EPILEPTIC SEIZURE DETECTION

Abstract

Electroencephalography (EEG) captures electrical activity in the brain and has an important function in the diagnosis of neurological conditions, especially epilepsy. This work focuses on automating the classification of EEG signals for the detection of epileptic seizures through the utilization of signal processing, conventional machine learning, and deep learning approaches. EEG data was acquired and divided into three groups: normal, epilepsy without seizure and with seizure. The signals were preprocessed with bandpass and notch filters to eliminate noise and power-line interference. Both statistical and frequency-domain features were used in feature extraction.

Traditional machine learning models—Support Vector Machine (SVM), K-Nearest Neighbors (KNN), Logistic Regression, Random Forest and Gradient Boosting—were trained and tested. Deep learning models like 1D CNN, 2D CNN, and hybrid models like CNN + LSTM/GRU were also created. ResNet18 was also used using spectrogram images to take advantage of transfer learning. Methods like Mixup augmentation, label smoothing, and layer normalization were used to improve performance and minimize overfitting.

Evaluation was performed with metrics such as accuracy, precision, recall, F1-score, confusion matrix, and ROC curves. The highest accuracies of 99% and 98% were obtained by the deep learning models, specifically 2D CNN and ResNet18. This work illustrates the potential of deep learning in EEG-based seizure detection and validates its use in real-time health monitoring and diagnostic applications.

Table of contents

List of figures

Abbreviations

EEG - Electroencephalography

CNN - Convolutional Neural Network

LSTM - Long Short Term Memory

GRU - Gated Recurrent Unit

RNN - Recurrent Neural Network

ROC curve - Receiver Operating Characteristic curve

TP - True Positive

TN - True Negative

FP - False Positive

FN - False Negative

LLE - Largest Lyapunov Exponent

CHAPTER 1

INTRODUCTION

1.1 Background

Epilepsy is among the most prevalent neurological conditions, which involves recurring and sudden seizures caused by atypical electrical brain activity. Precise seizure detection is important for proper patient monitoring, treatment, and enhancing the quality of life. Electroencephalography (EEG) is the most applied method of recording the electrical brain activity and diagnosing epilepsy. Yet, visual examination of EEG signals is time-consuming, labor-intensive, and liable to human mistake. Thus, there is immense demand for computerized and consistent seizure detection techniques based on machine learning and deep learning methods.

1.2 Problem Statement

Conventional approaches to the detection of epileptic seizures using EEG signal analysis may be time-consuming and subjective. Late or incorrect classification of seizure events can have adverse effects on patient health. Additionally, the non-stationarity and complexity in EEG signals increase the difficulty in manual analysis. It is, therefore, imperative that an automated system capable of identifying seizure and non-seizure events be developed that provides accurate detection and classification to support neurologists and medical professionals.

1.3 Significance of Automated EEG Analysis

Automated processing of EEG signals has various benefits: accelerated diagnosis, decreased human error, scalability to big datasets, and uniformity in assessments. Machine learning models can be trained using extracted features from EEG signals, while deep learning models can learn from raw or transformed EEG data (such as spectrograms) directly without handcrafted feature engineering. This results in more reliable and efficient seizure detection systems with the potential for real-time usage.

1.4 Introduction to Machine Learning and Deep Learning Models

In this project, several machine learning models like Logistic Regression, Support Vector Machine (SVM), K-Nearest Neighbours (KNN), Random Forest, and Gradient Boosting were applied to extracted EEG features for seizure classification. Moreover, deep learning models like 1D CNN, 2D CNN, 1D CNN with LSTM and GRU, GRU, LSTM, and ResNet18 were also investigated to classify EEG signals directly. These models were selected carefully to contrast the efficiency of feature-based traditional models with automatic feature-learning models for EEG signal classification.

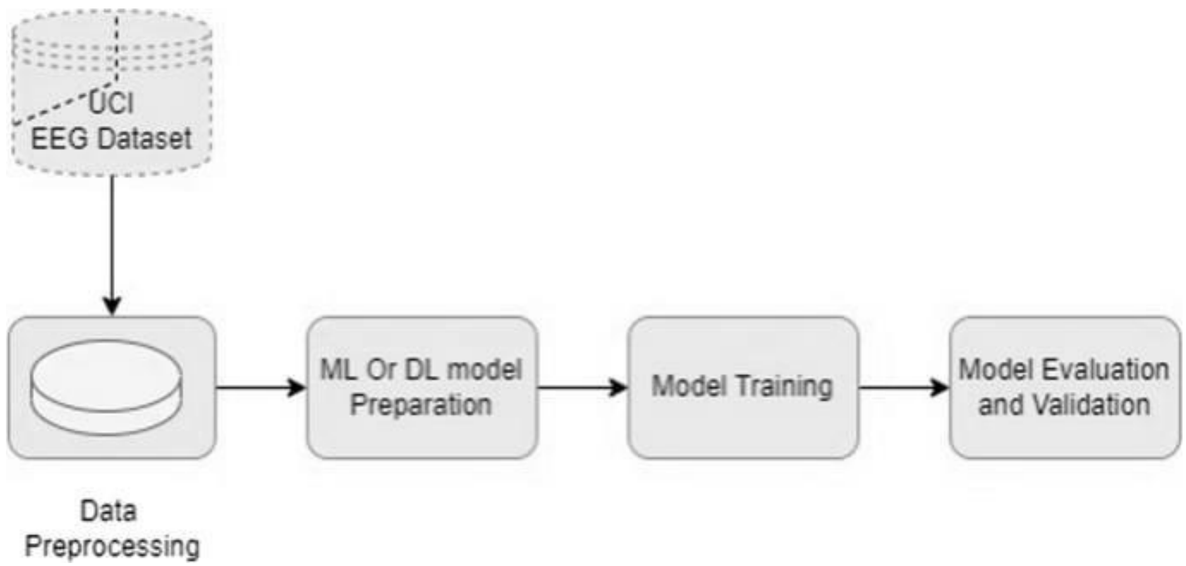

Fig 1.4.1 Workflow of Machine Learning

1.5 Motivation

The main driving force for this project is to utilize the power of machine learning and deep learning methodologies for the early and precise diagnosis of epileptic seizures. Through automation of the detection process, the workload on medical professionals can be minimized while achieving earlier and more accurate diagnosis. The project also focuses on the comparison of models and emphasizing the advantages of deep learning, particularly convolutional neural networks (CNNs), in EEG-based clinical diagnosis.

CHAPTER 2

LITERATURE SURVEY AND OBJECTIVE

Literature Survey

Epileptic seizure detection from EEG signals has been a dynamic and active research field. Several machine learning and deep learning techniques have been proposed to automate and improve seizure detection accuracy.

Siddiqui, Menendez,Huang and Hussain(2020) provided a detailed review on epileptic seizure detection with machine learning classifiers. Their study highlighted the significance of choosing proper statistical features and classifiers for precise EEG classification. They discovered that ensemble methods, especially Random Forest, tend to perform better than conventional models by efficiently managing the complexity of EEG signals.

Guerrero, Parada and Espitia (2021) explored EEG sound analysis using classical machine learning algorithms and convolutional neural networks (CNNs). Their research pointed out the capacity of CNNs to extract spatial and temporal features from EEG signals, leading to improved classification performance. They also showed that the combination of traditional methods with CNNs can achieve significant improvements in seizure detection accuracy.

Artur Gramacki and Jaroslaw Gramacki (2022) presented a deep learning framework to detect neonatal seizures through the application of convolutional neural networks. They found that appropriate preprocessing, balancing the data, and designing an optimal CNN architecture facilitated seizure episodes being detected at very high accuracy (96%–97%). They specifically highlighted the role of such methods as sliding windows to properly balance seizure and non-seizure classes.

