy and Pandas Libraries:** Numpy and Pandas are popular libraries for datamanipulation. They provide tools for handling arrays, matrices, and dataframes.They are necessary for the implementation of the project.

- **SciPy:** SciPy provides algorithms for optimization, integration, interpolation, eigenvalue problems, algebraic equations, differential equations, statistics and many other classes of problems. The algorithms and data structures provided by SciPy are broadly applicable across domains.

- **Matplotlib:** Matplotlib is a Python plotting library used to create static, animated,and interactive visualizations in Python. It can be used for a variety of tasks such as creating line plots, scatter plots, bar plots, histograms, and more. Matplotlib provides a wide range of customization options, including the ability to add titles,labels, and annotations to your plots.

Chapter 5 Modules Division

5.1 Feature Extraction and Selection

In this phase, feature extraction is achieved through the utilization of CNN-LSTM architectures, capable of capturing intricate patterns and dependencies within the data. These models excel in extracting features, providing a comprehensive representation of the dataset. It's imperative to ensure efficient storage of these extracted features to facilitate subsequent modeling processes effectively. Furthermore, if deemed necessary, feature selection techniques are applied to discern the most pertinent features, optimizing the dataset for improved model performance. Evaluating the impact of feature selection on model performance offers critical insights into the efficacy of the chosen features, guiding further iterations and refinements in the modeling approach.

5.2 Seizure Prediction

One of the main tasks in this module division is to classify EEG results as either suggestive of a seizure or not. Three base machine learning models, Random Forest, Support Vector Machine, and XGBoost are used to accomplish this. By combining these models, an ensemble model is produced that improves the robustness and accuracy of the seizure detection procedure. Through the analysis and categorization of the EEG results, each base model aids in the decision-making process. In order to run the ensemble model and provide predictions based on fresh EEG readings obtained from the EEG Data Input Module, the module uses an inference engine. By ensuring a thorough and dependable classification of seizures, this ensemble approach improves the efficacy of the seizure detection system.

5.3 Model Evaluation

Assessing the performance of the ensemble classifier using various metrics such as accuracy, precision, recall, F1-score, and area under the ROC curve.

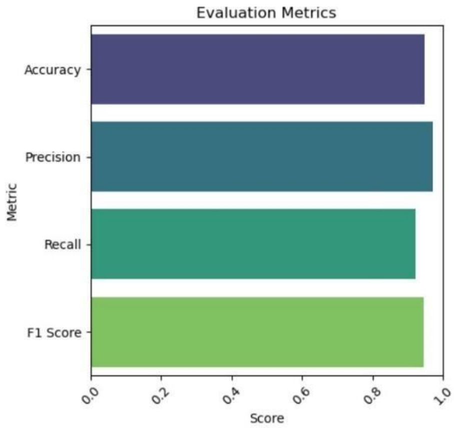


Figure 5.1: Model Evaluation Metrics

5.4 Output

This module displays the model's result, which is a prediction of an upcoming epileptic episode. It displays the findings of the seizure detection process to the user in text format. The categorization is binary, with two possible outcomes: "Preictal" indicating that the patient is expected to have a seizure and "Non preictal" indicating that no seizure is anticipated. The output shows these categorization findings on the screen, indicating whether a seizure is expected or not. Furthermore, the model ensures that the output is continuously updated in real time as the system processes and classifies fresh EEG values, allowing for rapid and responsive seizure risk monitoring. This model can be utilized by healthcare professionals to diagnose and treat patients accordingly.

Chapter 6 Techniques

6.1 Random Forest Algorithm

Random Forest is a versatile and powerful ensemble learning method widely used for both classification and regression tasks in machine learning. It operates by constructing multiple decision trees during training and outputting the mode of the classes for classification tasks or the mean prediction for regression tasks. It consists of a collection of decision trees, each trained on a random subset of the training data and a random subset of the features. This randomness and diversity among the trees contribute to the robustness and effectiveness of the Random Forest algorithm.

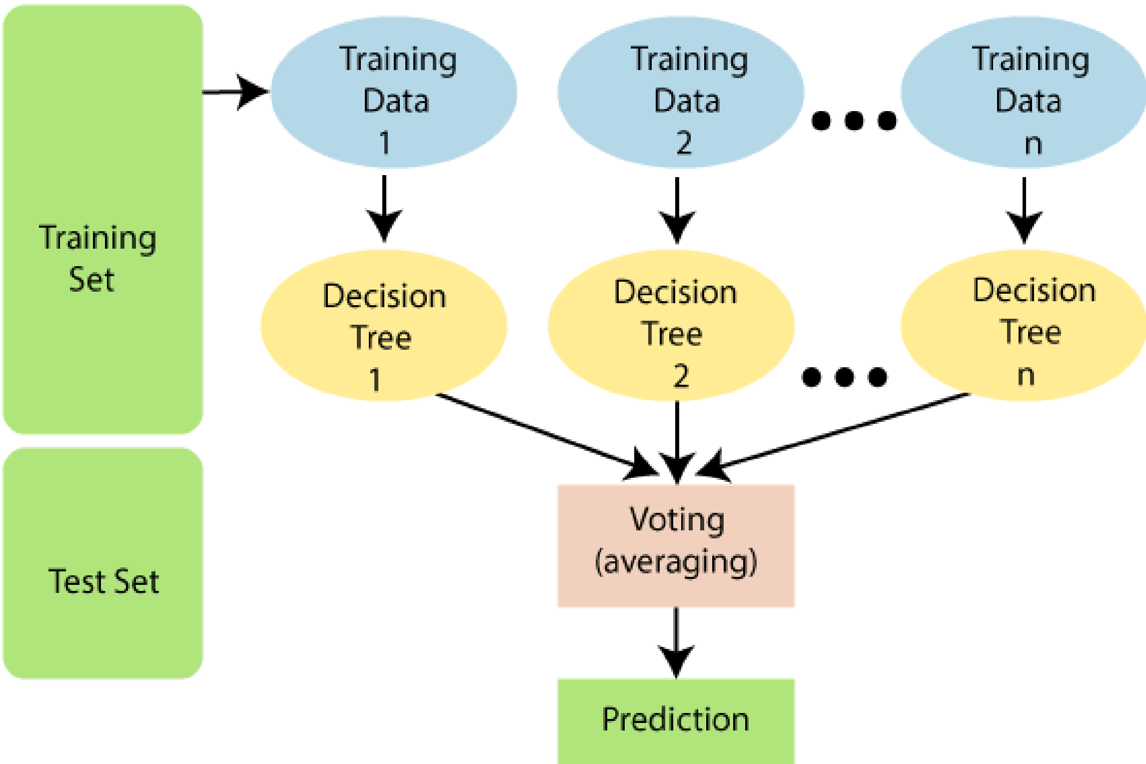


Figure 6.1: Random Forest

Random Forests have several key components:

- **Decision Trees:** The basic building blocks of Random Forests are decision trees, which make binary decisions at each node based on feature values.

- **Voting Mechanism**: In classification tasks, Random Forest combines the predictions of multiple decision trees through a voting mechanism, where the class with the most votes becomes the final prediction. In regression tasks, it averages the predictions of individual trees.

- **Random Feature Selection:** At each node of the decision tree, Random Forest selects a random subset of features to consider for splitting, which helps to decorrelate the trees and improve generalization.

6.1.1 Importance of Random Forest Algorithm

Random Forests offer several advantages that make them important in various machine learning tasks:

- **High Accuracy:** Random Forests typically provide high accuracy in both classification and regression tasks, making them suitable for a wide range of applications.

- **Robustness to Overfitting**: The ensemble nature of Random Forests helps to mitigate overfitting, making them less sensitive to noise and outliers in the data compared to individual decision trees.

- **Feature Importance**: Random Forests can provide insights into feature importance, helping users understand which features are most relevant for prediction.

6.1.2 Limitations of Random Forest Algorithm

Despite their many advantages, Random Forests also have limitations that should be considered:

- **Interpretability:** Random Forests can be challenging to interpret, especially when dealing with a large number of trees and features. Understanding how individual trees contribute to the overall prediction can be complex.

- **Computational Complexity:** Training a Random Forest model can be computationally expensive, especially with a large number of trees and features. Additionally, hyperparameter tuning may require extensive computational resources.

- **Biased Toward Majority Classes:** In classification tasks with imbalanced class distributions, Random Forests may be biased toward the majority classes, leading to suboptimal performance for minority classes, organizations or individuals with limited resources.

6.2 Support Vector Machine

Support Vector Machine (SVM) is a powerful supervised learning algorithm used for classification, regression, and outlier detection tasks. It works by finding the hyperplane that best separates the classes in the feature space while maximizing the margin between the classes. SVMs are particularly effective in high-dimensional spaces and are widely used in applications such as image classification, text classification, and bioinformatics.

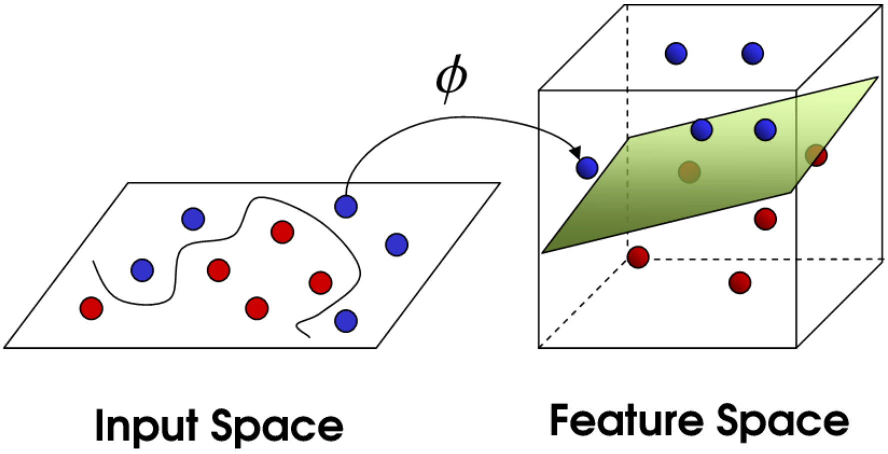


Figure 6.2: Kernel Trick for Support Vector Machine

SVM have several key components:

- **Hyperplane**: In SVM, the hyperplane is the decision boundary that separates the classes in the feature space. For binary classification, the hyperplane is defined as the line that maximizes the margin between the closest data points of different classes, known as support vectors.

- **Kernel Trick**: SVM can efficiently handle nonlinear classification tasks by mapping the input features into a higher-dimensional space using a kernel function. This allows SVM to find a linear decision boundary in the transformed feature space, even if the original feature space is nonlinear.

- **Support Vectors**: Support vectors are the data points that lie closest to the decision boundary and determine the position of the hyperplane. These are the critical points that influence the margin and the decision boundary.

6.2.1 Importance of Support Vector Machine

Support Vector Machines offer several advantages that make them important in various machine learning tasks:

- **Effective in High-Dimensional Spaces**: SVMs perform well in high-dimensional spaces, making them suitable for tasks with a large number of features, such as text classification and image recognition.

- **Robustness to Overfitting**: SVMs are less prone to overfitting compared to other classifiers, such as decision trees, especially when using a proper regularization parameter.

- **Versatility:** SVMs can be used for both linear and nonlinear classification tasks by choosing an appropriate kernel function. This flexibility allows SVMs to handle a wide range of data types and distributions.

6.2.2 Limitations of Support Vector Machine

Despite their advantages, Support Vector Machines also have limitations that should be considered:

- **Sensitivity to Parameter Tuning**: SVM performance is sensitive to the choice of hyperparameters, such as the regularization parameter (C) and the choice of kernel function. Proper parameter tuning is essential for achieving optimal performance.

- **Limited Interpretability in Nonlinear Cases**: SVMs with nonlinear kernels can produce complex decision boundaries that are difficult to interpret, especially in high-dimensional feature spaces. Understanding the model's behavior may require additional techniques, such as feature importance analysis.

- **Difficulty Handling Large Datasets**: SVM training time increases significantly with the size of the dataset, making them less suitable for very large datasets compared to other algorithms such as gradient boosting machines.

6.3 XGBoost Algorithm

XGBoost (Extreme Gradient Boosting) is a scalable and efficient machine learning algorithm used for supervised learning tasks, including classification, regression, and ranking problems. It belongs to the family of gradient boosting algorithms and has gained widespread popularity for its performance and flexibility in various machine learning competitions and real-world applications. XGBoost builds a predictive model by combining the predictions of multiple weak learners, typically decision trees, in an additive manner. It iteratively improves the model's performance by minimizing a loss function and adding new trees that complement the existing ones.

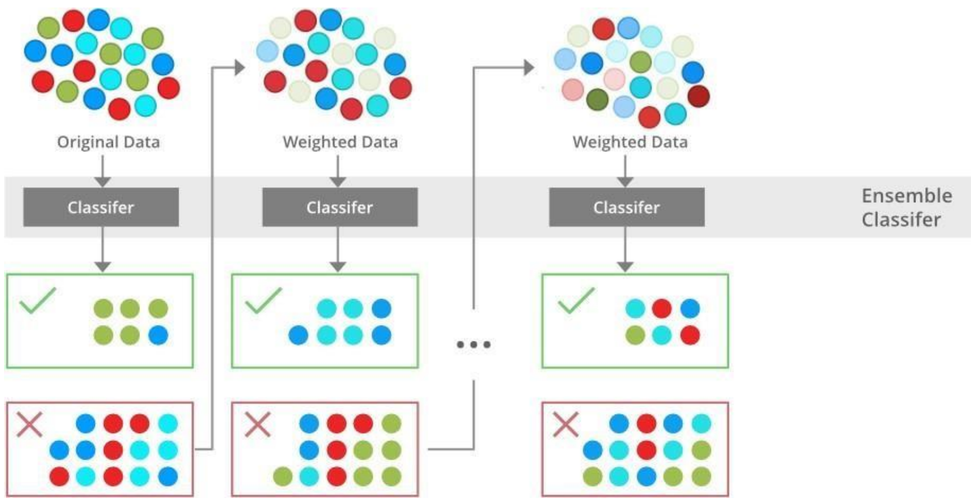


Figure 6.3: Gradient Boosting process used in XGBoost

Key components of XGBoost include:

- **Weak Learners**: XGBoost uses decision trees as weak learners by default, although it can also incorporate other types of base learners. Each tree is trained to predict the residuals (errors) of the previous trees, allowing XGBoost to correct mistakes made by earlier trees.

- **Gradient Boosting**: XGBoost employs a gradient boosting framework, where each new tree is trained to minimize the gradient of the loss function with respect to the model's predictions. This approach enables XGBoost to effectively handle complex nonlinear relationships in the data.

- **Regularization**: XGBoost includes several regularization techniques to prevent overfitting and improve generalization performance. These techniques include shrinkage (learning rate), tree depth regularization, and feature subsampling.

6.3.1 Importance of XGBoost Algorithm

XGBoost offers several advantages that make it an important tool in machine learning:

- **High Performance**: XGBoost consistently achieves state-of-the-art performance on a wide range of machine learning tasks, including classification, regression, and ranking. Its ensemble approach and regularization techniques help prevent overfitting and improve predictive accuracy.

- **Flexibility**: XGBoost supports various objective functions and evaluation metrics, allowing users to customize the model's behavior based on the specific task and dataset. It can handle both categorical and numerical features and is robust to missing data.

- **Interpretability:** Despite its complexity, XGBoost provides insights into feature importance and model behavior through built-in visualization tools and feature importance scores. This transparency helps users understand the model's decision-making process and identify key factors driving predictions.

6.3.2 Limitations of XGBoost Algorithm

Despite its many advantages, XGBoost also has some limitations to consider:

- **Hyperparameter Tuning:** XGBoost requires careful tuning of hyperparameters such as tree depth, learning rate, and regularization parameters to achieve optimal performance.

- **Computational Resources:** Training XGBoost models can be resource-intensive, particularly when using large ensembles of trees or complex feature engineering. It may require significant computational resources and memory, especially for distributed training on clusters.

- **Overfitting:** Like other ensemble learning methods, XGBoost is susceptible to overfitting, especially when using large ensembles of trees or high-dimensional feature spaces.

6.4 Majority Voting Algorithm

Majority Voting Algorithm is a simple yet effective ensemble learning method. Ensemble learning involves combining the predictions of multiple models to create a more robust and reliable final prediction. In Majority Voting, a group of diverse machine learning models is trained on the same dataset, each employing unique algorithms or techniques. When it's time to make a prediction, each model casts its vote for the outcome, and the final prediction is determined by the majority's decision.

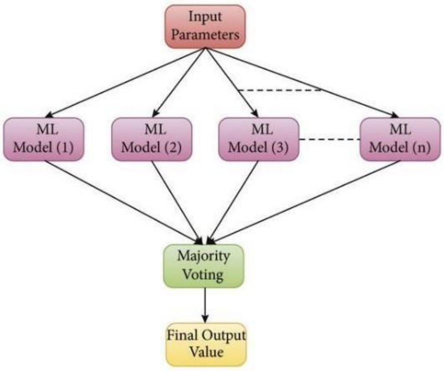


Figure 6.4: Majority Voting

6.4.1 Importance of Majority Voting Algorithm

The Majority Voting Algorithm offers several benefits and finds applications in various domains:

- The primary advantage of the Majority Voting Algorithm is its ability to improve prediction accuracy. Combining the predictions from multiple models leverages the strengths and compensates for the weaknesses of individual models. The aggregated prediction tends to be more reliable and robust, resulting in enhanced overall accuracy.
- Different machine learning models may have inherent biases due to their design or training data. By utilizing the Majority Voting Algorithm, these biases can be mitigated or even eliminated to a certain extent. The algorithm ensures that predictions are based on a diverse set of models, reducing the influence of individual biases and promoting fair and unbiased decision-making.
- The Majority Voting Algorithm enhances the robustness of machine learning systems. It reduces the risk of making incorrect predictions caused by the instability or limitations of individual models. By combining predictions, the algorithm creates a more stable and reliable decision-making framework that can handle diverse data patterns and adapt to different scenarios.
- In situations where individual models produce conflicting predictions or have

uncertainties, the Majority Voting Algorithm provides a mechanism to handle such uncertainties effectively. It considers the collective opinion of multiple models, which helps in making more informed decisions and reducing the impact of individual model variations or outliers.

- The Majority Voting Algorithm is flexible and compatible with various machine learning models and algorithms. It can be applied to both classification and regression problems, accommodating a wide range of applications. This versatility makes it suitable for diverse domains and allows integration with existing machine-learning pipelines.

6.4.2 Limitations of Majority Voting Algorithm

The majority voting algorithm, which combines the predictions of multiple base classifiers to make a final decision, has several disadvantages:

- **Sensitivity to Imbalanced Data:** Majority voting can be biased towards the majority class in imbalanced datasets. If one class heavily outweighs the others, the majority voting scheme may tend to predict that class more frequently, leading to poor performance on minority classes.

- **Equal Weighting of Classifiers:** In majority voting, each base classifier is typically given equal weight regardless of its individual performance. This can be problematic if some classifiers are consistently more accurate or reliable than others. In such cases, the influence of weaker classifiers may negatively impact the final decision.

- **Inefficiency with Large Number of Classes:** As the number of classes increases, the probability of ties in the voting process also increases. Resolving ties can become computationally expensive and may require additional measures to break ties effectively.

- **Lack of Probabilistic Interpretation:** Majority voting does not provide a probabilistic interpretation of the final decision. Instead, it simply outputs the most frequent class label without indicating the confidence or uncertainty associated with the prediction. This lack of probabilistic information may be crucial in certain applications where understanding the confidence level of predictions is essential.

Chapter 7 UML Diagrams

7.1 Use Case Diagram

A use case diagram is used to represent the dynamic behavior of a system. It encapsulates the system's functionality by incorporating use cases, actors, and their relationships. It models the tasks, services, and functions required by a system/subsystem of an application. It depicts the high-level functionality of a system and also tells how the user handles a system.

7.1.1 Purpose of Use Case Diagram

The main purpose of a use case diagram is to portray the dynamic aspect of a system. Itaccumulates the system's requirement, which includes both internal as well as external influences. It invokes persons, use cases, and several things that invoke the actors and elements accountable for the implementation of use case diagrams. It represents how anentity from the external environment can interact with a part of the system.

Following are the purposes of a use case diagram given below:

- It gathers the system's needs.
- It depicts the external view of the system.
- It represents the interaction between the actors.

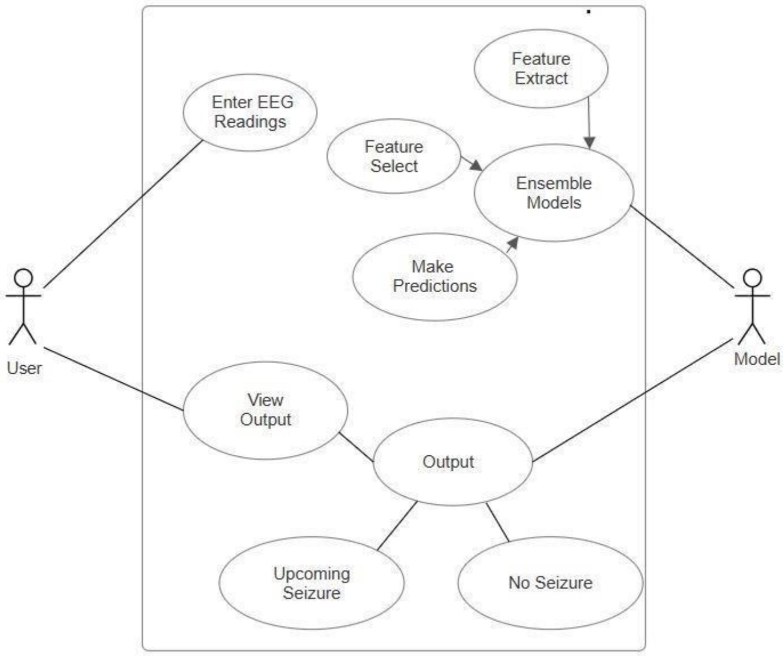


Figure 7.1: Use Case Diagram

7.2 Class Diagram

Class diagram is a static diagram. It represents the static view of an application. Class diagram is not only used for visualizing, describing, and documenting different aspectsof a system but also for constructing executable code of the software application.

Class diagram describes the attributes and operations of a class and also the constraintsimposed on the system. The class diagrams are widely used in the modelling of object-oriented systems because they are the only UML diagrams, which can be mapped directly with languages.

Class diagrams can also include other elements, such as interfaces, abstract classes, andpackages, which help to further organize and clarify the relationships between classes in a software system.

7.2.1 Purpose of Class Diagram

The purpose of class diagram is to model the static view of an application. Class diagrams are the only diagrams which can be directly mapped with object-oriented languages and thus widely used at the time of construction.
UML diagrams like activity diagram, sequence diagram can only give the sequence flowof the application, however class diagram is a bit different. It is the most popular UMLdiagram in the coder community.

The Purpose of the class diagram can be summarized as:

1. Describe responsibilities of a system.
2. Base for component and deployment diagrams.
3. Forward and reverse engineering.

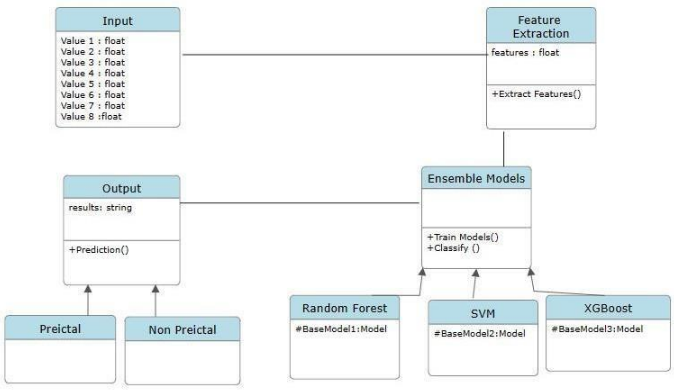


Figure 7.2: Class Diagram

7.3 Sequence Diagram

Sequence diagram is an event diagram. It reveals the circulation of messages in between various things or elements with time plus is commonly made use of to design the habits of a solitary usage instance. They work for developing as well as connecting the circulation of messages in between items in a software program system plus for recognizing possible issues or traffic jams in the system's actions.

In a series diagram the items are stood for as upright lifelines which diminish the size of the representation. Messages in between things are stood for as arrowheads that attach the lifelines with the message name as well as specifications composed over the arrowhead.

The order of the messages is revealed by their placement on the representation, with earlier messages on top plus later on messages near the bottom. Time is revealed horizontally with the left side of the representation standing for the beginning of the series plus the ideal side standing for the end.

7.3.1 Purpose of Sequence Diagram

The purpose of a sequence diagram is to visualize the interactions between objects in asoftware system over time. It shows the order in which messages are exchanged betweenobjects or components in a system, and can be used to model the behavior of a single use case or scenario.

Following are the purposes of a sequence diagram given below:

- Designing and modelling the behavior of a software system.
- Communicating system behavior to stakeholders.
- Testing and debugging.

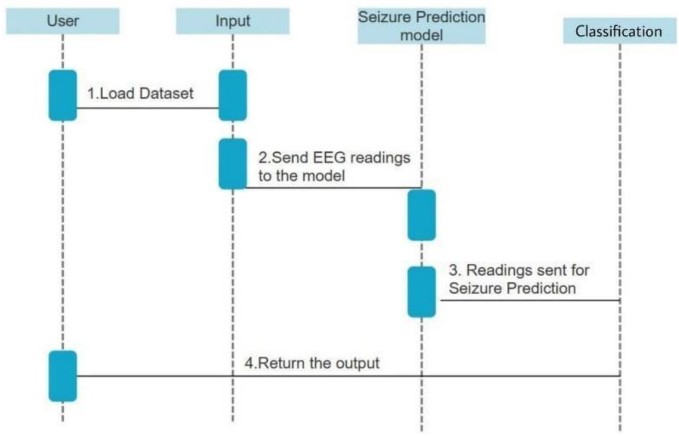


Figure 7.3: Sequence Diagram

7.4 Activity Diagram

Activity diagrams are a type of diagram used in software engineering and business process modeling to visualize the flow of activities involved in a system or process. These representations are for analyzing, creating, and recording processes as well as for communicating complex procedures to stakeholders.

The task templates are composed of nodes and edges. Nodes may stand for activities, decisions or other events in the system or process being modeled while edges represent the control flow between these nodes.

Tasks can be represented with different kinds of nodes, such as:

1. **Initial node:** Represents a starting point of the process or a system.
2. **Activity node:** Represents an activity or work that is performed as part of the process.
3. **Decision node:** Represents a place where one follows another depending on certain condition(s) only.
4. **Join node:** One place in which more than one path in the process come together again into a single path
5. **Final node:** Represents end point for either system or process

In task diagrams edges can be object flows or control flows. Control flow denotes how control moves between nodes while object flows denote how data or things move from one node to another

7.4.1 Purpose of Activity Diagram

The purpose of an activity diagram is to provide a visual representation of a system or process, making it easier to understand and analyze.

Here are some key points about the purpose of activity diagrams:

- Visualizing complex processes and workflows
- Improving communication and collaboration among stakeholders
- Analyzing the efficiency of a process and identifying potential bottlenecks orareas for improvement
- Designing new systems or processes by testing different scenarios andidentifying the best approach

- Documenting existing systems or processes to make it easier to maintain andupdate them over time.

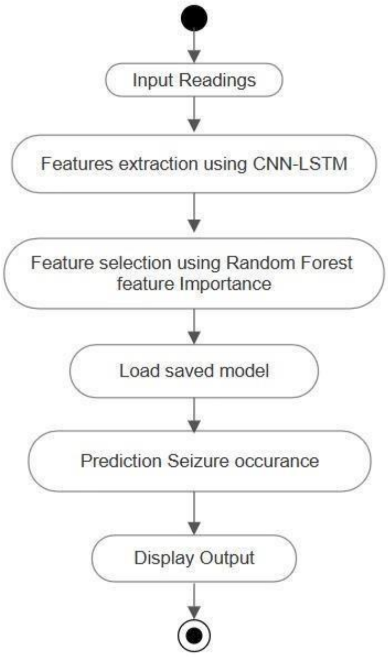


Figure 7.4: Activity Diagram

Chapter 8 Dataset Details

8.1 CHBMIT Dataset

The EEG data for training models in this study was obtained from the CHB-MIT Database. One of the most popular freely available EEG datasets for seizure detection and prediction is the CHB-MIT dataset. The CHB-MIT dataset consists of 22 pediatric patients' continuous scalp EEG recordings arranged into 23 cases. All signals are recorded at a resolution of 16 bits at 256 samples per second and the majority of the files contain recordings of 23 EEG channels. Annotations detailing the start and end of each seizure are included in the dataset. The prediction task typically views the period of time preceding each onset as the preictal stage. The Electroencephalography (EEG) dataset can be accessed by navigating through the directory. The dataset consists of raw EEG recordings saved in EDF format.

First, preprocessing involved extracting data from the CHB-MIT Scalp EEG Database, which was made available via PhysioNet in the European data format edf. Compatibility and accessibility are guaranteed by this format, which also includes metadata regarding epileptic periods in the dataset. The edf files were carefully used to extract the voltage levels from the EEG electrodes, which served as the basis for further preprocessing procedures. The process of annotating and classifying ictal and preictal states was carried out using the comprehensive data included in the edf files, guaranteeing that both states were fairly represented. To enable accurate labelling and further analysis, separate files were kept for the ictal and preictal periods.

During the last stage of preprocessing, discrete states were defined for further analysis by carefully labelling data points that represented preictal and non pre ictal states with '0' and '1' respectively.

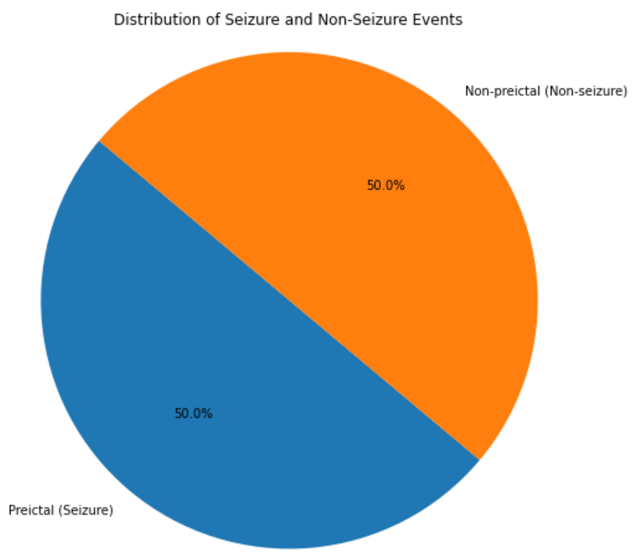


Figure 8.1: Distribution of Preictal and Non-Preictal classes

8.2 Dataset Splitting

Training Set (70%): Enriched with a diverse range of EEG patterns, including both normal and seizure activities, the training set forms the foundation for the model to learn and generalize.

Validation Set (15%): This set acts as an intermediate checkpoint during the training process. Models are evaluated on this dataset to gauge their performance and make adjustments, preventing overfitting to the training data.

Testing Set (15%): Kept entirely separate until the model is fully trained, the testing set serves as an unbiased benchmark to assess the model's real-world predictive capabilities. It provides insights into the model's ability to generalize to unseen data, especially concerning seizure detection.