Hepseeba Kode, Khaled Elleithy, and Laiali Almazaydeh (2024) investigated a hybrid strategy using traditional machine learning models like Random Forest and XGBoost together with deep learning methods like 1D CNN for seizure identification from EEG signals. Their result indicated that deep learning models, particularly 1D CNN models, perform significantly better than classical models by being able to catch the complex temporal patterns inherent in EEG data better.

Throughout these papers, it is repeatedly noted that deep learning models, especially CNN-based models, provide better performance than conventional machine learning models. Nevertheless, combining machine learning and deep learning models gives a solid, interpretable, and clinically useful platform for epileptic seizure detection.

Objectives

- To classify and identify epileptic seizure events and normal EEG signals employing machine learning and deep learning models.
- To utilize pre-trained deep learning frameworks like ResNet18 and design personalized models like 1D CNN, 2D CNN, and hybrid models (1D CNN + LSTM/GRU) for EEG signal classification.
- To preprocess EEG datasets efficiently employing normalization, filtering, segmentation, and feature extraction methods.
- To compare and analyse the performance of machine learning and deep learning algorithms according to accuracy, precision, recall, and F1-score measures.
- To select and suggest the most accurate and efficient model for real-time epileptic seizure detection from EEG signals.

CHAPTER-3

METHODOLOGY

3.1 MACHINE LEARNING APPROACH:

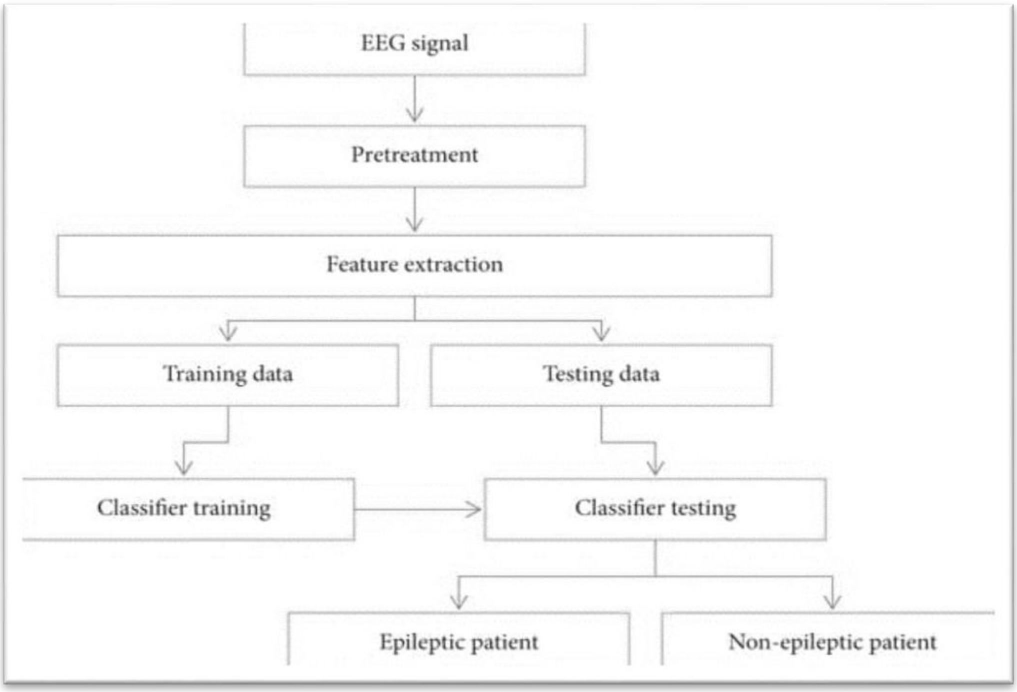

Fig 3.1.1 Block diagram of Machine Learning Approach

STEPS IN MACHINE LEARNING

3.1.1 Load Dataset

- The project begins by loading the EEG dataset into a Pandas Data Frame from Excel files. This step organizes the raw time-series data into a structured format, providing the foundation for efficient data processing and model development.
- The dataset used in this project was sourced from the Bonn University EEG Database, a widely accepted benchmark for EEG signal analysis.
- The database is divided into three primary classes:

Normal Class: EEG signals from healthy individuals (200 subjects).

Epilepsy Without Seizure Class: EEG signals recorded from epilepsy patients during non-seizure periods (200 subjects).

Epilepsy With Seizure Class: EEG signals recorded during active seizure episodes (100 subjects).

- Each subject's EEG recording consists of 4096-time samples, representing the brain's electrical activity over a period of time.

3.1.2 Data Preprocessing

Bandpass Filtering

- A Butterworth bandpass filter with a frequency range of 0.5 Hz to 50 Hz is applied to each EEG signal.
- The purpose of filtering is to remove noise and retain important brainwave frequencies, such as delta (0.5–4 Hz), theta (4–8 Hz), alpha (8–12 Hz), and beta (12–30 Hz) bands.
- A sampling frequency of 256 Hz is assumed for the recordings**.**

Normalization

- After filtering, all EEG signals are standardized using z-score normalization
- This step transforms the data to have a mean of 0 and standard deviation of 1.
- Normalization ensures that all signals are on the same scale, which improves the performance and stability of machine learning models.

Segmentation

- Each normalized EEG signal is segmented into overlapping windows for better feature extraction:
    - Window size: 512 samples (2 seconds of signal at 256 Hz)
    - Step size: 256 samples (50% overlap between consecutive windows)
- Segmenting long signals into smaller fixed-size windows increases the number of trainings samples and captures important local variations within the EEG signals.

Label assigning

- Label 0: Assigned to EEG windows from Normal healthy individuals.
- Label 1: Assigned to EEG windows from Epilepsy patients during non-seizure periods.

- Label 2: Assigned to EEG windows from Epilepsy patients during seizure episodes.

By assigning labels at the window level, the classification model can be trained to accurately differentiate between normal EEG, epileptic EEG without seizure, and epileptic EEG during seizure events.

3.1.3 Feature Extraction

In this step, a set of statistical, entropy-based, and frequency-domain features is extracted from the EEG windows for further analysis and classification. The following features are computed for each EEG window:

Time-Domain Features

- Mean: The average value of the EEG signal over the window. This provides insight into the overall level of the signal.
- Standard Deviation: Measures the spread or variability of the signal values around the mean.
- Kurtosis: Quantifies the peak sharpness of the signal distribution. Higher kurtosis indicates a signal with more extreme values than a normal distribution.
- Skewness: Measures the asymmetry of the signal's distribution. Positive skew indicates a long tail on the right, and negative skew indicates a long tail on the left.

Entropy-Based Features

- Approximate Entropy: Measures the regularity or unpredictability of the signal. Lower values indicate more regular and predictable signals, while higher values indicate greater randomness.
- Sample Entropy: A more robust version of Approximate Entropy, which quantifies the complexity of the signal while being less sensitive to data length.

Frequency-Domain Features

- Power Spectral Density (PSD): Represents how the signal's power is distributed across different frequencies. This feature gives insights into the energy content of the signal across the frequency spectrum.

- Alpha Power (8–12 Hz): The power in the alpha band, associated with calm and relaxed brain states.
- Beta Power (12–30 Hz): The power in the beta band, associated with active thinking and mental engagement.
- Theta Power (4–8 Hz): The power in the theta band, typically seen during drowsiness or light sleep.
- High-Frequency Energy (HFE): The energy above 30 Hz, which may be relevant for detecting epileptic seizures.