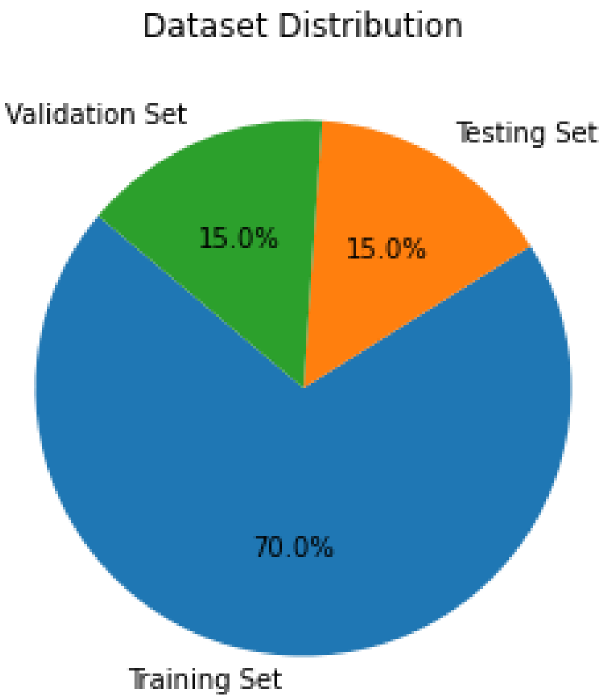


Figure 8.2: Dataset Splitting

Chapter 9 Codes

9.1 Code for Data Acquisition

LOADING THE DATASET

#In[1]:

```
import mne
import os
patient_folder = 'C:/Users/Admin/Desktop/Major Project/Final-Year-Project-
ML/Dataset/'combined_fif_path = os.path.join(patient_folder,
'Combined_eeg.fif')

if os.path.exists(combined_fif_path):

    try:
        raw_data_combined = mne.io.read_raw_fif(combined_fif_path,
preload=True)
        print("Combined data loaded into memory successfully.")

    except Exception as e:
        print(f"Error loading combined file: {e}")
else:
    print(f"Combined file not found at: {combined_fif_path}. Please check the
file path.")
```

#Out[1]:

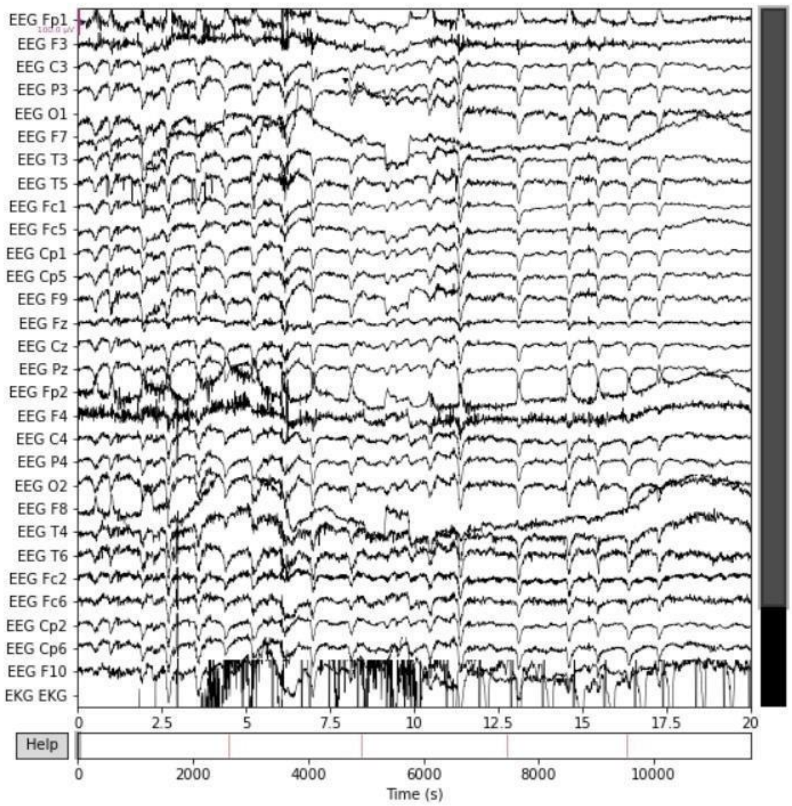
```
Reading 0 ... 5980671  =      0.000 ... 11680.998 secs...
Combined data loaded into memory successfully.
```

#In[2]:

```
info_combined = raw_data_combined.info
print(info_combined)

raw_data_combined.plot(n_channels=30, duration=20, scalings={'eeg': 50e-6})
plt.show()
```

#Out[2]:

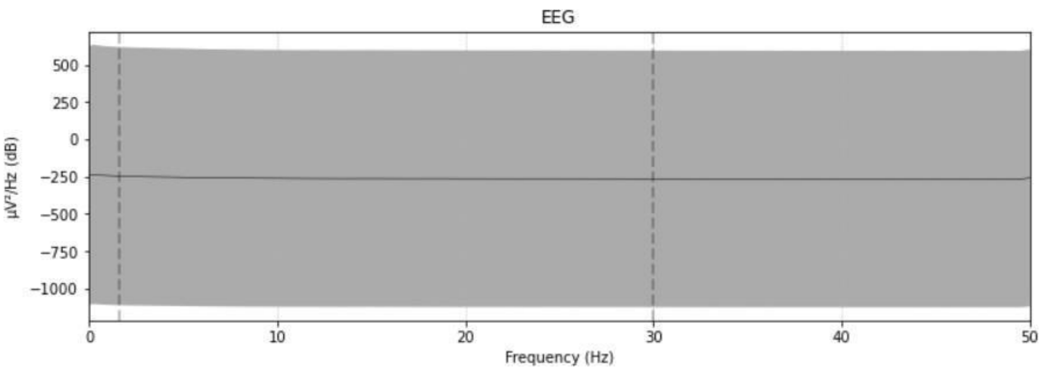
```
Using matplotlib as 2D backend.
```



#In[3]:

raw_data_combined.plot_psd(fmax=50, average=True)
plt.show()

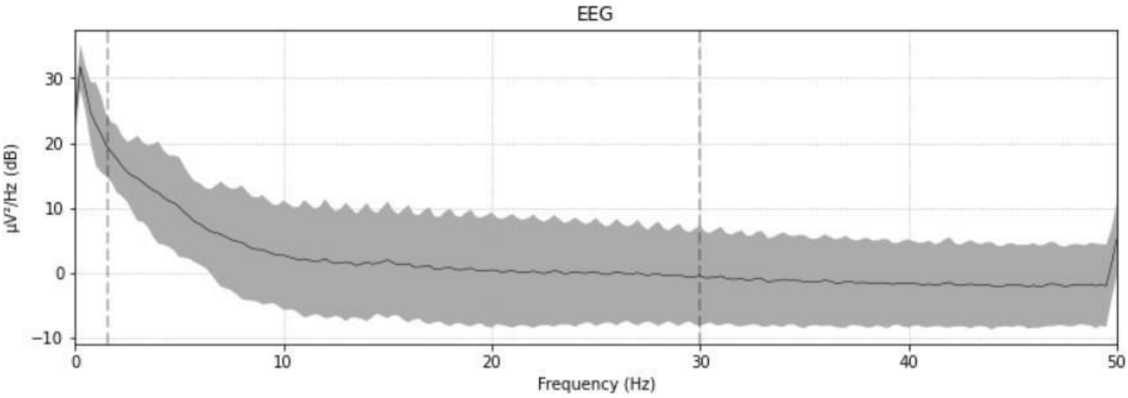
#Out[3]:



#In[4]:

```
# Mark channels as bad
raw_data_combined.info['bads'] = ['SPO2', 'HR', 'MK']

# Plot PSD
raw_data_combined.plot_psd(fmax=50, average=True)
plt.show()
```

#Out[4]:



9.2 Code for Feature Extraction

#In[5]:

```
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from keras.models import Sequential
from keras.layers import Conv1D, MaxPooling1D, LSTM, Dense, Flatten

# Load the CSV dataset
dataset = pd.read_csv('D:/Project/new.csv')
X = dataset.iloc[:, :-1].values # Features
y = dataset.iloc[:, -1].values  # Outcome

# Split the dataset into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
random_state=42)

# Standardize the features
scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)
```

```
# Reshape the features for CNN input (assuming each sample has 23 channels)
X_train = X_train.reshape(X_train.shape[0], X_train.shape[1], 1)
X_test = X_test.reshape(X_test.shape[0], X_test.shape[1], 1)

# Define the CNN-LSTM model
model = Sequential()

# Convolutional layers
model.add(Conv1D(filters=32, kernel_size=3, activation='relu',
input_shape=(X_train.shape[1], 1)))
model.add(MaxPooling1D(pool_size=2))
model.add(Conv1D(filters=64, kernel_size=3, activation='relu'))
model.add(MaxPooling1D(pool_size=2))

# LSTM layer
model.add(LSTM(units=50, return_sequences=True))
model.add(LSTM(units=50))

# Dense layers
model.add(Dense(units=64, activation='relu'))
model.add(Dense(units=1, activation='sigmoid'))

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

# Train the model
model.fit(X_train, y_train, epochs=5, batch_size=32, validation_data=(X_test,
y_test))

# Evaluate the model
loss, accuracy = model.evaluate(X_test, y_test)
print(f'Test Loss: {loss}, Test Accuracy: {accuracy}')

# Extract features using the trained model
feature_extractor = Sequential(model.layers[:-1]) # Exclude the output layer
X_train_features = feature_extractor.predict(X_train)
X_test_features = feature_extractor.predict(X_test)

# Now you can use the extracted features for further analysis or classification tasks
```

#**Out[5]:**

```
Epoch 1/5
250/250 ━━━━━━━━━━━━━━━━━━━━ 13s 20ms/step - accuracy: 0.7719 - loss: 0.4086 - val_accuracy: 0.8795 - val_loss: 0.2851
Epoch 2/5
250/250 ━━━━━━━━━━━━━━━━━━━━ 4s 15ms/step - accuracy: 0.9063 - loss: 0.2264 - val_accuracy: 0.8905 - val_loss: 0.2787
Epoch 3/5
250/250 ━━━━━━━━━━━━━━━━━━━━ 4s 14ms/step - accuracy: 0.9082 - loss: 0.2229 - val_accuracy: 0.9215 - val_loss: 0.1965
Epoch 4/5
250/250 ━━━━━━━━━━━━━━━━━━━━ 4s 15ms/step - accuracy: 0.9284 - loss: 0.1820 - val_accuracy: 0.9200 - val_loss: 0.1891
Epoch 5/5
250/250 ━━━━━━━━━━━━━━━━━━━━ 4s 14ms/step - accuracy: 0.9350 - loss: 0.1703 - val_accuracy: 0.9375 - val_loss: 0.1559
63/63 ━━━━━━━━━━━━━━━━━━━━ 1s 7ms/step - accuracy: 0.9365 - loss: 0.1734
Test Loss: 0.15594246983528137, Test Accuracy: 0.9375
250/250 ━━━━━━━━━━━━━━━━━━━━ 2s 6ms/step
63/63 ━━━━━━━━━━━━━━━━━━━━ 1s 18ms/step
```

9.3 Code for Model Training and Evaluation

#In[6]:

```
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.svm import SVC
from xgboost import XGBClassifier
from sklearn.feature_selection import SelectFromModel
from sklearn.metrics import accuracy_score, classification_report,
confusion_matrix
# Load the CSV dataset
dataset = pd.read_csv('D:/Project/all.csv')
# Handling missing values
dataset.dropna(inplace=True) # Drop rows with missing values
# Assuming 'data.csv' contains your dataset
X = dataset.iloc[:, :-1].values # Features
y = dataset.iloc[:, -1].values  # Outcome
# Split the dataset into training, validation, and testing sets
X_train, X_temp, y_train, y_temp = train_test_split(X, y, test_size=0.3,
random_state=42)
X_val, X_test, y_val, y_test = train_test_split(X_temp, y_temp, test_size=0.5,
random_state=42)
# Feature selection using Random Forest Feature Importance
feat_selector = SelectFromModel(RandomForestClassifier(n_estimators=100,
random_state=42))
X_train_selected = feat_selector.fit_transform(X_train, y_train)
X_val_selected = feat_selector.transform(X_val)
X_test_selected = feat_selector.transform(X_test)
# Individual classifiers
rf_clf = RandomForestClassifier(n_estimators=100, random_state=42)
svm_clf = SVC(kernel='rbf', gamma='scale')
xgb_clf = XGBClassifier()
# Train the models
rf_clf.fit(X_train_selected, y_train)
svm_clf.fit(X_train_selected, y_train)
xgb_clf.fit(X_train_selected, y_train)
# Predictions on validation set
rf_val_pred = rf_clf.predict(X_val_selected)
svm_val_pred = svm_clf.predict(X_val_selected)
xgb_val_pred = xgb_clf.predict(X_val_selected)
# Combine predictions using weighted average voting
ensemble_preds_val = np.array([rf_val_pred, svm_val_pred, xgb_val_pred])
# Define weights for each classifier
```

```
weights = np.array([0.4, 0.3, 0.5]) # Example weights, you can adjust as needed
weights = weights.reshape(-1, 1) # Reshape weights to match
ensemble_preds_val shape
# Calculate weighted average for validation set
weighted_sum_val = np.sum(ensemble_preds_val * weights, axis=0)
# Round the weighted sum to the nearest integer to get the majority vote
majority_vote_val = np.round(weighted_sum_val).astype(int)
# Calculate accuracy on validation set
ensemble_accuracy_val = accuracy_score(y_val, majority_vote_val)
print(f'Ensemble Accuracy on Validation Set: {ensemble_accuracy_val}')
# Predictions on testing set
rf_test_pred = rf_clf.predict(X_test_selected)
svm_test_pred = svm_clf.predict(X_test_selected)
xgb_test_pred = xgb_clf.predict(X_test_selected)
# Combine predictions using weighted average voting for testing set
ensemble_preds_test = np.array([rf_test_pred, svm_test_pred, xgb_test_pred])
# Calculate weighted average for testing set
weighted_sum_test = np.sum(ensemble_preds_test * weights, axis=0)
# Round the weighted sum to the nearest integer to get the majority vote
majority_vote_test = np.round(weighted_sum_test).astype(int)
# Calculate accuracy on testing set
ensemble_accuracy_test = accuracy_score(y_test, majority_vote_test)
print(f'Ensemble Accuracy on Testing Set: {ensemble_accuracy_test}')
# Classification report and confusion matrix for ensemble classifier on testing set
print("Ensemble Classifier Metrics on Testing Set:")
print(classification_report(y_test, majority_vote_test))
print("Confusion Matrix:")
print(confusion_matrix(y_test, majority_vote_test))
```

#**Out[6]:**

```
Ensemble Accuracy on Validation Set: 0.9476666666666667
Ensemble Accuracy on Testing Set: 0.9473333333333334
Ensemble Classifier Metrics on Testing Set:
              precision    recall  f1-score   support

         0.0       0.93      0.97      0.95      1532
         1.0       0.97      0.92      0.94      1468

    accuracy                           0.95      3000
   macro avg       0.95      0.95      0.95      3000
weighted avg       0.95      0.95      0.95      3000

Confusion Matrix:
[[1490   42]
 [ 116 1352]]
```

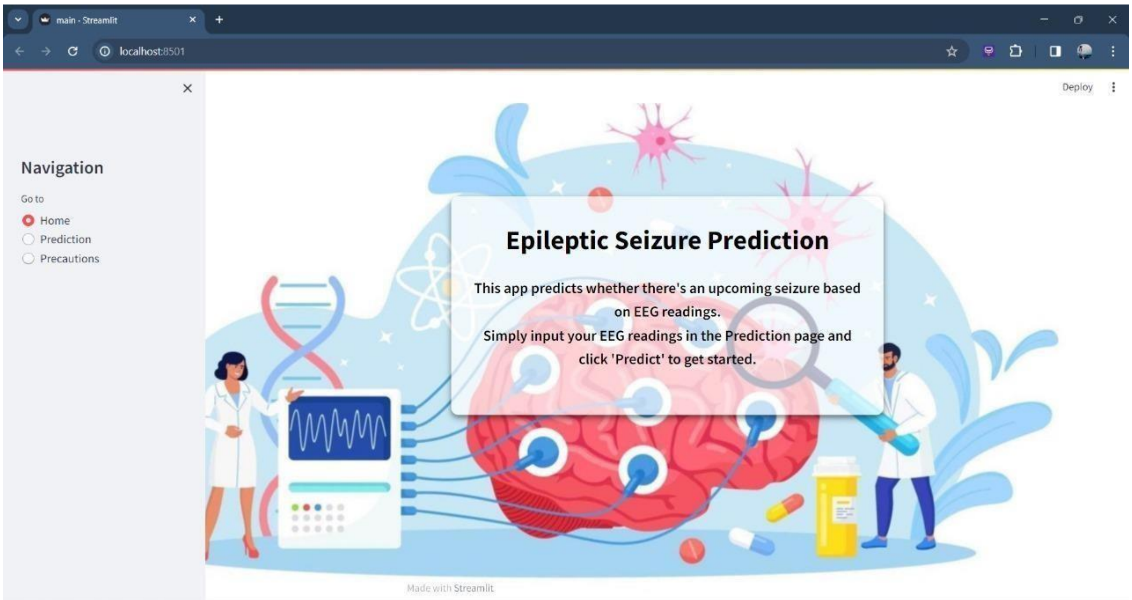
9.4 Code for Web Application

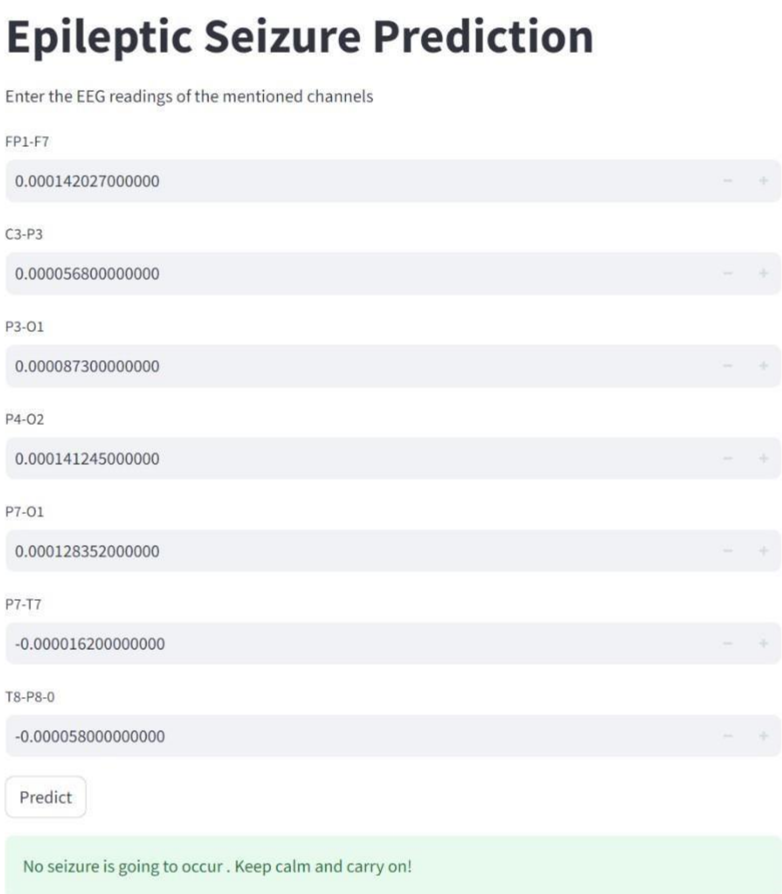
#In[15]:

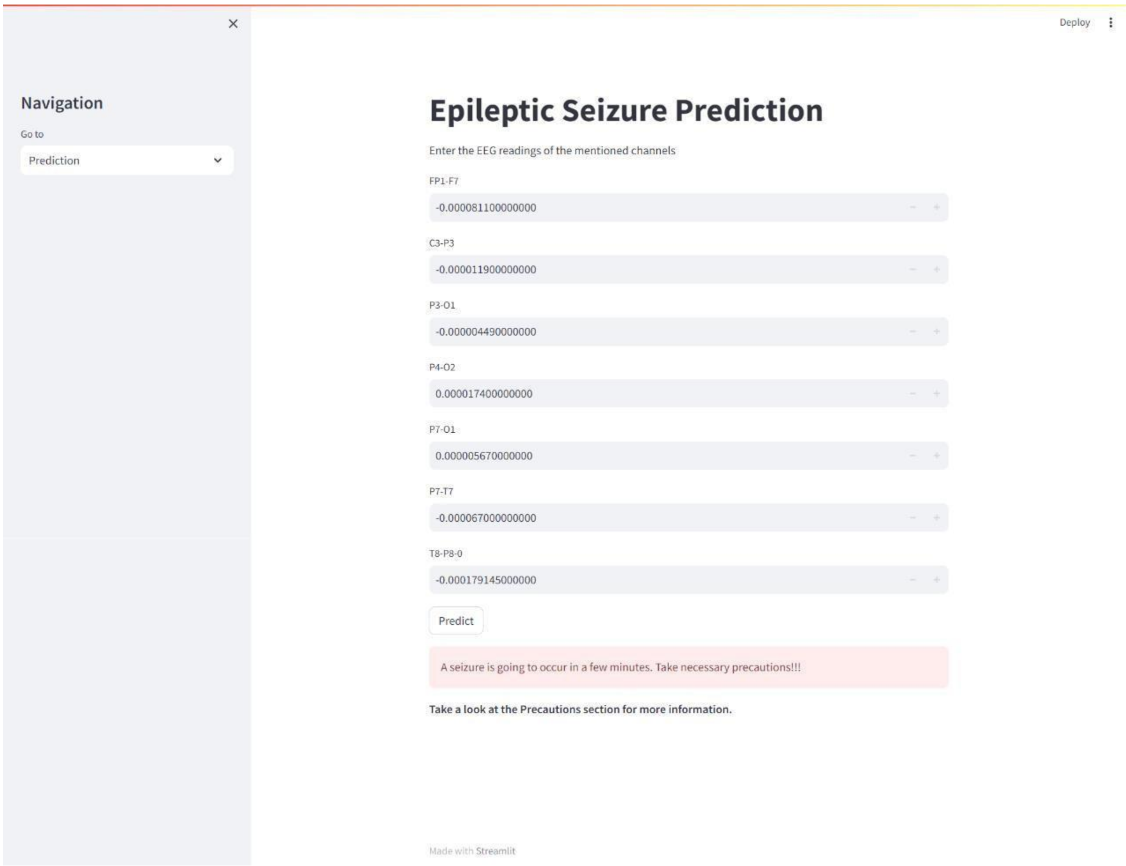
```
import streamlit as st
import pickle
import numpy as np
model = pickle.load(open('EE_model.pkl', 'rb'))
def risk_potability_prediction(input_data):
    input_as_array = np.array(input_data).reshape(1,-1)
    prediction = model.predict(input_as_array)[0]
    return prediction
def main():
    st.set_page_config(page_title='EEG - Based Epileptic Seizure Prediction',
page_icon=':potable_water:')
    st.title('EEG - Based Epileptic Seizure Prediction')
    st.write('This app predicts Epileptic Seizure Prediction')
    st.subheader('Epileptic Seizure Prediction')
    mar = st.number_input('# FP1-F7', format="%.7f", min_value=0.0,
max_value=100.0, value=7.0, step=0.1)
    deb = st.number_input('C3-P3', format="%.7f", min_value=0.0, value=50.0,
step=1.0)
    dis = st.number_input('P3-O1', format="%.7f", min_value=0.0, value=50.0,
step=1.0)
    gen = st.number_input('P4-O2', format="%.7f", min_value=-0.0000303,
value=50.0, step=1.0)
    crs = st.number_input('P7-O1', format="%.7f", min_value=0.0,
max_value=1000.0, step=1.0)
    gdp = st.number_input('P7-T7', format="%.7f", min_value=-0.0000303,
value=50.0, step=1.0)
    pqg = st.number_input('T8-P8-0', format="%.7f", min_value=0.0, value=50.0,
step=1.0)
    pqg1 = st.number_input('T8-P8-1', format="%.7f", min_value=0.0, value=50.0,
step=1.0)
try:
    prediction = risk_potability_prediction(input_data)
    if prediction == 0:
        st.error('The Patient is affected by Epileptic Seizure.')
    else:
        st.success('The Patient is not affected by Epileptic Seizure.')
except Exception as e:
    st.error(f"An error occurred: {str(e)}")

    st.write('---')
```

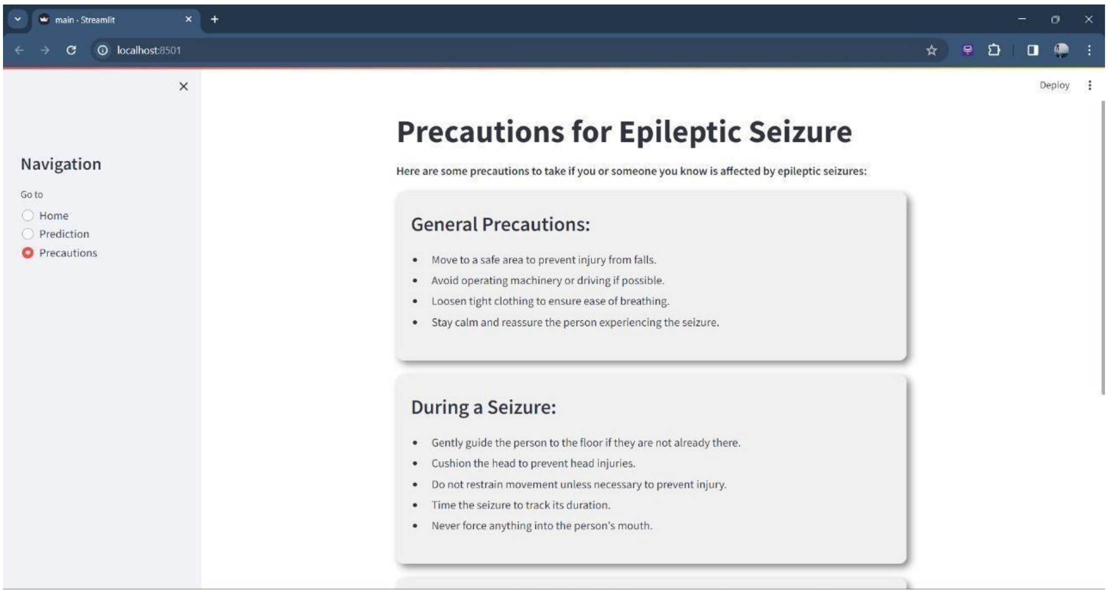
```
if__name__== '_main_':
    main()
```

#Out[16]:






Deploy
Navigation
Go to
Prediction
Epileptic Seizure Prediction
Enter the EEG readings of the mentioned channels
FP1-F7
-0.000081100000000
C3-P3
-0.000011900000000
P3-O1
-0.000004490000000
P4-O2
0.000017400000000
P7-O1
0.000005670000000
P7-T7
-0.000067000000000
T8-P8-0
-0.000179145000000
Predict
A seizure is going to occur in a few minutes. Take necessary precautions!!!
Take a look at the Precautions section for more information.
Made with Streamlit


main · Streamlit
localhost:8501
Deploy
Navigation
Go to
Home
Prediction
Precautions
Precautions for Epileptic Seizure
Here are some precautions to take if you or someone you know is affected by epileptic seizures:
General Precautions:
Move to a safe area to prevent injury from falls.
Avoid operating machinery or driving if possible.
Loosen tight clothing to ensure ease of breathing.
Stay calm and reassure the person experiencing the seizure.
During a Seizure:
Gently guide the person to the floor if they are not already there.
Cushion the head to prevent head injuries.
Do not restrain movement unless necessary to prevent injury.
Time the seizure to track its duration.
Never force anything into the person's mouth.

Chapter 10 Experimental Analysis and Results

10.1 Evaluation Metrics

#In[7]:

```
print("Ensemble Classifier Metrics on Validation Set:")
print(classification_report(y_val, majority_vote_val))
print("Confusion Matrix:")
print(confusion_matrix(y_val, majority_vote_val))
```

#**Out[7]:**

```
Ensemble Classifier Metrics on Validation Set:
              precision    recall  f1-score   support

         0.0       0.92      0.98      0.95      1485
         1.0       0.98      0.92      0.95      1515

    accuracy                           0.95      3000
   macro avg       0.95      0.95      0.95      3000
weighted avg       0.95      0.95      0.95      3000

Confusion Matrix:
[[1450   35]
 [ 122 1393]]
```

#In[8]:

```
from sklearn.metrics import accuracy_score, precision_score, recall_score,
f1_score, confusion_matrix
# Accuracy
accuracy = accuracy_score(y_test, majority_vote_test)
print(f'Accuracy: {accuracy}')
# Precision
precision = precision_score(y_test, majority_vote_test)
print(f'Precision: {precision}')
# Recall (Sensitivity)
recall = recall_score(y_test, majority_vote_test)
print(f'Recall: {recall}')
# F1 Score
f1 = f1_score(y_test, majority_vote_test)
print(f'F1 Score: {f1}')
# Confusion Matrix
conf_matrix = confusion_matrix(y_test, majority_vote_test)
print('Confusion Matrix:')
```

print(conf_matrix)

#Out[8]:

```
Accuracy: 0.9473333333333334
Precision: 0.96987087517934
Recall: 0.9209809264305178
F1 Score: 0.9447938504542279
Confusion Matrix:
[[1490   42]
 [ 116 1352]]
```

#**In[9]:**

```
# Specificity
specificity = conf_matrix[0, 0] / (conf_matrix[0, 0] + conf_matrix[0, 1])
print(f'Specificity: {specificity}')
```

#**Out[9]:**

```
Specificity: 0.9725848563968669
```

#In[10]:

```
# False Positive Rate (FPR)
fpr = conf_matrix[0, 1] / (conf_matrix[0, 1] + conf_matrix[0, 0])
print(f'False Positive Rate (FPR): {fpr}')
```

#**Out[10]:**

```
False Positive Rate (FPR): 0.02741514360313316
```

#**In[11]:**

```
# False Negative Rate (FNR)
fnr = conf_matrix[1, 0] / (conf_matrix[1, 0] + conf_matrix[1, 1])
print(f'False Negative Rate (FNR): {fnr}')
```

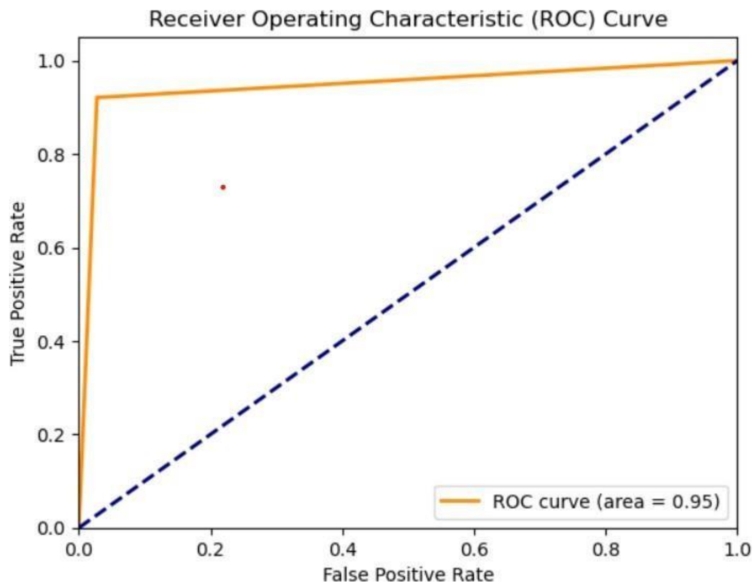
#**Out[11]:**

```
False Negative Rate (FNR): 0.07901907356948229
```

#In[12]:

```
from sklearn.metrics import roc_curve, auc
import matplotlib.pyplot as plt

# Compute ROC curve and ROC area for each class
fpr, tpr, _ = roc_curve(y_test, majority_vote_test)
roc_auc = auc(fpr, tpr)

# Plot ROC curve
plt.figure()
plt.plot(fpr, tpr, color='darkorange', lw=2, label='ROC curve (area = %0.2f)' %
roc_auc)
plt.plot([0, 1], [0, 1], color='navy', lw=2, linestyle='--')
plt.xlim([0.0, 1.0])
plt.ylim([0.0, 1.05])
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('Receiver Operating Characteristic (ROC) Curve')
plt.legend(loc='lower right')
plt.show()

# Print AUC-ROC score
print(f'AUC-ROC Score: {roc_auc}')
```

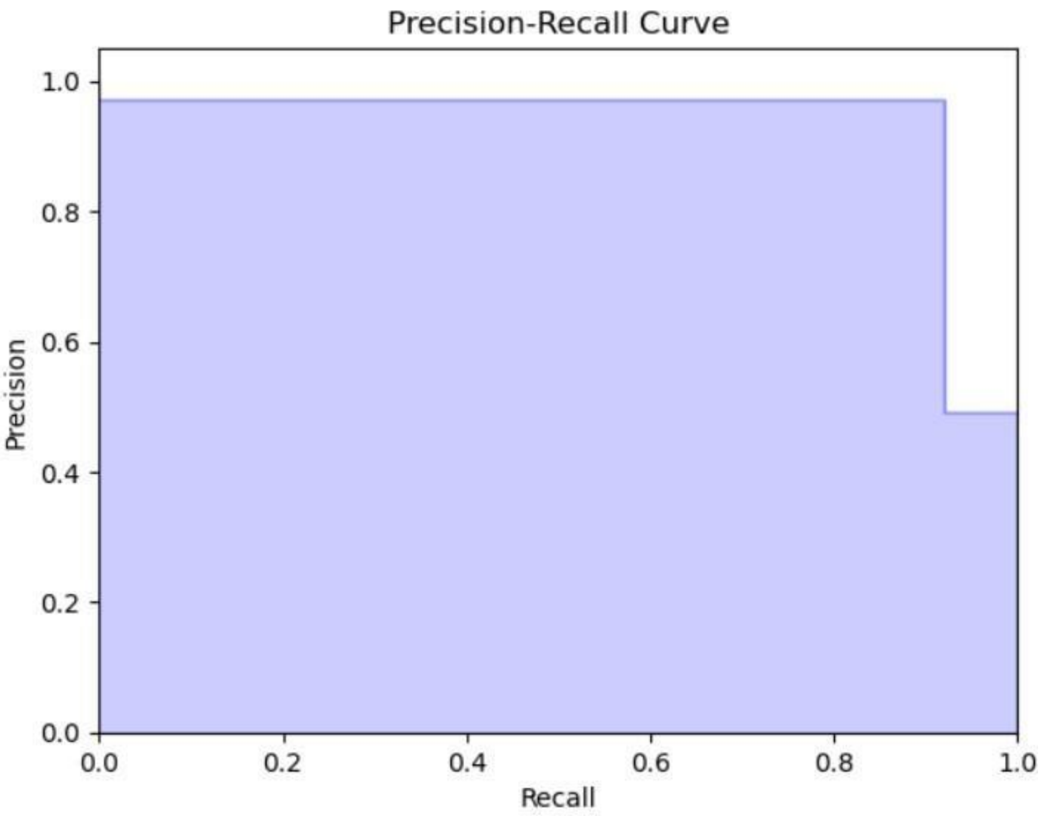
#Out[12]:



#In[13]:

```
from sklearn.metrics import precision_recall_curve, average_precision_score
# Compute precision-recall curve and AP score for each class
precision, recall, _ = precision_recall_curve(y_test, majority_vote_test)
ap = average_precision_score(y_test, majority_vote_test)
# Plot precision-recall curve
plt.figure()
plt.step(recall, precision, color='b', alpha=0.2, where='post')
plt.fill_between(recall, precision, step='post', alpha=0.2, color='b')
plt.xlabel('Recall')
plt.ylabel('Precision')
plt.ylim([0.0, 1.05])
plt.xlim([0.0, 1.0])
plt.title('Precision-Recall Curve')
plt.show()
# Print Average Precision (AP) score
print(f'Average Precision (AP): {ap}')
```

#**Out[13]:**

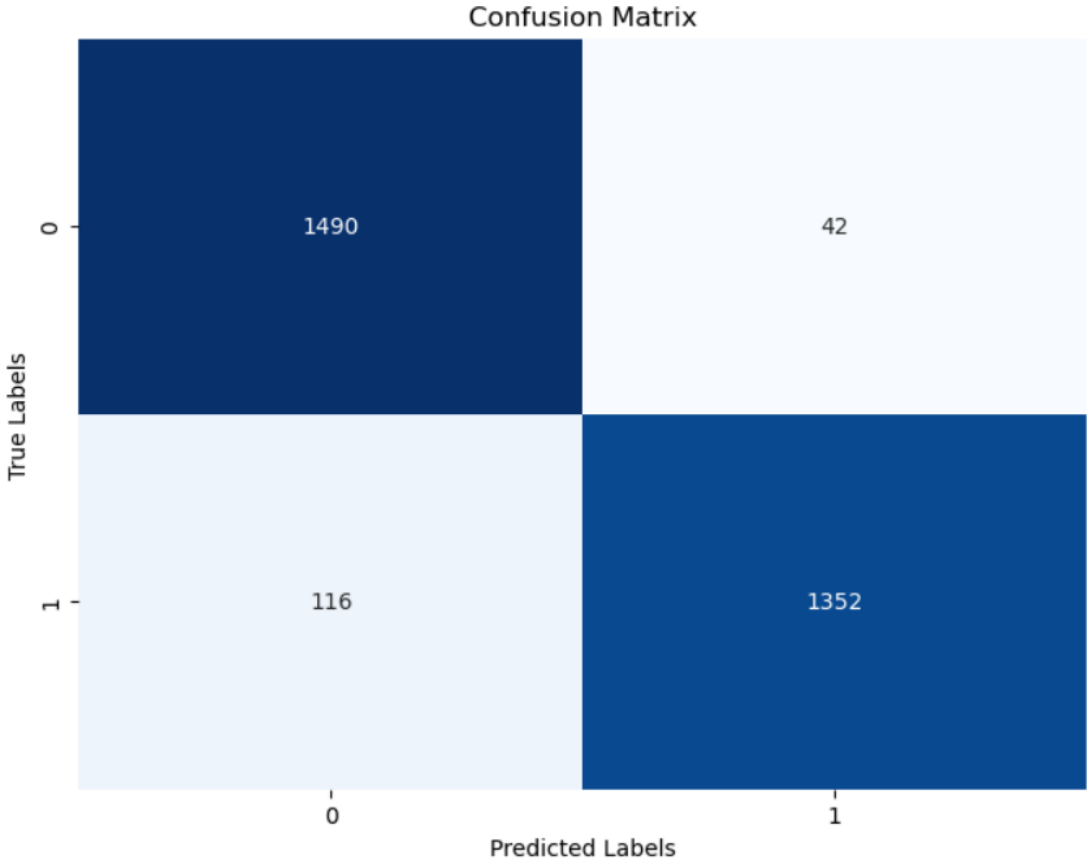



```
Average Precision (AP): 0.9318992438073123
```

#In[14]:

```
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.metrics import confusion_matrix

# Compute confusion matrix
conf_matrix = confusion_matrix(y_test, majority_vote_test)

# Plot confusion matrix
plt.figure(figsize=(8, 6))
sns.heatmap(conf_matrix, annot=True, fmt='d', cmap='Blues', cbar=False)
plt.title('Confusion Matrix')
plt.xlabel('Predicted Labels')
plt.ylabel('True Labels')
plt.show()
```

#Out[14]:



Chapter 11 Installations

11.1 Introduction

The implementation phase is a crucial stage in the software development life cycle. Thisphase involves the actual implementation of the project plan that was developed during the previous stages, such as the requirement gathering and analysis phase and the designphase. In other words, it is the phase where the project plan is put into action.

The execution stage usually begins when a substantial part of the code for the program has actually been composed. Now the group in charge of execution concentrates on equating the needs defined in the need stage right into a sensible framework that can be applied in a program's language. This includes creating code together with creating formulas that fulfill the requirements detailed in the job strategy.

In our task, we have actually chosen to take advantage of both Jupyter Notebook as well as Spyder as our Integrated Development Environments (IDEs) for coding jobs. Jupyter Notebook supplies an easy-to-use user interface for code make-up as well as implementation, promoting smooth partnership within our group. Additionally, Spyder an open-source IDE customized for clinical shows in Python, enhances our toolkit with its effective functions as well as abilities. By using both Jupyter Notebook as well as Spyder we guarantee adaptability together with effectiveness in our growth operations encouraging our group to take on varied difficulties easily.

11.2 Tools Used

11.2.1 Jupyter Notebook

Our Integrated Development Environment (IDE) was Jupyter Notebook. A web-based environment used frequently in data science and machine learning projects is Jupyter Notebook. With its help, anyone can create and share documents with code samples, formulas, illustrations, and explanatory text. Moreover, Python, R, Julia, and other programming languages are supported by Jupyter Notebook. The interface isuser-friendly and features a web browser-based editor that makes it easy to write and execute code. The notebook layout consists of cells that hold either code or text allowing for an organization of code, into sections. Jupyter Notebook includes integrated features for visualizing data through tools like Matplotlib and Seaborn simplifying the creation of charts and graphs, within the notebook itself.

11.2.2 Spyder

Spyder is an environment that's available, for free and open source. It is created in Python by scientists, engineers and data analysts for their use. The platform offers a blend of editing, analysis, debugging and profiling tools similar to those found in development software. Additionally, it provides features for data exploration, interactive execution, deep inspection and visually appealing data visualization. Making it a versatile tool, for work.

11.3 Libraries Used

11.3.1 Numpy

NumPy, short, for python is a Python library used for computing and handling arrays with dimensions or just one dimension. It's an open-source Python library known as Numerical Python. It is designed to perform complex mathematical, image processing, quantum computing, and statistical operations, etc., on matrices and multidimensional arrays.

11.3.2 Pandas

Pandas is a Python library used for working with data sets. It has functions for analyzing, cleaning, exploring, and manipulating data. Pandas allows us to analyze big data and make conclusions based on statistical theories. Pandas can clean messy data sets, and make them readable and relevant. Relevant data is very important in data science.

11.3.3 Matplotlib

Matplotlib is an amazing visualization library in Python for 2D plots of arrays. Matplotlib is a multi-platform data visualization library built on NumPy arrays and designed to work with the broader SciPy stack. Matplotlib is a cross-platform, data visualization and graphical plotting library for Python and its numerical extension NumPy. As such, it offers a viable open-source alternative to MATLAB. Developers can also use matplotlib's APIs (Application Programming Interfaces) to embed plots inGUI applications.

11.3.4 Seaborn

Seaborn is a library that uses Matplotlib underneath to plot graphs. It will be used to visualize random distributions. Seaborn is a library for making statistical graphics in Python. It builds on top of matplotlib and integrates closely with pandas data structures.Seaborn helps you explore and understand your data.

11.3.5 Sklearn

Scikit-learn (Sklearn) is the most useful and robust library for machine learning in Python. It offers a variety of tools, for machine learning and statistical analysis, such as classification, regression, clustering and dimensionality reduction through an interface, in Python.

11.3.6 Keras

Keras is a high-level neural network API developed in Python that may be used with TensorFlow, CNTK, or Theano. It offers a simple interface for creating deep learning models and enables quick and easy experimentation. Keras includes a large number of pre-built layers, loss functions, and optimizers that may be simply coupled to create a custom deep learning model.

11.3.7 Tensorflow

TensorFlow is an open-source software library that supports dataflow and differentiable programming across a variety of activities. It is utilized in machine and deep learning applications like as neural networks, natural language processing, and computer vision. TensorFlow offers both a low-level API for creating custom deep learning models and high-level APIs like Keras, which make it easier to construct and train deep learning models.

11.3.8 MNE

MNE-Python is an open-source Python package that processes, analyzes, and visualizes functional neuroimaging data (EEG, MEG, sEEG, ECoG, and fNIRS). Depending on your analytic requirements, you may choose to install a number of related or compatible software programs.

Chapter 12 Conclusion and Future Score

12.1 Conclusion

In conclusion, our project has successfully developed an advanced epileptic seizure prediction system leveraging ensemble learning techniques applied to EEG signals. Utilizing a diverse preprocessed dataset sourced from the CHB-MIT database, our approach combined deep learning models such as CNN and LSTM with traditional classifiers including XGBoost, SVM, and RF. The ensemble classifier exhibited robust performance, achieving notable results on both the validation and testing sets: an accuracy of approximately 94.77% on the validation set and 94.73% on the testing set. Precision, recall, and F1-scores remained consistently high for both seizure and non-seizure instances. This model's accurate prediction of impending seizures provides a crucial opportunity for timely intervention, empowering individuals with epilepsy to take necessary precautions. Through meticulous analysis and evaluation of prediction results, we have gained valuable insights into the model's behavior and identified areas for potential improvement. Overall, our ensemble-based seizure prediction method shows potential for improving the quality of life for people with epilepsy, contributing to better healthcare outcomes, and meeting an urgent societal need.

12.2 Future Scope

Within the domain of epilepsy intervention, our ensemble-based epileptic seizure prediction model represents a major intermediate step towards further development and generalization with yet unexplored barriers and prospects. In hindsight, the following areas of potential exploration and improvement can be pinpointed. Making the model capable of real-time predictions can be a major step towards patient-centric healthcare, as it could be implemented with wearable EEG devices and smartphone apps for immediate notification and possible emergency actions for the individual. Alternative techniques of feature engineering such as multi-modal data integration and individual model formulation also appear to be a possibility. Clinical validation and integration with the help of healthcare professionals are particularly important for regulatory approval and broader adoption. In addition, a commitment to the highest levels of interpretability and explainability of the model’s predictions will result in greater transparency and acceptance by clinicians and patients. Efforts dedicated to ensuring the model is improved continually and expanded globally are critical for the future of epilepsy and health equity worldwide. Thus, by supporting various ways to innovate and collaborate, we can achieve further progress in epilepsy management and significantly improve the lives of those living with epilepsy.

References

1. Mohamad Shahbazi, Hamid Aghajan (2018). A Generalizable Model for Seizure Prediction Based on Deep Learning using CNN-LSTM Architecture, 11.

2. Ziyu Wang, Jie Yang, Mohamad Sawan (2021). A Novel Multi-Scale Dilated 3D CNN for Epileptic Seizure Prediction, 4.

3. N. N. Saranya, D. Karthika Renuka, R. Geetha Rajakumari (2021). Ensemble Classification for Epileptic Seizure Prediction, 5.

4. Felix George, Alex Joseph, Bibin Baby, Alex John, Tonny John, M Deepak, G Nithin, P.S. Sathidevi (2020). Epileptic Seizure Prediction using EEG Images, 4.

5. Ruyan Wang, Linhai Wang, Peng He, Yaping Cui, Dapeng Wu (2022). Epileptic Seizure Prediction Based on Unsupervised Learning for Feature Extraction, 6.

6. Tian Yu, Boyuan Cui, Yaqian Xu, Xilin Liu (2023). Refine EEG Spectrogram Synthesized by Generative Adversarial Network for Improving the Prediction of Epileptic Seizures, 4.

7. Milind Natu, shilpa Gite, Mrinal Bachute, Ketan Kotecha (2022). Review on Epileptic Seizure Prediction: Machine Learning and Deep Learning Approaches, 7.

8. Xiaoyan Wei, Lin Zhou, Zhen Zhang, Ziyi Chen, Yi Zhou (2019). Early prediction of epileptic seizures using a long-term recurrent convolutional network, 5.

9. Usman, S. M. Khalid, S Akhtar, R. Bortolotto, Z. Bashir, Zamp Qiu. Using scalp EEG and intracranial EEG signals for predicting epileptic seizures: Review of available methodologies, 8.

10. Anibal Romney, Vidya Manian (2021). Optimizing Seizure Prediction from Reduced Scalp EEG Channels Based on Spectral Features and MAML, 6.

11. Prosper Chiemezuo, Noble-Nnakenyi, Kehinde Adebola Olatunji, Oluwatoyin Bunmi Abiola, Abiodun Oguntimilehin (2022). Predicting Epileptic Seizures using Ensemble Method, 7.

12. Nhan Duy Truong, Levin Kuhlmann, Mohammad Reza Bonyadi, Damien Querlioz, Luping Zhou, Omid Kavehei (2019). Epileptic Seizure Forecasting with Generative Adversarial Networks, 9.

13. Tian Yu, Boyuan Cui, Yaqian Xu, Xilin Liu (2019). Epileptic Seizure Prediction with Multi-View Convolutional Neural Networks, 10.

14. Simin Khalilpour, Amin Ranjbar, Mohammad Bagher Menhaj, Afshin Sandooghdar (2020). Application of 1-D CNN to Predict Epileptic Seizures using EEG Records, 11.

15. Khansa Rasheed, Junaid Qadir, Senior Member, Levin Kuhlmann, and Adeel Razi (2021). A Generative Model to Synthesize EEG Data for Epileptic Seizure Prediction, 11.

16. Xiang Lu, Anhao Wen, Lei Sun; Hao Wang, Yinjing Guo, Yande Ren (2023). An Epileptic Seizure Prediction Method Based on CBAM-3D CNN-LSTM Model, 9.

17. Burak Gözütok, Ahmet Ademoülu (2021). Epileptic Seizure Prediction Using Convolutional Autoencoder Based Deep Learning,8.

18. Ahmed M. Abdelhameed, Magdy Bayoumi (2018). Semi-Supervised Deep Learning System for Epileptic Seizures Onset Prediction, 6.

19. Varsha K. Harpale; Vinayak K. Bairagi (2016). Time and frequency domain analysis of EEG signals for seizure detection: A review, 9.

20. Khansa Rasheed, Adnan Qayyum, Junaid Qadir, ShobiSivathamboo, Patrick Kwan, Levin Kuhlmann, TerenceO'Brien , and Adeel Razi. Machine Learning for Predicting Epileptic Seizures Using EEG Signals: A Review,11.

IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received November 7, 2021, accepted December 1, 2021, date of publication December 8, 2021, date of current version December 20, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3134166

Optimizing Seizure Prediction From Reduced Scalp EEG Channels Based on Spectral Features and MAML

ANIBAL ROMNEY[1], (Member, IEEE), AND VIDYA MANIAN[2], (Member, IEEE)
[1]Department of Bioengineering, University of Puerto Rico at Mayagüez, Mayagüez, PR 00682, USA
[2]Department of Electrical and Computer Engineering, University of Puerto Rico at Mayagüez, Mayagüez, PR 00682, USA
Corresponding author: Anibal Romney (anibal.romney@upr.edu)

ABSTRACT Epilepsy is a severe neurological disease with high prevalence and morbidity worldwide. The unpredictability of seizures prevents physicians from tailoring drugs and therapies. Recent non-invasive seizure prediction research has not improved the overall quality of life for patients. Therefore, new research studies on seizure prediction must integrate data, embedded devices, and algorithms. For a seizure prediction system to emerge as a feasible solution, we must address a reduction in EEG scalp electrode channels, along with a decrease in computational resources to train the time-series signal. In this work, we propose an optimized patient-specific channel reduction for seizure prediction using Model Agnostic Meta-Learning (MAML) applied to a Deep Neural Network (DNN). We selected and optimized the number of channels from each of the 23 subjects of the CHB-MIT Dataset. The feature vectors are extracted using Ensemble Empirical Mode Decomposition (EEMD) and Sequential Feature Selection (SFS). We implemented the MAML model to classify the small EEG data generated from the reduced number of subject-dependent electrodes. The experiment results yield an average sensitivity and specificity of 91% and 90%, respectively. Our study demonstrates that MAML is a promising approach to learn EEG patterns to predict epileptic seizures with few EEG scalp electrodes.

INDEX TERMS Seizure prediction, channels reduction, scalp EEG, preictal state, MAML.