Chaotic Features

- Largest Lyapunov Exponent (LLE): A measure of the signal's chaotic nature, indicating the sensitivity to initial conditions. Higher values reflect more chaotic behaviour in the signal.
- Hjorth Complexity: Measures the complexity of the signal's time course, providing information about the shape and smoothness of the signal.

Statistical Time-Domain Feature

- Root Mean Square (RMS): Provides a measure of the signal's overall magnitude. Higher RMS values indicate stronger or more intense signal activity.

Fractal Feature

- Hurst Exponent: A measure of the long-term memory of the signal. It quantifies whether the signal exhibits persistent behaviour (positive values) or anti-persistent behaviour (negative values).

3.1.4 Data splitting

A commonly used technique for data splitting is the 80-20 split, where 80% of the data is used for training the model, and 20% is reserved for testing the model's performance. This allows the model to learn from a large portion of the data while leaving a separate set for evaluation.

- Training Set (80%): This subset is used to train the machine learning model.
- Testing Set (20%): This subset is used to assess the model's performance, simulating how the model will perform on new, unseen data.

- In this case, the dataset was split using Stratified Sampling to ensure that the distribution of labels (i.e., seizure vs. non-seizure) is maintained across both the training and testing sets

3.1.5 Model Training

The following machine learning models were trained on the dataset:

- Logistic Regression: A linear model used for classification tasks, ideal for situations where the relationship between the input features and the target variable is approximately linear.
- Support Vector Machine (SVM): A powerful classifier that works well for both linear and non-linear classification problems. It is particularly useful in high-dimensional spaces like EEG data.
- K-Nearest Neighbours (KNN): A non-parametric model that classifies data based on the majority label of its nearest neighbours. KNN is known for its simplicity and effectiveness in handling non-linear relationships.
- Random Forest: An ensemble learning method that constructs multiple decision trees and combines their results to improve classification accuracy. Random Forest is highly effective in reducing overfitting and improving generalization.
- Gradient Boosting: A boosting algorithm that builds models sequentially, where each new model attempts to correct the errors of the previous one. It is known for its ability to deliver high performance in classification tasks.

3.1.6 Model Evaluation

The model is evaluated for different scoring metrics like

- Precision, Recall, F1_Score
- Accuracy Score
- Confusion Matrix

F1_Score

The F1 Score is a performance metric commonly used for classification problems, especially in cases of imbalanced datasets. It combines two other metrics, precision and recall, into a single value that balances both, making it particularly useful when both false positives and false negatives carry significant consequences.

Precision measures the accuracy of positive predictions:

$$\text{Precision} = TP/(TP+FP) \qquad (3.1)$$

Recall (also known as sensitivity or true positive rate) measures the completeness of positive predictions:

$$\text{Recall} = TP/(TP+FN) \qquad (3.2)$$

Calculating the F1 Score

The F1 Score is the harmonic mean of precision and recall, defined as

$$\text{F1 Score} = 2(\text{Precision}.\text{Recall})/(\text{Precision}+\text{Recall}) \qquad (3.3)$$

Accuracy Score:

The accuracy score is a performance metric used to evaluate the performance of a classification model. It measures the proportion of correct predictions made by the model compared to the total number of predictions.

Formula:

Accuracy = (Number of correct predictions)/ (Total number of predictions)

It can also be expressed as:

$$\text{Accuracy} = (TP+TN)/(TP+TN+FP+FN) \qquad (3.4)$$

- TP (True Positive): Correctly predicted positive instances.
- TN (True Negative): Correctly predicted negative instances.
- FP (False Positive): Incorrectly predicted negative instances as positive.
- FN (False Negative): Incorrectly predicted positive instances as negative.

Confusion Matrix:

A table showing the counts of true positives, true negatives, false positives, and false negatives for each class.

The confusion matrix is useful for visualizing the model's classification results and identifying specific misclassifications. This can help pinpoint issues like class imbalances or systematic errors in predictions for certain classes

3.2 DEEP LEARNING APPROACH

In the early phases of detecting epileptic seizures, conventional machine learning algorithms such as Random Forest, Support Vector Machines (SVM), and Decision Trees were utilized. These models depended significantly on manual feature extraction, necessitating the careful selection and input of specific statistical features (e.g., mean, standard deviation, entropy) into the model.

However, this methodology presented certain drawbacks: manual feature extraction is labor intensive and may overlook critical hidden patterns. Machine learning models excel with straightforward features but encounter difficulties with complex, high-dimensional data such as EEG signals and they are unable to autonomously learn temporal or spatial dependencies from raw data.

To address these issues, deep learning was introduced, which provides several benefits automatic feature extraction through deep neural networks that learn valuable features from raw EEG data without manual input, enhanced capability to manage complex data, as deep models like CNNs LSTMs and ResNet18 effectively capture intricate patterns in time-series EEG data, improved accuracy, as deep learning architectures generally outperform traditional machine learning models in tasks involving complex data. The ability to learn temporal dependencies through models like LSTM and GRU, which is essential for seizure detection and scalability, allowing deep learning models to be expanded with larger datasets for better performance. Consequently, we shifted from machine learning to deep learning to enhance the efficiency, accuracy, and automation of epileptic seizure detection using EEG signals.

We proposed and implemented several deep learning models for EEG-based seizure detection, including 1D CNN, LSTM, GRU, 1D CNN + LSTM, 1D CNN + GRU, Custom 2D CNN, and ResNet18 (using Transfer Learning) as detailed below.

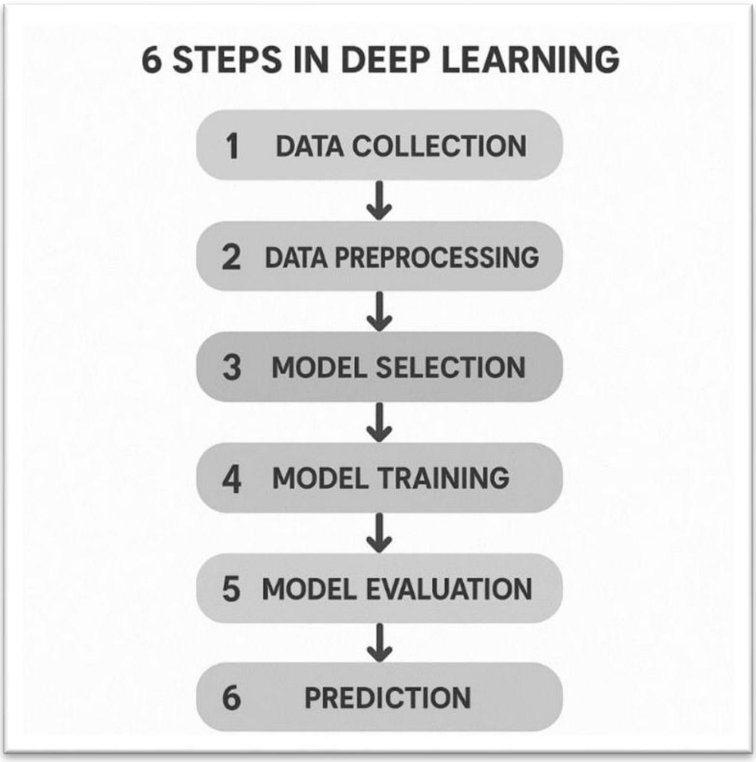

Fig 3.2.1 Block diagram for Deep Learning

STEPS IN DEEP LEARNING:

Step 1: Data Collection

- Obtain EEG signal data from several subjects.
- Classify the data into three classes:
    - Normal brain activity
    - Epilepsy without seizure
    - Epilepsy with seizure

Step 2: Data Preprocessing

- Transform raw EEG data from.txt format to structured.xlsx format.
- Normalize EEG signals to have all values in a common range (e.g., [0,1]).
- Cut continuous EEG signals into smaller fixed-size windows (e.g., 2 seconds per window).
- Assign each windowed segment with its respective class.
- Use One-Hot Encoding for the class labels to ready the data for multi-class classification.

Step 3: Data Splitting

- Split the preprocessed dataset into:
    - Training Set (e.g., 80% of data)
    - Validation Set (e.g., 20% of data)
- Make sure to have random and balanced splitting among all three classes.

Step 4: Model Development

Create and implement several deep learning models:

- **1D CNN:** To learn spatial features from EEG sequences.
- **LSTM:** To learn long-term temporal dependencies in EEG signals.
- **GRU:** To model sequential patterns efficiently with fewer parameters.
- **1D CNN + LSTM:** A hybrid model combining CNN for feature extraction and LSTM for temporal learning.
- **1D CNN + GRU**: Similar hybrid model using GRU instead of LSTM.
- **Custom 2D CNN:** Reshape EEG into 2D input and use convolutional layers.
- **ResNet18 (Transfer Learning):** Utilize a pre-trained ResNet18 model fine-tuned for EEG 2D classification.

Step 5: Model Compilation:

Select suitable components for all models:

- Loss Function: Cross-Entropy Loss for multi-class classification.
- Optimizer: Adam optimizer with learning rate tuning.
- Metrics: Monitor Accuracy during training.

Step 6: Model Training

Train every model individually using the training dataset:

- Set batch size and number of epochs.
- Track loss and accuracy at every epoch.
- Implement early stopping if validation loss begins to rise, to avoid overfitting.
- For Transfer Learning (ResNet18), freeze lower layers first and fine-tune higher layers.

Step 7: Model Evaluation

Once trained, test each model on the validation dataset. Create evaluation reports with:

- Accuracy
- Precision
- Recall
- F1-score
- Confusion Matrix
- ROC-AUC Curves for multi-class analysis

Step 8: Model Performance Comparison:

Compare all models based on evaluation metrics:

- 1D CNN performance
- LSTM standalone performance
- GRU standalone performance
- Improvement using 1D CNN + LSTM and 1D CNN + GRU

3.2.1 1DCNN (Convolutional Neural Network):

A 1D Convolutional Layer (Conv1D) in deep learning is specifically used to process one-dimensional (1D) sequence data. This layer is especially suitable for temporal sequence tasks like audio analysis, time-series prediction, or natural language processing (NLP), where the data itself is linear and sequential. The main operation in a Conv1D layer is to slide a convolutional filter (or kernel) over the input sequence. This filter consists of a set of weights to learn that the network adapts upon training. Convolution computes a product of filter values and input values at every position in the segment of sequence and adds these results to provide a single point of output. This operation extends over the complete sequence to form an output transformed sequence.

Feature	Description
Input	1D sequence data (e.g., time series, text).
Operation	Sliding convolutional filters over the sequence.
Purpose	Extracting high-level features from sequence data.
Applications	Audio signal processing, time-series analysis, NLP.

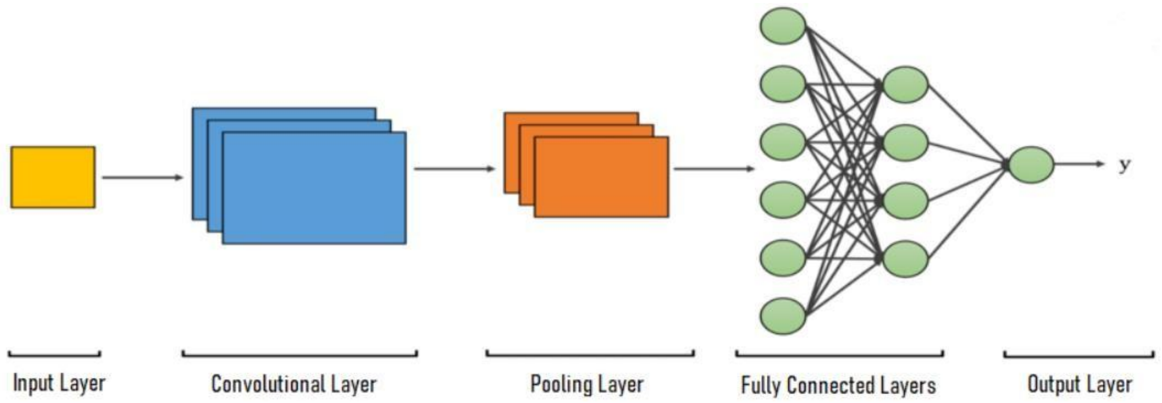

Fig3.2.1.1 1D CNN Architecture

Our 1D CNN model architecture consists of two convolutional layers, two pooling layers, a flatten layer, and dense layers for classification.

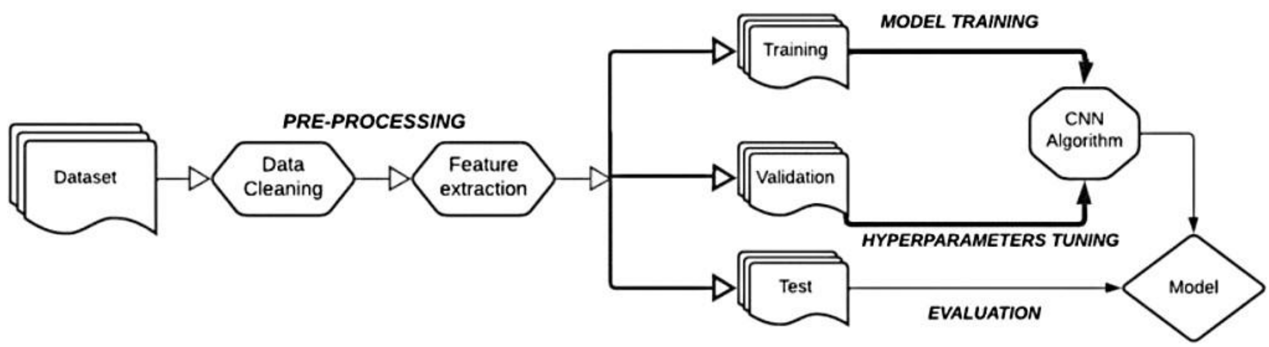

Fig 3.2.1.2 Block diagram for 1D CNN

Model Limitations:

- Limited Temporal Understanding:

  1D CNNs are excellent in learning local patterns in EEG signals but fail to learn long-term temporal dependencies like LSTM or GRU models. EEG signals have significant time-based changes that may be lost if a pure CNN is used.
- Fixed Feature Extraction:

  The fixed-size convolutional filters used in 1D CNN would perhaps fail to detect features happening at varying scales or speeds over varying patients or seizures.

- Sensitivity to Data Segmentation:

  Performance of 1D CNN largely relies upon how the EEG signals are being segmented. Wrong segmentation may cause critical patterns being lost.

  Deep CNN models typically need large quantities of labeled data to generalize. With small EEG data, overfitting (learning too much from training data but failing to generalize to new data) becomes a threat.

- Limited Sequential Memory:

  In contrast to LSTM/GRU, 1D CNNs lack memory — they process each window of data independently, without learning history or sequence of previous windows.

3.2.2 1DCNN (Convolutional Neural Network) +LSTM (Long Short Term Memory:

The 1D CNN + LSTM model is a hybrid deep learning framework that incorporates strengths of Convolutional Neural Networks (CNNs) and Long Short-Term Memory (LSTM) networks. In our project, we used a combination of around 5–7 layers. First, 1D Convolutional layers to extract local features, followed by pooling. Then an LSTM layer to capture temporal patterns, and finally dense layers for classification. We also used dropout to prevent overfitting.