I. INTRODUCTION

Epilepsy is a severe neurological disorder whose central aspect is the recurrence of seizure episodes [1]–[3]. The unpredictability of seizures and the accompanying symptoms, including unusual behavior, sudden falling, jerking movements, and altered consciousness [4], [5], make the condition unbearable for the patient and difficult to manage. Seizures are triggered by the synchronous activation of millions of neurons that generate an action potential or spike that propagates partially or entirely through the brain's cortex [6]. The epileptiform generated during seizure onset is recorded by an electroencephalogram (EEG), a gold standard in the field of neuroscience. This signal has four stages: interictal, preictal, ictal, and postictal [8]. Surgery, antiepileptic drugs, and vagus nerve stimulation are the leading therapies for epileptic seizures [7]–[9]. Physicians currently prescribe neuro-stimulation therapies and drug regimens without a monitoring signal to modulate or adapt treatment in response to physiological changes during seizure onset [10]. An algorithm that overcomes the unpredictable aspect of seizures in real-time would allow the physician to design a therapeutic and pharmacological solution tailored to the patient [11]. Regardless of the advances in non-invasive seizure prediction over the past decades, a real-time seizure prediction that would significantly impact patients' quality of life has yet to be accomplished [12]–[17]. For a real-time wearable seizure prediction system to become a realistic alternative for patients, the number of scalp electrode channels must be reduced to obtain better patient handling and acceptance while decreasing preprocessing and algorithm training. Also, the number of channel electrodes should be patient-specific based on the type and location of seizure. Some of the most recent research on seizure prediction uses patient-specific modeling but with a global EEG electrodes selection [18]–[20]. The patient-specific channel reduction would generate a short data set for training and testing. The model-agnostic meta-learning (MAML)

The associate editor coordinating the review of this manuscript and approving it for publication was Cesar Vargas-Rosales.

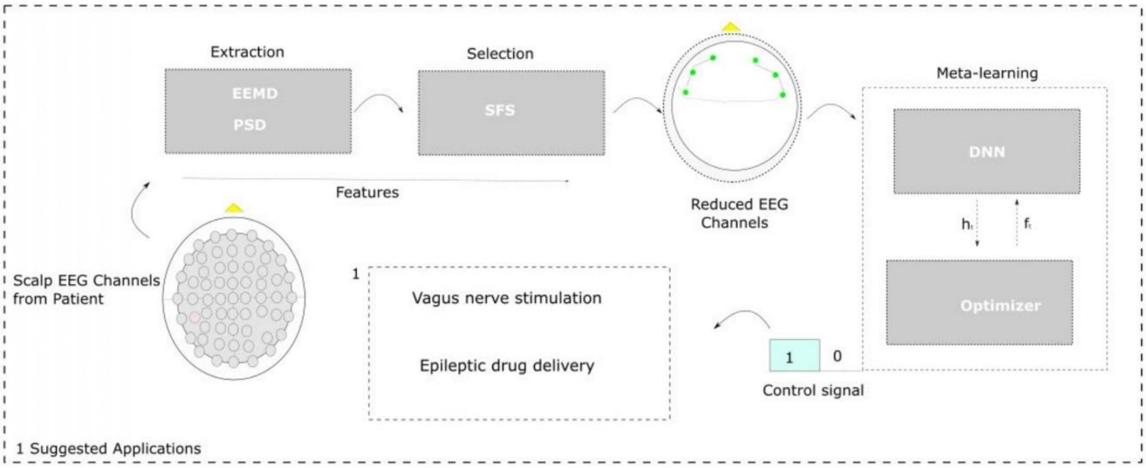


FIGURE 1. **The proposed model implemented on each of the 23 patient EEG recordings.**

algorithm works effectively on small data, although it has not yet been implemented in seizure prediction. The most advanced machine learning methods, including deep neural networks, have helped improve the learning process, but with little success in training small data sets [21], [22], [16]. To tackle all constraints mentioned above, we implemented patient-specific channel reduction, feature-engineering, and MAML algorithm for a high sensitivity and specificity seizure prediction. We use MAML to train a model on various training activities to solve new learning tasks using only a few training samples [16], [23]. Figure 1 shows the proposed general model. The MAML algorithm is applied to a DNN to train an optimized number of personalized electrode-channel features from each subject to learn preictal signatures.

This study's main contributions are summarized as follows:

- This work presents the first seizure prediction model on patient-specific electrode channels trained with a MAML algorithm on a deep neural network.
- This model is suitable for a wearable seizure prediction device on a patient-specific electrode channels selection.
- We improve seizure prediction portability by combining a personalized electrode channels and a low data algorithm applied on each patient.

We organize the remaining sections of this work as follows. Section II provides related work on seizure prediction. In Section III, we present the methodology. Section IV provides results and discussion. Finally, conclusions are given in Section V.

II. EPILEPTIC SEIZURE PREDICTION

Many researchers have addressed the problem of seizure prediction over the past four decades. Many authors have developed multiple feature extraction and classification methods and applied them to the epileptiform EEG time-series signal. However, a great interest in seizure prediction has increased in recent years because of new mathematical analyses, modern machine learning algorithms, and a better understanding of the preictal state in the scalp EEG signal. Currently, the most proposed seizure prediction methods based on machine learning include Support Vector Machine (SVM), Bayesian Gaussian, Random Forest, Logistic Classifier, and XGBoost [24], [25], [10]. Support Vector Machine has demonstrated superior success with high sensitivity. Al Ghayab *et al.* [20] proposed simple random sampling (SRS) techniques to extract features from the time domain. The least-square SVM classifier used the selected six features to predict between healthy and epileptic subjects with 96.62% sensitivity. Hosseini *et al.* [26] reported a cloud-based pervasive data collection for automatic and real-time seizure detection with a wavelet transform to extract features from the frequency bands. Contrary to the previous work, they developed multiple algorithms based on ensemble learning and randomness and achieved better predictions using a SVM. The cross-validation and sensitivity of the proposed method yielded 95% and 94%, respectively. These results are remarkable but have not yet been applied to reduced electrode channels on patient-specific data.

Algorithms based on neural networks have recently gained ground in seizure prediction [9], [18]. The most widely used of these is the convolutional neural network. Nejedly *et al.* [27] developed a CNN for automated artifact detection. The method provides independent detections for each separate channel and generates an artifact probability matrix (APM). A more improved CNN application in seizure prediction is presented in Eberlein *et al.* [28]. Unlike the previous method, binary classification is conducted without handcrafted feature extraction. CNN is used for unsupervised

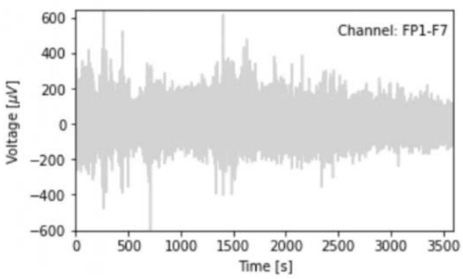


FIGURE 2. Sample EEG signal for subject S01.

feature extraction. A similar approach is used with the deep learning prediction model in Daoud and Bayoumi [9], where the raw data is directly fed after segmentation to the Multi-layer Perception to classify between preictal and interictal.

Meta-learning can optimize deep neural networks performance. The concept behind meta-learning is to learn a general-purpose algorithm by generalizing across tasks and allowing each new task to be learned better than the previous one. The concept behind meta-learning is to learn a general purpose algorithm by generalizing across tasks and allowing each new task to be learned better than the previous one [29], [30]. Meta-learning represents the process of generalized learning, in which the learning is done with few examples, just as humans learn new concepts and skills faster and efficiently. Finn *et al.* [31] proposed an algorithm for meta-learning that is model-agnostic and compatible with any gradient-trained model. In this study, the model parameters are explicitly trained, so a few gradient steps with a small training data set would produce a good generalization performance on a new task. Sucholutsky and Schonlau [32] presented a one-shot learning, where the model must learn a new class from one example. The model must learn N new classes given only $M < N$ examples. They generalize the nearest neighbor classifier soft label k to implement the algorithm in different learning scenarios.

A common limitation of previous methods is the lack of a patient-specific EEG electrode selection design, where the number and location of EEG electrodes may differ for each patient. Also, most methods elude a meta-learning approach to cope with a low patient-specific data set. Other aspects of improving seizure prediction research are maximizing sensitivity, specificity, false-positive rate, and extending the prediction horizon time.

III. MATERIALS AND METHODS

The EEG dataset from the Children's Hospital Boston and the Massachusetts Institute of Technology [33] is used to validate the proposed method. The dataset consists of EEG recordings from pediatric subjects with intractable seizures. The international 10–20 electrode positions system is used in the recordings. Table 1 shows the 23 channels and their location in the scalp EEG from recording using European Data Format (EDF) files. Table 2 shows the CHB-MIT dataset, collected from 23 subjects (5 males, ages 3–22 and 17 females, ages 1.5– 19). The sampling rate of the recording is 256 Hz. We conduct preprocessing, feature extraction, feature selection, and seizure prediction from each subject's raw EEG signal.

TABLE 1. 23 scalp EEG channels in the CHB-MIT recordings.

1	2	3	4	5	6
FP1-F7	F7-T7	T7-P7	P7-O1	FP1-F3	F3-C3
7	8	9	10	11	12
C3-P3	P3-O1	FP2-F4	F4-C4	C4-P4	P4-O2
13	14	15	16	17	18
FP2-F8	F8-T8	T8-P8	P8-O2	FZ-CZ	CZ-PZ
19	20	21	22	23	
P7-T7	T7-FT9	FT9-FT10	FT10-T8	T8-P8	

A. PREPROCESSING

We apply to the time series EEG signal a 6th order band-pass Butterworth filter ranging from 2.5 Hz to 40 Hz to remove power harmonics, noise, and muscle artifacts. Figure 1 shows a sample EEG signal from channel FP1-F7 of subject S01. The amplitude is in μV and the time in seconds.

B. FEATURES EXTRACTION AND SELECTION

We extract features using two algorithms with demonstrated success in biomedical signal processing [34]: the empirical mode decomposition (EMD) and the power spectral density (PSD). The EMD decomposes the data x (t) into intrinsic mode functions (IMFs) represented by cj and the residue (r) of the data after the last component. The original equation introduced by Huang *et al.* [35] is shown below:

$$\boldsymbol{x}\left(\boldsymbol{t}\right)=\sum\nolimits_{j=1}^{n}\boldsymbol{c_j}+\boldsymbol{r_n}$$

An improvement over the original EMD is the ensemble empirical mode decomposition (EEMD) proposed by Wu and Huang [36]. EEMD uses the white noise characteristics to perturb the signal in its true solution neighborhood and cancel itself out once it has served its purpose. The algorithm adds a different set of white noise to the signal in several iterations, allowing better scale separation. Algorithm 1 formulates the ensemble EMD as follows:

The spectrum reveals important aspects of a time series signal [25]. The PSD is defined as the discrete–time Fourier transform of the covariance sequence:

$$\varphi\left(\omega\right)=\sum_{k=-\infty}^{\infty}r\left(k\right)e^{-iwk}$$

Algorithm 1 : Ensemble Empirical Mode Decomposition

1) add a white noise series to the targeted data;
2) decompose the data with added white noise into IMFs;
3) repeat step 1 and step 2 again and again, but with different white noise series each time; and
4) obtain the (ensemble) means of corresponding IMFs of the decompositions as the result.

TABLE 2. CHB-MIT dataset.

Subject	Gender	Age	Total Seizures	EDF Files
S_{01}	F	11	7	43
S_{02}	M	11	3	35
S_{03}	F	14	7	38
S_{04}	M	22	3	43
S_{05}	F	7	5	3
S_{06}	F	1.5	7	24
S_{07}	F	14.5	3	19
S_{08}	M	3.5	5	29
S_{09}	F	10	3	19
S_{10}	M	3	7	33
S_{11}	F	12	3	38
S_{12}	F	2	13	42
S_{13}	F	3	8	42
S_{14}	F	9	7	32
S_{15}	M	16	14	53
S_{16}	F	7	5	19
S_{17}	F	12	3	20
S_{18}	F	18	6	36
S_{19}	F	19	3	30
S_{20}	F	6	6	29
S_{21}	F	13	4	33
S_{22}	F	9	3	34
S_{23}	F	6	3	9

We can recover $r(k)$ from a given $\varphi(\omega)$ as

$$r(k) = \frac{1}{2\pi}\int_{-\pi}^{\pi} \varphi(\omega)\, e^{-i\omega k} d\omega$$

We implement feature selection based on the sequential forward selection (SFS) algorithm, a wrapper-based feature selection method, which means it uses a machine-learning algorithm to select a subset of the most relevant attributes (features) [37]. For this work, we used a linear discriminant analysis (LDA) with the SFS.

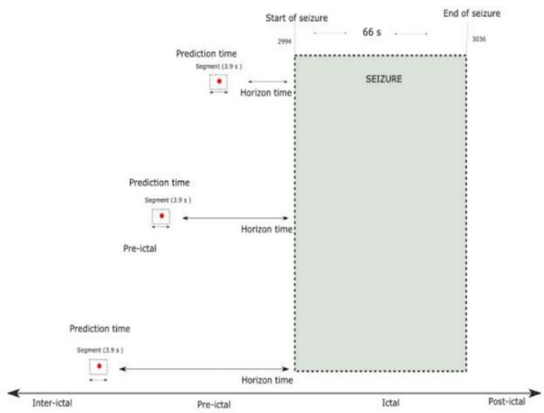


FIGURE 3. Prediction and horizon times for subject S01.

TABLE 3. Performance metrics in seizure prediction.

Measures	Computation
Accuracy-The number of correct predictions from all predictions made.	$\frac{TP+TN}{TP+TN+FP+FN}$
Sensitivity- True positive rate (TPR) of a test.	$\frac{TP}{TP+FN}$
Specificity- True negative rate (TNR) of a test	$\frac{TN}{TN+FP}$
False positive rate (FPR)	$\frac{FP}{FP+TN}$
False positive rate per hour (FPR/h).	(FPR/h) is calculated in the horizon time in the ictal transition of the epileptic EEG signal
Receiver operating characteristic (ROC).	ROC is a plot of TPR (sensitivity) against FPR (1-specificity)
Where TP =True Positive, TN =True Negative, FN = False Negative, FP = False Positive	

C. ELECTRODE-CHANNELS SELECTION

For each patient's recordings, the most informative and relevant electrodes are selected. Intrinsic mode functions (IMFs) are extracted from the 23 original channels using EEMD. This vector of features is then given as input to the SFS algorithm for feature selection. SFS generates a vector (V) with all 23 channels and their relevance scores sorted in descending order. From this channel vector (V), the two highest scored electrode channels are chosen ($e = 2$). This number is selected as the initial value since the minimum number of channels

Algorithm 2 : Model-Agnostic Learning for Fast Adaptation of Deep Networks

Given: $\rho(\tau)$: distribution over tasks
Given: α, β: step size hyperparameters
(1) randomly initialize θ
(2) **while** not done do
(3) Sample batch of tasks $\tau_i \sim \rho(\tau)$
(4) **For** all τ_i **do**
(5) Evaluate $\nabla_\theta \pounds\tau_i(f_\theta)$ with respect
(6) to K examples
(7) Compute adapted parameters
(8) with gradient descent:
(9) $\theta_i' = \theta - \alpha \nabla_\theta \pounds\tau_i(f_\theta)$
(10) **end for**

reported in the literature to support seizure prediction are two and three electrodes [38], [39]. The signal from each selected channel is discomposed to its IMFs using EEMD. The average PSD is computed for each corresponding channel IMF. These selected features are the inputs to the DNN meta-learning model. The optimal number of electrodes is attained when (sensitivity, specificity > 90%); otherwise, we increase by one the number of channels with the highest score to be used in the next iteration. These steps are repeated for each patient dataset.

A model is specifically trained with each subject's recoding. The duration of a seizure varies from 10 s to 63 s. The seizure prediction horizon (SPH) is the time between the alarm and the seizure onset, as defined in [25]. We are using three SPHs of 23 min, 10 min, and 5 min, as shown in Figure 3. In each prediction horizon, we are using two overlapping windows of 3.9 seconds.

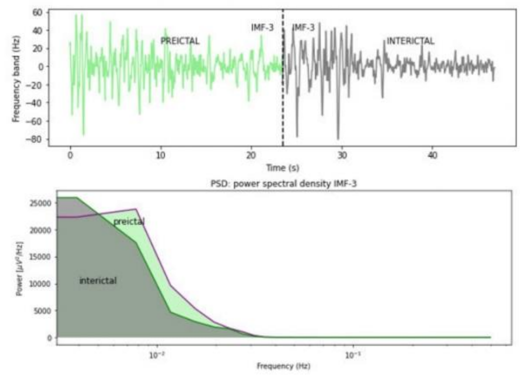
D. MAML TRAINING

We are using Model-Agnostic Meta-Learning on DNN to train the initial parameters of the model for each patient to maximize the seizure prediction performance. The parameters are updated through gradient steps computed with a small amount of data from the new task. Algorithm 2 shows the model-agnostic method [31].