How it works:

1. 1D CNN:

The 1D CNN part of the model takes the sequential input and uses convolutional filters to learn local patterns and features. These filters slide across the input data, extracting relevant information.

2. LSTM:

The output of the 1D CNN is then fed into an LSTM layer, which processes the sequential features to learn long-term dependencies and relationships within the data.

3. Output Layer:

Finally, a fully connected layer or other output layer can be used to make predictions based on the extracted features and learned dependencies.

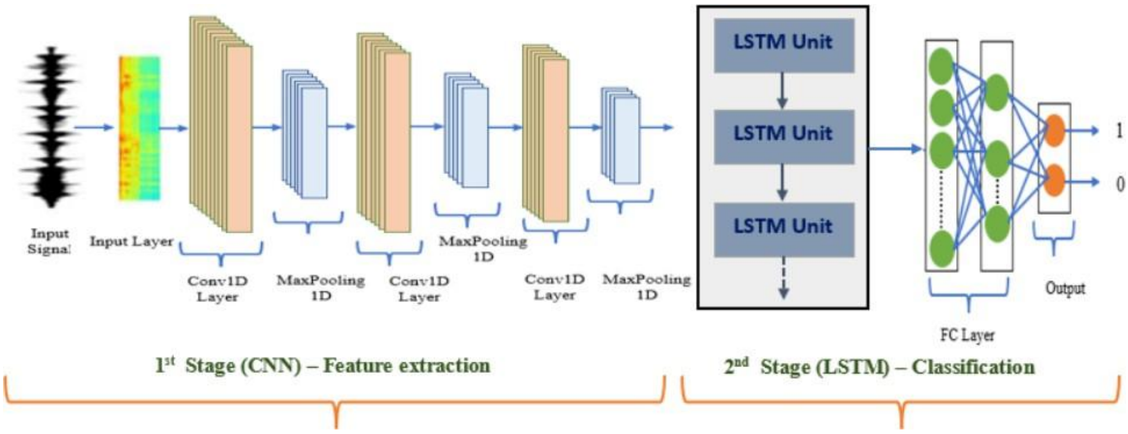

Fig 3.2.2.1 Architecture of 1DCNN+LSTM

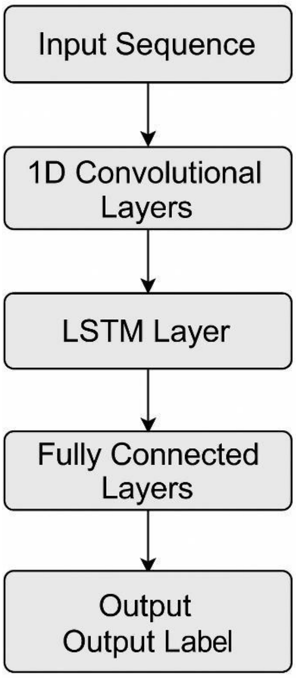

Fig 3.2.2.2 Block diagram of 1DCNN+LSTM

Limitations of 1DCNN+LSTM:

Training Time:

Hybrid models are slower to train because two complex architectures are being combined.

Overfitting Risk:

Unless well-tuned, the model will overfit the training data, particularly with smaller EEG datasets.

High Computational Resources Needed:

Requires more GPU memory and compute power than simple 1D CNN or LSTM alone.

3.2.3 1DCNN (Convolutional Neural Network) + GRU (Gated Recurrent Unit):

The 1D CNN + GRU model is a powerful architecture that effectively learns from time-series data by combining the local feature extraction capability of CNN and the sequential learning capability of GRU. This approach is particularly useful in applications like EEG seizure detection, where both local patterns and long-term temporal dependencies in the data need to be captured to make accurate predictions.

Work of 1D CNN + GRU model:

Input Data: This is just raw sequential data, like EEG signals or time-series stuff.

1D CNN Layers: These layers grab the local features from the input data.

GRU Layers: Here, we focus on capturing the time relationships and what comes next in the data.

Dense Layers: After the GRU layers, we have fully connected layers that handle the learned insights.

Output Layer: Finally, we use softmax activation to classify things into set categories.

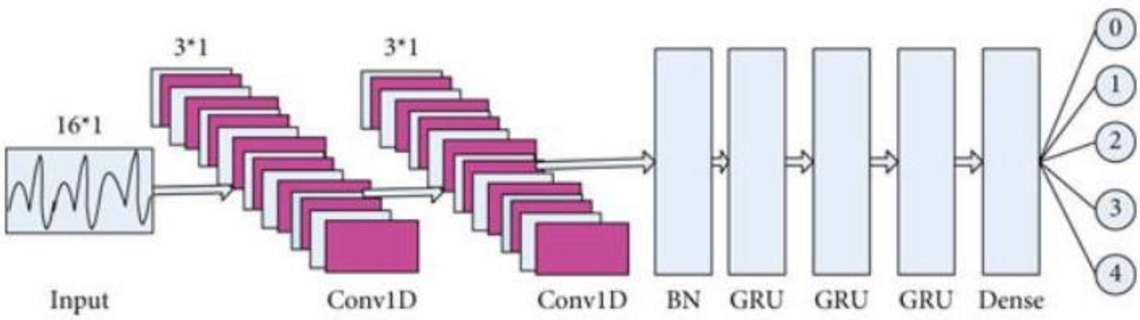

Fig 3.2.3.1 Architecture of 1DCNN+GRU

Limitations of the 1D CNN + GRU combination:

1. High Computational Cost

The combination of CNN and GRU can be computationally expensive due to the need for processing both local patterns (via CNN) and long-term dependencies (via GRU). This is particularly true for large datasets or long sequences.

2. Difficulty Handling Very Long Sequences

While 1D CNN captures local features well, it struggles with long-term dependencies by itself. GRU helps with long-term dependencies, but may still struggle when the sequence is very long, due to the limitations of recurrent models in handling distant past information.

3. Overfitting Risk

The combination of multiple CNN filters and GRU units can lead to overfitting, especially when the dataset is small. This increases the risk of the model memorizing the training data rather than generalizing well.

4. Vanishing Gradient Problem (For GRU)

When working with very long sequences, GRU still encounters the vanishing gradient problem, which can impede learning, particularly in deeper networks, even though it is less severe than with conventional RNNs.

5. Tuning Complexity

Hyperparameter tuning becomes complex because both the CNN and GRU components require careful adjustments. The number of filters, kernel size for CNN, and GRU units need to be optimized, which can be time-consuming.

3.2.4 LSTM (LONG SHORT-TERM MEMORY):

LSTM stands for **Long Short-Term Memory**. It is a special type of Recurrent Neural Network (RNN) designed to learn and remember long-term dependencies in sequence data.

In traditional RNNs, as the sequence becomes longer, they struggle to remember information from earlier in the sequence due to a problem called vanishing gradients. LSTM networks solve this by introducing a unique structure called a memory cell and gates, which control the flow of information.

An LSTM unit has three main gates:

- **Forget Gate**: Decides which information should be thrown away from the cell.
- **Input Gate**: Decides which new information should be added to the cell.
- **Output Gate**: Decides what information should be output.

Because of these gates, LSTM can store important information for a long time, making it very powerful for tasks like EEG signal analysis.

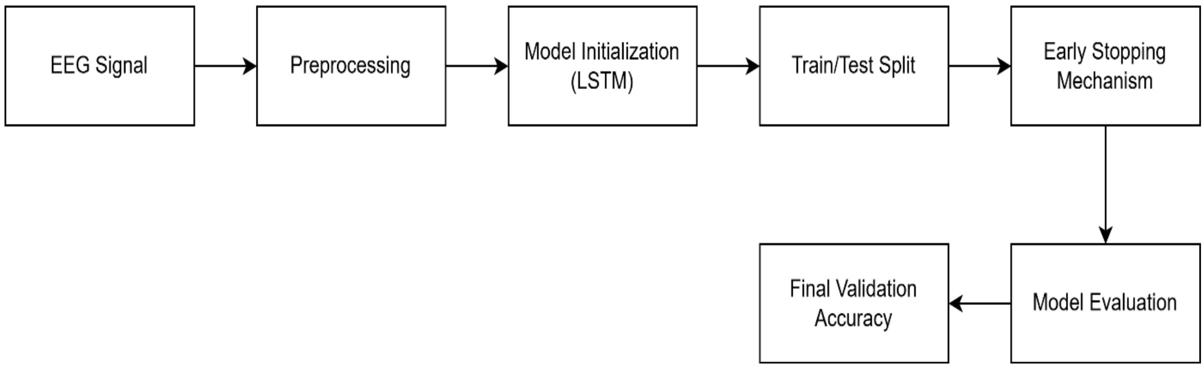

Fig 3.2.4.1 Block diagram of LSTM

Why LSTM is Useful for EEG Seizure Detection

In EEG seizure detection, the brain's electrical signals change over time, and it is important to capture these time-based patterns. LSTM networks are ideal for this task because they can remember important information from earlier time steps and connect it with later data. Unlike traditional models, LSTM can effectively handle the long and complex sequences of EEG signals, making it easier to detect seizures accurately. By learning the hidden temporal relationships in EEG data, LSTM improves both the sensitivity and reliability of seizure prediction.

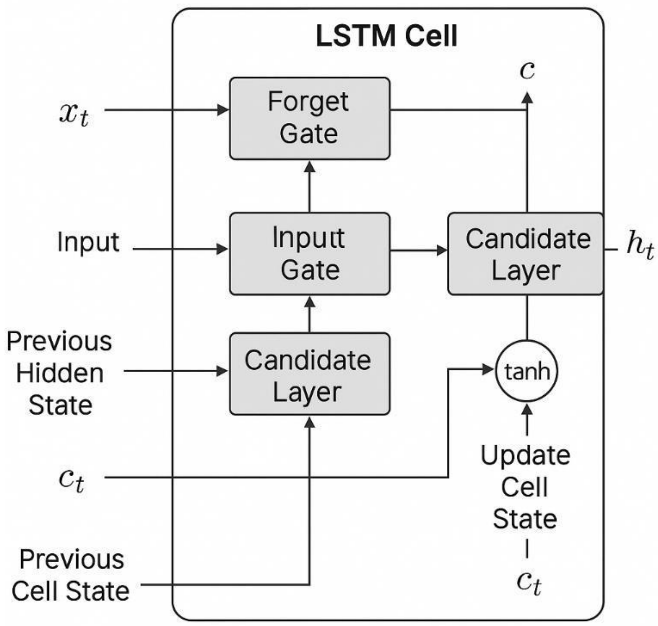

Fig 3.2.4.2 LSTM Architecture

Limitations:

1. Computational Complexity

- LSTMs have many parameters (gates, cell states, hidden states).
- This makes them heavy in terms of memory usage and computation time.
- Training the LSTM model could be slow, especially with large EEG datasets.

2. Difficult to Parallelize

- Each LSTM output depends on previous outputs (sequential).
- So, unlike CNNs where operations can be done in parallel, LSTM must process one time step at a time.
- This makes training slower on modern GPUs.

3. Overfitting on Small Datasets

- LSTMs are very powerful — but if the EEG dataset is small, the model may overfit (memorize training data without generalizing).
- You already applied Early Stopping — which is good to control this**.**

4. Vanishing and Exploding Gradients (still possible)

- LSTMs were designed to solve the vanishing gradient problem better than vanilla RNNs.
- For very long sequences, LSTM can still struggle with learning dependencies.

3.2.5 GRU (Gated Recurrent Unit)

GRU is a simplified version of LSTM (Long Short-Term Memory) designed for sequential data like EEG signals. It combines the forget and input gates of LSTM into a single update gate, making the architecture simpler and faster. GRU uses fewer parameters than LSTM, leading to faster training and less risk of overfitting on smaller datasets. Despite being simpler, GRUs often achieve similar or even better performance than LSTMs on many tasks. In EEG projects like seizure detection, GRU can efficiently capture temporal patterns while being lighter and faster than LSTM.

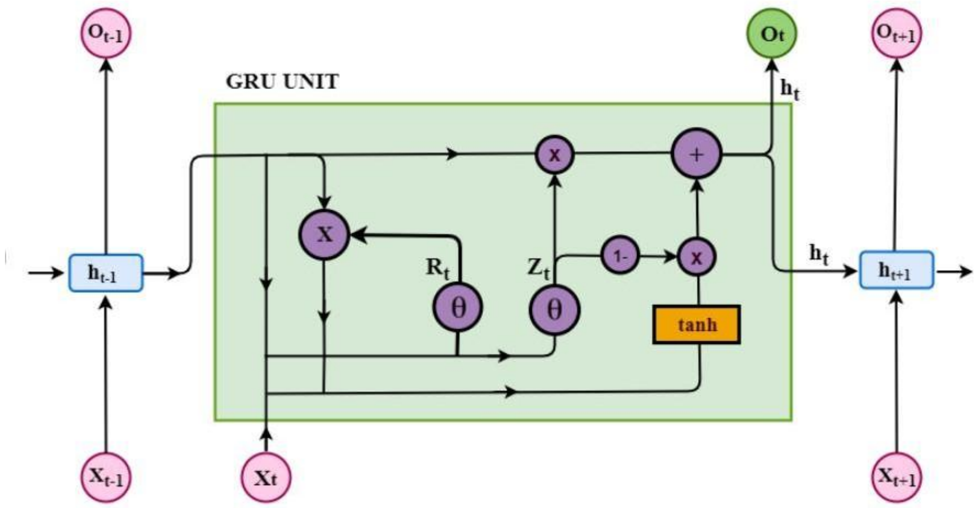

Fig 3.2.5.1 GRU Architecture

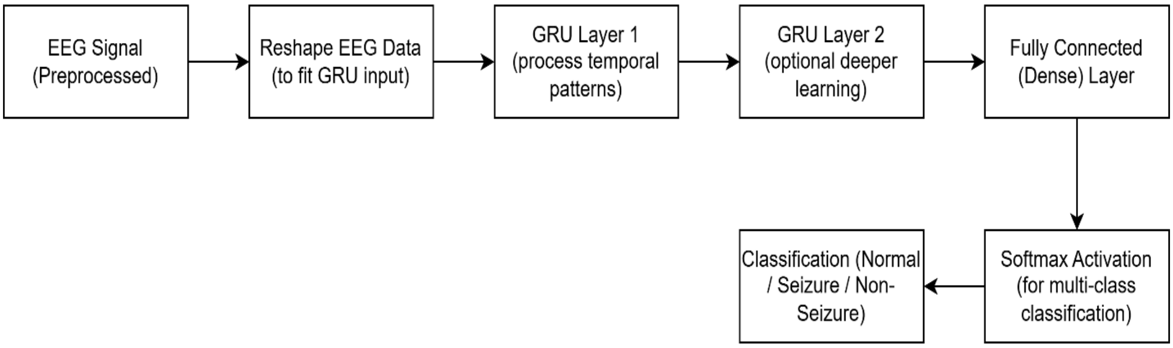

Fig 3.2.5.2 Block Diagram of GRU

Why is GRU useful in EEG Seizure Detection?