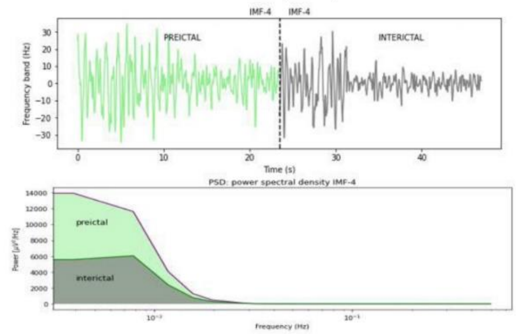
A DNN with eight hidden layers is designed. We define the sigmoid activation function for output layers and the RELU activation function for input layers. The model runs with a 10% batch size and 100 epochs. We apply the Model-Agnostic algorithm to the DNN for seizure prediction in each patient data set. The feature vector is split 70% for training, and 30% for testing. All patient-specific implementation is conducted using the Python programming language.

E. PREDICTION PERFORMANCE METRICS

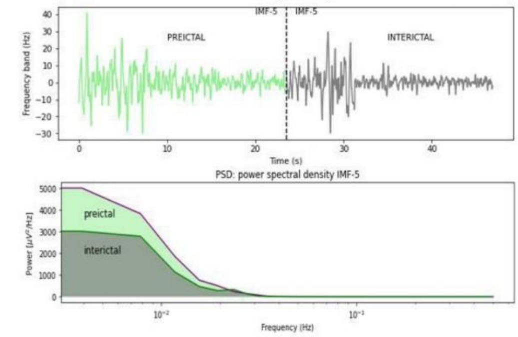
Accuracy measure falls short to determine the performance of the seizure prediction model; hence, four additional indicators (sensitivity, specificity, AUC-ROC, and false positive rate per hour) are employed to evaluate the performance of the prediction model [10], [40], which are calculated in Table 3.



A. Intrinsic Mode Function 3 (IMF-3) and its PSD.



B. Intrinsic Mode Function 4 (IMF-4) and its PSD.

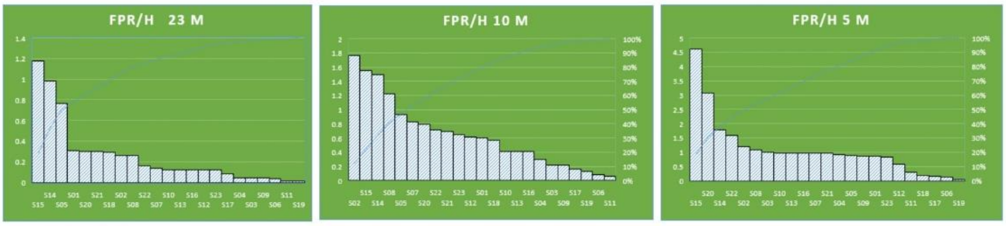


C. Intrinsic Mode Function 5 (IMF-5) and its PSD.

FIGURE 4. PSD of the intrinsic mode functions during the preictal and interictal transitions for subject S01. A) IMF-3 and its PSD, B) IMF-4 and its PSD, and C) IMF-5 and its PSD.

IV. RESULTS AND DISCUSSION

For each subject trial EEG signal, the intrinsic mode functions (IMFs) and their corresponding power spectra density are extracted in the preictal and interictal states of the epileptiform. Figure 4 shows the extracted IMF_3, IMF_4, and IMF_5 using the EEMD algorithm in the preictal and interictal segments of subject S01. Table 4 and Table 5 show a higher



A. FPR/h for 23 minutes horizon time B. FPR/h for 10 minutes horizon time C. FPR/h for 5 minutes horizon time

FIGURE 9. False positive rate per hour for each of the 23 subjects in different horizon times. A) 23 minutes, B) 10 minutes, and C) 5 minutes.

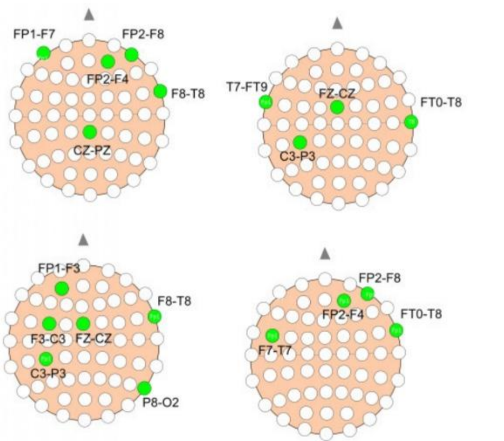


FIGURE 5. Optimal number of scalp EEG channel electrodes for subject S_{01}, S_{02}, S_{03} and S_{04}.

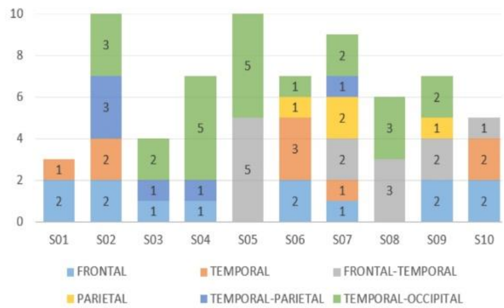


FIGURE 6. Channel location and distribution per brain lobes area.

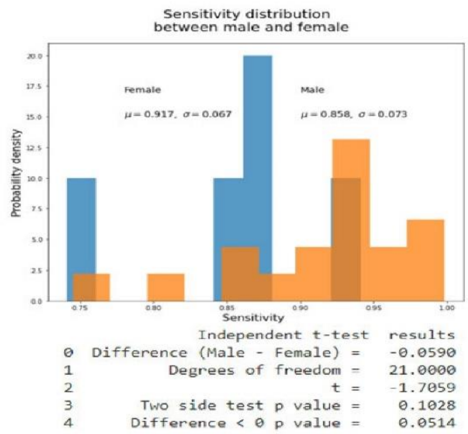


FIGURE 7. Sensitivity distribution between male and female subjects and an independent t-test result.

TABLE 4. Extracted features during the preictal frame of subject S_{01}.

Extracted Features	IMF_3	IMF_4	IMF_5
Maximum	57.226	34.568	40.408
Minimum	-75.928	-35.004	-29.250
Mean	-0.089	0.117	0.057
Standard deviation	16.588	11.396	7.218
Coefficient of Var.	-18551.2	9671.6	12452.7
Avg PSD(V^2/Hz)	548.246	269.260	106.668

average PSD of the oscillatory IMFs in the preictal than the corresponding interictal state.

The power spectral density of the intrinsic mode functions IMF_3, IMF_4, and IMF_5 extracted from the preictal and interictal segments is computed for each of the three prediction horizon times. Figure 4 shows a higher power density of the oscillatory components of the preictal state of the IMF_5.

These signature features contribute to a reduced patient-specific set of electrodes spread in four lobes (frontal, temporal, parietal, and occipital), as shown in Figure 6. The optimal number of electrodes is determined by a threshold sensitivity of 0.90. Table 6 and Figure 5 show the set of electrodes and its EEG scalp location for each subject according to the international 10–20 electrode position system. Sequential feature selection, a wrapped feature selection algorithm, is implemented for ranking the relevance of each electrode to yield the highest sensitivity when given as input to the MAML learning

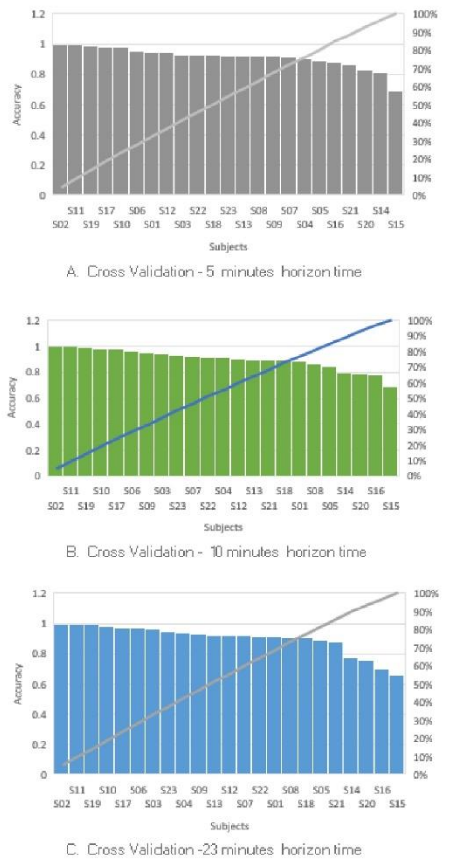


FIGURE 8. 10-fold cross-validation average score for each of the 23 subjects in the a) 5 horizon time, b) 10 horizon time, and c) 23 horizon time.

TABLE 5. Extracted features during the interictal frame of subject S_{01}.

Extracted Features	IMF_3	IMF_4	IMF_5
Maximum	45.972	30.850	29.906
Minimum	-81.192	-30.450	-29.639
Mean	0.52	-0.188	-0.118
Standard deviation	15.365	8.219	5.794
Coefficient of Var.	2923.1	-4349.2	-4900.4
Avg. PSD(V^2/Hz)	472.964	126.821	67.676

framework. Clearly, from Table 6, a specific reduced number of electrodes is attained for all 23 cases, in which the SFS algorithm is applied to the intrinsic functions extracted from the subject's 23 channels. The highest number of electrodes per subject is six, and the lowest number is two, representing a promising reduction scheme for wearable real-time seizure prediction. The MAML algorithm weight optimization process successfully learned the signature patterns of the IMFs and their PSD in the preictal and interictal transitions after 10 epochs, rendering a high optimized model for the 5 min, 23 min, and 30 min horizon times.

TABLE 6. Selected electrodes per subject.

Subject	Gender	No. of channels	Channels
S_{01}	M	5	FP1F7, F8T8, FP2F8, FP2F4, CZPZ
S_{02}	F	4	T7FT9, FZCZ, FT0T8, F7T7
S_{03}	M	6	FP1F3, F8T8, F3C3, FZCZ, C3P3, P8O2
S_{04}	F	4	FTT7, FP2F4, FP2F8, FT0T8
S_{05}	M	3	C3P3, P3O1, FP2F4
S_{06}	F	5	FP2F4, F4C4, C4P4, P4O2, FP2F8
S_{07}	M	3	C3P3, P3O1, FP2F4
S_{08}	F	5	FP2F4, F4C4, C4P4, P4O2, FP2F8
S_{09}	M	3	C3P3, P3O1, FP2F4
S_{10}	F	5	FP2F4, F4C4, C4P4, P4O2, FP2F8
S_{11}	M	5	F7T7, FP1F3, F4C4, P4O2, T8P8
S_{12}	F	4	F7T7, P4O2, T8P8, FT9FT10
S_{13}	M	4	FP1F7, F8T8, T8P8, P8O2
S_{14}	F	3	F7T7, T7P7, FT10T8
S_{15}	M	2	FP1F7, T7FT9
S_{16}	F	5	FP2F4, F4C4, C4P4, P4O2, FP2F8
S_{17}	M	4	P7O1, FP2F4, P4O2, F8T8
S_{18}	F	4	P7O1, FP2F4, P4O2, F8T8
S_{19}	M	6	F3C3, C3P3, FP2F4, C4P4, F8T8, FT9FT10
S_{20}	F	3	FP1F7, P8O2, T7FT9
S_{21}	M	4	P3O1, C4P4, P7T7, FT9FT10
S_{22}	F	5	FP1F7, F8T8, T8P8, P8O2, T7FT9
S_{23}	M	4	P3O1, FZCZ, P7T7, FT9FT10

To evaluate the classification performance of the MAML prediction model, we applied the measures of sensitivity, specificity, and the area under the ROC curve (AUC) to each of the 23 subjects testing data, as shown in Table 7. The performance of the MAML model tested in the female subjects scored similar sensitivity to that of the male subjects, as shown in Figure 7, and there is no significant difference in the gender groups since the p-value is not less than 0.05. The difference in the power spectral density of *IMFs* in the preictal and interictal segments is sustained across the 23 subjects with varying power levels and frequency distribution. In the training phase, these patterns are learned by the MAML-DNN combination to build a MAML model, which achieves an average sensitivity and specificity of 91% and 90%, respectively, in the testing phase.

A consistent accuracy score is established with the 10-fold cross-validation across all three horizon times. Figure 8 shows the 10-fold cross-validation average score for each subject in the 23 horizon time. The MAML prediction algorithm uses the second derivative to update the weight of the DNN. Once the model is trained and updated, a fast execution time in the testing phase is obtained. The false positive

TABLE 7. Prediction performance in three horizons.

	5 Min Horizon				10 Min Horizon				23 Min Horizon			
Subject	Sen	Spec	AUC	F1	Sen	Spec	AUC	F1	Sen	Spec	AUC	F1
S01	0.938	0.929	0.946	0.938	0.903	0.899	0.907	0.902	0.865	0.884	0.824	0.813
S02	0.870	0.900	0.785	0.757	0.913	0.706	0.810	0.827	0.890	0.900	0.785	0.757
S03	0.941	0.918	0.930	0.930	0.930	0.964	0.947	0.946	0.940	0.985	0.963	0.962
S04	0.857	0.924	0.891	0.887	0.906	0.951	0.928	0.926	0.896	0.983	0.940	0.937
S05	0.853	0.927	0.890	0.885	0.922	0.846	0.884	0.888	0.963	0.709	0.836	0.854
S06	0.940	0.989	0.965	0.964	0.939	0.986	0.963	0.962	0.939	0.986	0.963	0.962
S07	0.939	0.920	0.929	0.929	0.933	0.863	0.898	0.901	0.895	0.947	0.921	0.919
S08	0.880	0.910	0.785	0.757	0.913	0.796	0.810	0.827	0.890	0.900	0.785	0.757
S09	0.910	0.928	0.930	0.930	0.920	0.964	0.947	0.946	0.940	0.985	0.963	0.962
S10	0.941	0.919	0.930	0.931	0.904	0.932	0.918	0.916	0.938	0.953	0.945	0.945
S11	0.992	0.976	0.984	0.984	0.995	0.990	0.992	0.992	1.00	0.995	0.995	0.995
S12	0.918	0.951	0.934	0.933	0.942	0.898	0.920	0.921	0.872	0.955	0.914	0.910
S13	0.941	0.919	0.930	0.931	0.904	0.932	0.918	0.916	0.938	0.953	0.945	0.945
S14	0.806	0.852	0.829	0.825	0.872	0.751	0.811	0.822	0.900	0.627	0.763	0.791
S15	0.741	0.615	0.678	0.696	0.638	0.742	0.690	0.672	0.792	0.554	0.673	0.707
S16	0.941	0.919	0.930	0.931	0.904	0.932	0.918	0.916	0.938	0.953	0.945	0.945
S17	0.988	0.986	0.987	0.987	0.979	0.973	0.976	0.976	0.984	0.970	0.977	0.977
S18	0.745	0.985	0.864	0.845	0.965	0.905	0.935	0.937	0.982	0.889	0.936	0.938
S19	0.998	0.996	0.997	0.997	0.995	0.978	0.987	0.987	0.981	0.995	0.988	0.988
S20	0.876	0.745	0.811	0.820	0.771	0.868	0.820	0.809	0.557	0.885	0.721	0.665
S21	0.849	0.920	0.885	0.880	0.896	0.884	0.890	0.890	0.883	0.887	0.885	0.884
S22	0.961	0.867	0.914	0.917	0.951	0.881	0.916	0.918	0.938	0.940	0.939	0.938
S23	0.967	0.932	0.949	0.950	0.962	0.892	0.927	0.929	0.961	0.955	0.958	0.958

TABLE 8. Comparison with state-of-the-art epileptic seizure prediction methods.

Year	Authors	Classifying Algorithm	Dataset	Number of Channels	Patient-Specific Channels	Sen (%)	FPR (/h)	Horizon Time (min)
2017	Prathap *et al.* [41]	SPARSE	CHB	23	N0	86.11	-	-
2016	Cho *et al.* [38]	SVM	CHB	3	No	82.44	-	5
2020	Zhang *et. al.* [21]	CNN	CHB	18	No	92.2	0.12	30
2018	Truong *et al.* [42]	CNN	CHB	22	No	81.2	0.16	5
2017	Alotaiby *et al.* [43]	LDA	CHB	23	No	81	0.47	60
2020	Romney *et al.* [39]	DNN	CHB	2	No	86.7	0.27	23
2017	Chu *et al.* [44]	ATTRACTOR	CHB	-	No	86.77	0.367	55.3
2017	Birjandtalab *et al.* [45]	KNN	CHB	3	No	80.87	2.5	-
-	*Proposed Method*	MAML	CHB	(2,3,4,5,6,7)	Yes	92.31	0.26	23

rate per hour (FPR/h) for the three horizon times is shown in Figure 9. Most of the electrodes are located predominantly in the temporal lobes, as shown in Figure 6, which is consistent with the most frequent seizure types documented in the neurological literature. Electrodes located in the frontal-temporal brain area are the most significant for seizure prediction. The comparative study shown in Table 8 includes all the recent works implemented with reduced electrodes using the CHB-MIT database. The comparison shows that the proposed model with MAML surpasses similar studies in sensitivity and false positive rate performance measures. The proposed method yielded an average FPR/h of 0.26, which is among the lowest number obtained in recent seizure prediction research with any algorithm.