1.Captures Temporal Dependencies

EEG signals are time-series data — patterns over time matter (like sudden spikes during a seizure). GRU can remember important features over time and forget irrelevant noise, helping to identify seizure patterns effectively.

2.Faster and Lighter

EEG datasets can be large and complex. GRUs have fewer gates (only update and reset gates) compared to LSTMs. This makes GRU faster to train and requires less memory, very useful when you have limited computational resources.

3. Prevents Overfitting

GRU's simpler architecture compared to LSTM reduces overfitting, making it generalize better to new, unseen EEG data.

Limitations:

Short-Term Memory Bias

Although GRUs capture temporal patterns, they might forget very long-term dependencies compared to more complex models like LSTMs.

Difficulty Handling Complex Temporal Structures

EEG signals are highly complex; GRUs might oversimplify very intricate time-based features, affecting classification quality.

Overfitting Risk

GRUs can overfit on small EEG datasets if not properly regularized (dropout, early stopping needed).

Less Interpretability

Like most deep learning models, GRUs act like black boxes, making it hard to explain why a specific EEG segment is classified as seizure/non-seizure.

Performance Depends Heavily on Hyperparameters

The model's success is sensitive to hidden size, number of layers, learning rate, etc., requiring careful tuning.

Computational Cost (Moderate)

Although lighter than LSTM, GRUs still require considerable training time if the EEG data is large or very high-resolution.

3.2.6 CUSTOM 2D CNN:

A Custom 2D CNN is a specially designed convolutional neural network tailored for EEG signal classification.In this model, preprocessed EEG signals are reshaped into a 2D format and passed through layers of convolution, batch normalization, activation (ReLU), and pooling. Residual blocks are added to improve deeper learning without vanishing gradients. The model automatically learns spatial patterns from the EEG signals that differentiate between Normal, Seizure, and Non-Seizure classes. Finally, fully connected layers and a softmax output perform the final classification. This architecture improves feature extraction and gives higher accuracy compared to traditional CNNs.

Why Custom 2D CNN is Useful in EEG Seizure Detection:

1. Captures Spatial Relationships:

By reshaping EEG signals into a 2D format, the CNN can capture spatial dependencies between different parts of the signal, improving seizure pattern recognition.

2. Residual Connections Improve Training:

Using residual blocks (skip connections) allows the model to be deeper without suffering from vanishing gradients, leading to better learning and higher accuracy.

3. Efficient and Robust:

A custom design allows optimization for EEG-specific challenges, making the network lighter, faster, and more robust to noise and signal variations.

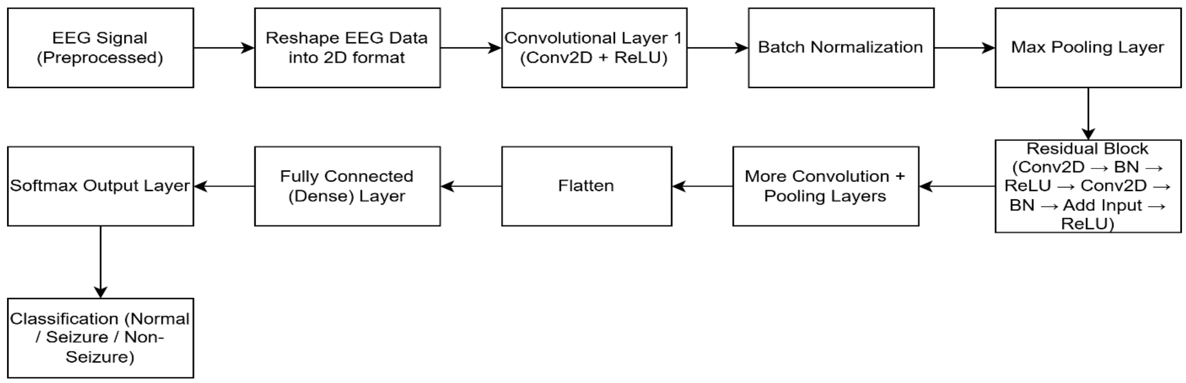

Fig 3.2.6.1 Block diagram for custom 2D CNN

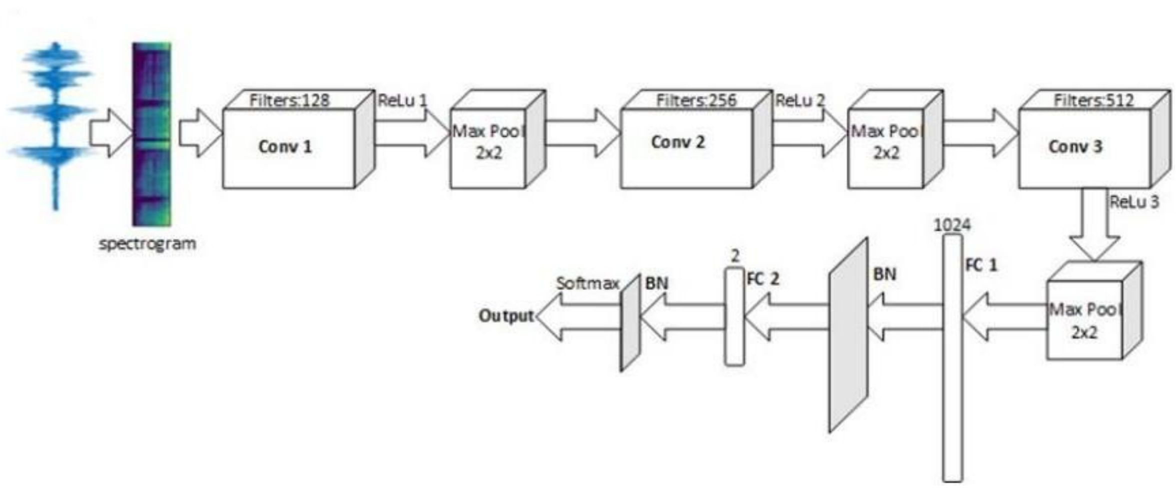

Fig 3.2.6.2 2D CNN Architecture

Limitations:

1. High Computational Resources:

Even a custom CNN may need powerful GPUs and longer training times, especially when deeper architectures are used.

2. Loss of Temporal Information:

When EEG signals are reshaped into 2D formats, temporal (time-related) dynamics might not be fully captured, which are important for detecting seizures.

3. Sensitive to Preprocessing:

The CNN’s performance heavily depends on how well the EEG data is preprocessed (e.g., noise removal, scaling).

Poor preprocessing can lead to degraded model performance.

4. Overfitting Risk:

Custom 2D CNNs, especially deep ones, can easily overfit if the dataset is small or imbalanced, even when techniques like dropout and regularization are applied.

3.2.7 Resnet-18

ResNet-18 is a popular deep learning model introduced as part of the Residual Networks (ResNet) family. It has 18 layers deep (including convolutional, batch normalization, activation, and fully connected layers). Its main feature is the use of skip connections (also called residual connections) that allow the network to bypass some layers, helping to solve the vanishing gradient problem. ResNet-18 is lightweight compared to deeper versions (like ResNet-50 or ResNet-101) and is often used when computational resources are limited but deep learning performance is still needed. It is highly effective for image classification, feature extraction, and even signal classification tasks like EEG-based seizure detection after proper adaptation.