V. CONCLUSION

In this work, we present a MAML seizure prediction method with a patient-specific electrode channels selection. Our approach reduced the number of electrodes arranged on the scalp for each subject to pave the way toward real-time seizure prediction with high sensitivity and specificity. The prediction model is built on patient-specific data using the model-agnostic meta-learning algorithm. The individual performance was validated using 10-F cross-validation. The reduced electrode-channel selection is sufficiently sensitive to capture or exclude patterns that lead to seizure prediction with varying horizon times. Patient-specific electrode reduction allowed us to train individual model for the subject's unique epileptiform EEG signals. Although the results look promising for wearable seizure prediction, the reduction of EEG channels should not be the preferred means to optimize EEG application in hospitals where all electrodes are required for accurate diagnosis. The performance measures of sensitivity, specificity, and FPR/h of the proposed model are encouraging. However, additional work is needed to implement the MAML model in a miniaturized device to test the sampled signal from the wireless scalp electrodes in real-time.

REFERENCES

[1] K. M. Fiest, K. M. Sauro, S. Wiebe, S. B. Patten, C.-S. Kwon, J. Dykeman, T. Pringsheim, D. L. Lorenzetti, and N. Jetté, "Prevalence and incidence of epilepsy," *Neurology*, vol. 88, no. 3, pp. 296–303, Jan. 2017, doi: 10.1212/WNL.0000000000003509.

[2] *The Economic Burden of Epilepsy in Australia, 2019–2020 | Epilepsy Foundation*. Accessed: Aug. 6, 2020. [Online]. Available: https://epilepsyfoundation.org.au/2020/02/08/the-economic-burden-of-epilepsy-in-australia-2019-2020/

[3] J. I. Sirven, "Epilepsy: A spectrum disorder," *Cold Spring Harbor Perspect. Med.*, vol. 5, no. 9, Sep. 2015, Art. no. a022848, doi: 10.1101/cshperspect.a022848.

[4] H. Khan, L. Marcuse, M. Fields, K. Swann, and B. Yener, "Focal onset seizure prediction using convolutional networks," *IEEE Trans. Biomed. Eng.*, vol. 65, no. 9, pp. 2109–2118, Sep. 2018, doi: 10.1109/TBME.2017.2785401.

[5] M. Zhou, C. Tian, R. Cao, B. Wang, Y. Niu, T. Hu, H. Guo, and J. Xiang, "Epileptic seizure detection based on EEG signals and CNN," *Frontiers Neuroinform.*, vol. 12, p. 95, Aug. 2018, doi: 10.3389/fninf.2018.00095.

[6] T. N. Alotaiby, S. A. Alshebeili, T. Alshawi, I. Ahmad, and F. E. A. El-Samie, "EEG seizure detection and prediction algorithms: A survey," *EURASIP J. Adv. Signal Process.*, vol. 2014, no. 1, pp. 1–21, Dec. 2014, doi: 10.1186/1687-6180-2014-183.

[7] P. A. Abhang, B. W. Gawali, and S. C. Mehrotra, "Technological basics of EEG recording and operation of apparatus," in *Introduction to EEG- and Speech-Based Emotion Recognition*. New York, NY, USA: Academic, 2016, pp. 19–50, doi: 10.1016/b978-0-12-804490-2.00002-6.

[8] V. Senger and R. Tetzlaff, "Cellular nonlinear network-based signal prediction in epilepsy: Method comparison," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, May 2015, pp. 397–400, doi: 10.1109/ISCAS.2015.7168654.

[9] H. Daoud and M. A. Bayoumi, "Efficient epileptic seizure prediction based on deep learning," *IEEE Trans. Biomed. Circuits Syst.*, vol. 13, no. 5, pp. 804–813, Oct. 2019, doi: 10.1109/TBCAS.2019.2929053.

[10] F. Samie, S. Paul, L. Bauer, and J. Henkel, "Highly efficient and accurate seizure prediction on constrained IoT devices," in *Proc. Design Autom. Test Eur. Conf. Exhib. DATE*, Jan. 2018, pp. 955–960, doi: 10.23919/DATE.2018.8342147.

[11] A. Kumar and S. Sharma, *Seizure, Complex Partial*. Treasure Island, FL, USA: StatPearls Publishing, 2018.

[12] C. Baumgartner, J. P. Koren, and M. Rothmayer, "Automatic computer-based detection of epileptic seizures," *Frontiers Neurol.*, vol. 9, p. 639, Aug. 2018, doi: 10.3389/fneur.2018.00639.

[13] A. H. Osman and A. A. Alzahrani, "New approach for automated epileptic disease diagnosis using an integrated self-organization map and radial basis function neural network algorithm," *IEEE Access*, vol. 7, pp. 4741–4747, 2019, doi: 10.1109/ACCESS.2018.2886608.

[14] E. B. Geller, T. L. Skarpaas, R. E. Gross, R. R. Goodman, G. L. Barkley, C. W. Bazil, M. J. Berg, G. K. Bergey, S. S. Cash, A. J. Cole, and R. B. Duckrow, "Brain-responsive neurostimulation in patients with medically intractable mesial temporal lobe epilepsy," *Epilepsia*, vol. 58, no. 6, pp. 994–1004, 2017, doi: 10.1111/epi.13740.

[15] N. Wang and M. R. Lyu, "Extracting and selecting distinctive EEG features for efficient epileptic seizure prediction," *IEEE J. Biomed. Health Inform.*, vol. 19, no. 5, pp. 1648–1659, Sep. 2015, doi: 10.1109/JBHI.2014.2358640.

[16] A. Ahmadi and H. Soltanian-Zadeh, "Epileptic seizure prediction using spectral entropy-based features of EEG," in *Proc. 4th Int. Conf. Pattern Recognit. Image Anal. (IPRIA)*, Mar. 2019, pp. 124–129, doi: 10.1109/PRIA.2019.8785984.

[17] J. G. Servin-Aguilar, L. Rizo-Dominguez, J. A. Pardinas-Mir, C. Vargas-Rosales, and I. Padilla-Cantoya, "Epilepsy seizure detection: A heavy tail approach," *IEEE Access*, vol. 8, pp. 208170–208178, 2020, doi: 10.1109/ACCESS.2020.3038397.

[18] I. Kiral-Kornek, S. Roy, E. Nurse, B. Mashford, P. Karoly, T. Carroll, D. Payne, S. Saha, S. Baldassano, T. O'Brien, and D. Grayden, "Epileptic seizure prediction using big data and deep learning: Toward a mobile system," *EBioMedicine*, vol. 27, pp. 103–111, Dec. 2017, doi: 10.1016/j.ebiom.2017.11.032.

[19] J. Kung, D. Kim, and S. Mukhopadhyay, "Adaptive precision cellular nonlinear network," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 26, no. 5, pp. 841–854, May 2018, doi: 10.1109/TVLSI.2018.2794498.

[20] H. Ghayab, Y. Li, S. Abdulla, M. Diykh, and X. Wan, "Classification of epileptic EEG signals based on simple random sampling and sequential feature selection," *Brain Informat.*, vol. 3, no. 2, pp. 85–91, 2016, doi: 10.1007/s40708-016-0039-1.

[21] Y. Zhang, Y. Guo, P. Yang, W. Chen, and B. Lo, "Epilepsy seizure prediction on EEG using common spatial pattern and convolutional neural network," *IEEE J. Biomed. Health Informat.*, vol. 24, no. 2, pp. 465–474, Feb. 2020, doi: 10.1109/JBHI.2019.2933046.

[22] A. Kumar and M. H. Kolekar, "Machine learning approach for epileptic seizure detection using wavelet analysis of EEG signals," in *Proc. Int. Conf. Med. Imag., m-Health Emerg. Commun. Syst. (MedCom)*, Nov. 2014, pp. 412–416, doi: 10.1109/MedCom.2014.7006043.

[23] A. Shahid, N. Kamel, A. S. Malik, and M. A. Jatoi, "Epileptic seizure detection using the singular values of EEG signals," in *Proc. ICME Int. Conf. Complex Med. Eng.*, May 2013, pp. 652–655, doi: 10.1109/ICCME.2013.6548330.

[24] A. S. Zandi, R. Tafreshi, M. Javidan, and G. A. Dumont, "Predicting epileptic seizures in scalp EEG based on a variational Bayesian Gaussian mixture model of zero-crossing intervals," *IEEE Trans. Biomed. Eng.*, vol. 60, no. 5, pp. 1401–1413, May 2013, doi: 10.1109/TBME.2012.2237399.

[25] Z. Zhang and K. K. Parhi, "Low-complexity seizure prediction from iEEG/sEEG using spectral power and ratios of spectral power," *IEEE Trans. Biomed. Circuits Syst.*, vol. 10, no. 3, pp. 693–706, Jun. 2016, doi: 10.1109/TBCAS.2015.2477264.

[26] M.-P. Hosseini, A. Hajisami, and D. Pompili, "Real-time epileptic seizure detection from EEG signals via random subspace ensemble learning," in *Proc. IEEE Int. Conf. Autonomic Comput. (ICAC)*, Jul. 2016, pp. 209–218, doi: 10.1109/ICAC.2016.57.

[27] P. Nejedly, J. Cimbalnik, P. Klimes, F. Plesinger, J. Halamek, V. Kremen, I. Viscor, B. H. Brinkmann, M. Pail, M. Brazdil, G. Worrell, and P. Jurak, "Intracerebral EEG artifact identification using convolutional neural networks," *Neuroinformatics*, vol. 17, no. 2, pp. 225–234, Apr. 2019, doi: 10.1007/s12021-018-9397-6.

[28] M. Eberlein, R. Hildebrand, R. Tetzlaff, N. Hoffmann, L. Kuhlmann, B. Brinkmann, and J. M'uller, "Convolutional neural networks for epileptic seizure prediction," in *Proc. IEEE Int. Conf. Bioinf. Biomed. (BIBM)*, Dec. 2018, pp. 2577–2582, doi: 10.1109/BIBM.2018.8621225.

[29] T. Hospedales, A. Antoniou, P. Micaelli, and A. Storkey, "Meta-learning in neural networks: A survey," 2020, *arXiv:2004.05439*.

[30] C. Lemke, M. Budka, and B. Gabrys, "Metalearning: A survey of trends and technologies," *Artif. Intell. Rev.*, vol. 44, no. 1, pp. 117–130, Jun. 2015, doi: 10.1007/s10462-013-9406-y.

[31] C. Finn, P. Abbeel, and S. Levine, "Model-agnostic meta-learning for fast adaptation of deep networks," 2017, *arXiv:1703.03400*.

[32] I. Sucholutsky and M. Schonlau, "'Less than one'-shot learning: Learning N classes from M<N samples," 2020, *arXiv:2009.08449*.

[33] *CHB-MIT Scalp EEG Database v1.0.0*. Accessed: Oct. 30, 2021. [Online]. Available: https://physionet.org/content/chbmit/1.0.0/

[34] J. B. Ali, N. Fnaiech, L. Saidi, B. Chebel-Morello, and F. Fnaiech, "Application of empirical mode decomposition and artificial neural network for automatic bearing fault diagnosis based on vibration signals," *Appl. Acoust.*, vol. 89, no. 3, pp. 16–27, Mar. 2015, doi: 10.1016/j.apacoust.2014.08.016.

[35] N. E. Huang, Z. Shen, S. R. Long, M. C. Wu, H. H. Shih, Q. Zheng, N. C. Yen, C. C. Tung, and H. H. Liu, "The empirical mode decomposition and the Hilbert spectrum for nonlinear and non-stationary time series analysis," *Proc. Roy. Soc. London A*, vol. 454, no. 1971, pp. 903–995, 1998, doi: 10.1098/rspa.1998.0193.

[36] Z. Wu and N. E. Huang, "Ensemble empirical mode decomposition: A noise-assisted data analysis method," *Adv. Adapt. Data Anal.*, vol. 1, no. 1, pp. 1–41, Jan. 2009, doi: 10.1142/S1793536909000047.

[37] A. Marcano-Cedeno, J. Quintanilla-Dominguez, M. G. Cortina-Januchs, and D. Andina, "Feature selection using sequential forward selection and classification applying artificial metaplasticity neural network," in *Proc. IECON 36th Annu. Conf. IEEE Ind. Electron. Soc.*, Nov. 2010, pp. 2845–2850, doi: 10.1109/IECON.2010.5675075.

[38] D. Cho, B. Min, J. Kim, and B. Lee, "EEG-based prediction of epileptic seizures using phase synchronization elicited from noise-assisted multivariate empirical mode decomposition," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 25, no. 8, pp. 1309–1318, Aug. 2017, doi: 10.1109/TNSRE.2016.2618937.

[39] A. Romney and V. Manian, "Comparison of frontal-temporal channels in epilepsy seizure prediction based on EEMD-ReliefF and DNN," *Computers*, vol. 9, no. 4, p. 78, Sep. 2020, doi: 10.3390/computers9040078.

[40] W. Zhao, W. Zhao, W. Wang, X. Jiang, X. Zhang, Y. Peng, B. Zhang, and G. Zhang, "A novel deep neural network for robust detection of seizures using EEG signals," *Comput. Math. Methods Med.*, vol. 2020, pp. 1–9, Apr. 2020, doi: 10.1155/2020/9689821.

[41] P. Prathap and T. A. Devi, "EEG spectral feature based seizure prediction using an efficient sparse classifier," in *Proc. Int. Conf. Intell. Comput., Instrum. Control Technol. (ICICICT)*, Jul. 2017, pp. 721–725, doi: 10.1109/ICICICT1.2017.8342653.

[42] N. D. Truong, A. D. Nguyen, L. Kuhlmann, M. R. Bonyadi, J. Yang, S. Ippolito, and O. Kavehei, "Convolutional neural networks for seizure prediction using intracranial and scalp electroencephalogram," *Neural Netw.*, vol. 105, pp. 104–111, Sep. 2018, doi: 10.1016/j.neunet.2018.04.018.

[43] T. N. Alotaiby, S. A. Alshebeili, F. M. Alotaibi, and S. R. Alrshoud, "Epileptic seizure prediction using CSP and LDA for scalp EEG signals," *Comput. Intell. Neurosci.*, vol. 2017, pp. 1–11, Jan. 2017, doi: 10.1155/2017/1240323.

[44] H. Chu, C. K. Chung, W. Jeong, and K.-H. Cho, "Predicting epileptic seizures from scalp EEG based on attractor state analysis," *Comput. Methods Programs Biomed.*, vol. 143, pp. 75–87, May 2017, doi: 10.1016/j.cmpb.2017.03.002.

[45] J. Birjandtalab, M. B. Pouyan, D. Cogan, M. Nourani, and J. Harvey, "Automated seizure detection using limited-channel EEG and non-linear dimension reduction," *Comput. Biol. Med.*, vol. 82, pp. 49–58, Mar. 2017, doi: 10.1016/j.compbiomed.2017.01.011.

ANIBAL ROMNEY (Member, IEEE) received the M.S. degree in electrical engineering from the University of Puerto Rico at Mayagüez (UPRM), Mayagüez, PR, USA, where he is currently pursuing the Ph.D. degree in bioengineering. In 2012, he became a Full Professor at the University of Puerto Rico at Aguadilla. His research interests include machine learning, molecular biology, neuroscience, biosensors, and robotics. In 2001, he was awarded the NSF grant in communications and information technologies. As a Co-PI of a three-year project, he has enhance the preparation of underserved students for employment in the fields of manufacturing communications and information technologies industries.

VIDYA MANIAN (Member, IEEE) received the Ph.D. degree in computing, information science, and engineering from the University of Puerto Rico at Mayagüez (UPRM), Mayagüez, PR, USA, in 2004. She was a Visiting Scholar with the Lane Department of Computer Science and Electrical Engineering, West Virginia University, Morgantown, WV, USA. In 2005, she was a Postdoctoral Associate in electrical and computer engineering (ECE) at UPRM. In 2006, she joined at UPRM, as an Assistant Professor in ECE. She is currently a Professor in ECE with UPRM, and the Director of the Brain–Computer Interface Laboratory. Her research interests include multispectral image processing, bioengineering, bio-sensory data fusion, neuroscience, and machine learning. In 2013, the First Lady of Puerto Rico recognized Dr. Manian for her outstanding contribution to education and research.