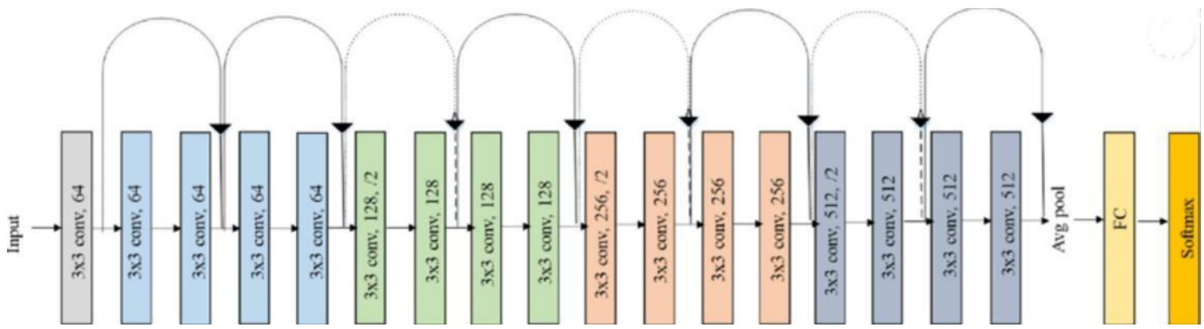

Fig 3.2.7.1 ResNet-18 Architecture

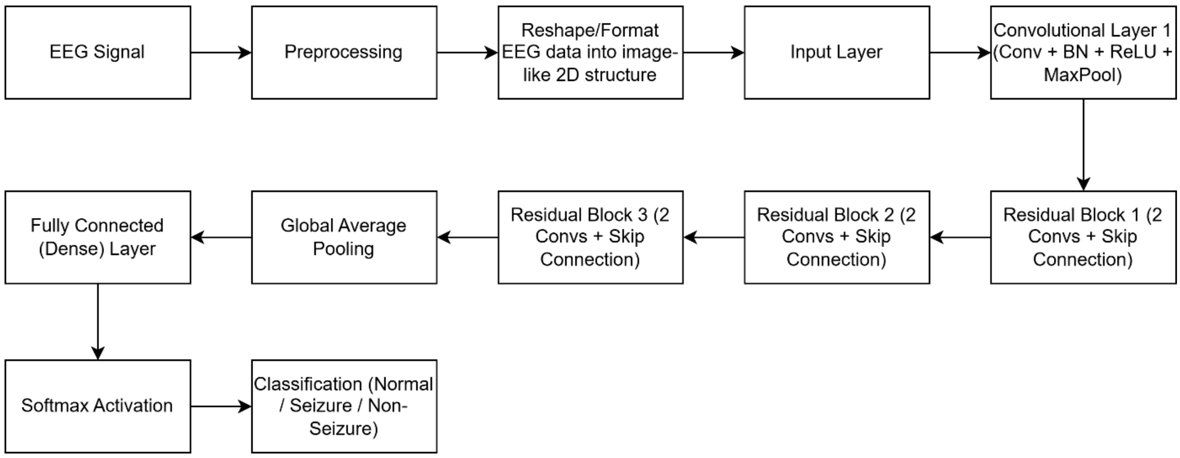

Fig 3.2.7.2 Block diagram of ResNet18 Architecture

Why ResNet-18 is Useful in EEG Seizure Detection:

1. Captures Complex Patterns:

EEG signals are complex and noisy. ResNet-18, with its deep structure and residual connections, can capture both low-level and high-level features in the signal data without degrading performance.

2. Avoids Vanishing Gradient Problem:

In deep networks, gradients can become very small during training (vanish). ResNet-18's skip connections allow gradients to flow directly through the network, making training more stable and faster — important for EEG data where small changes are critical.

3. Feature Reusability:

Residual blocks encourage the network to reuse important features, helping to focus on seizure-related patterns while ignoring irrelevant noise.

CHAPTER 4

RESULTS AND DISCUSSION

4.1 Results for Machine Learning Classification report

1.Random Forest

Classification Report:

	precision	recall	f1-score	support
0	0.98	0.97	0.97	600
1	0.97	0.97	0.97	600
2	0.98	0.99	0.99	300
accuracy			0.97	1500
macro avg	0.98	0.98	0.98	1500
weighted avg	0.97	0.97	0.97	1500

Fig 4.1.1 1DCNN Classification Report

2.Support Vector Machine

✅ Support Vector Machine Accuracy: 0.9320

	precision	recall	f1-score	support
0	0.93	0.93	0.93	583
1	0.92	0.92	0.92	607
2	0.96	0.95	0.95	310
accuracy			0.93	1500
macro avg	0.94	0.94	0.94	1500
weighted avg	0.93	0.93	0.93	1500

Fig 4.1.2 Support Vector Machine Classification Report

3. Gradient Boosting

✅ Gradient Boosting Accuracy: 0.9633

	precision	recall	f1-score	support
0	0.96	0.97	0.96	583
1	0.96	0.95	0.95	607
2	0.98	0.98	0.98	310
accuracy			0.96	1500
macro avg	0.97	0.97	0.97	1500
weighted avg	0.96	0.96	0.96	1500

Fig 4.1.3 Gradient Boosting Classification Report

4. K-neighbors

✅ K-Nearest Neighbors Accuracy: 0.9333

	precision	recall	f1-score	support
0	0.93	0.93	0.93	583
1	0.92	0.92	0.92	607
2	0.98	0.96	0.97	310
accuracy			0.93	1500
macro avg	0.94	0.94	0.94	1500
weighted avg	0.93	0.93	0.93	1500

Fig 4.1.4 K-neighbors Classification Report

5. Logistic Regression

✅ Logistic Regression Accuracy: 0.9247

	precision	recall	f1-score	support
0	0.93	0.94	0.94	583
1	0.91	0.91	0.91	607
2	0.95	0.92	0.94	310
accuracy			0.92	1500
macro avg	0.93	0.92	0.93	1500
weighted avg	0.92	0.92	0.92	1500

Fig 4.1.5 Logistic Regression Classification Report

Discussion

Comparison of the ML Models

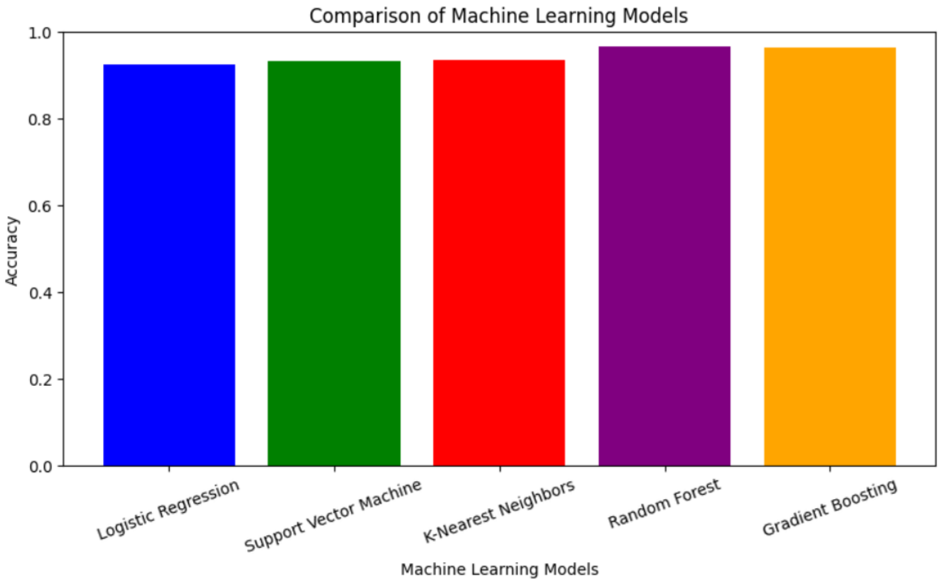

Fig 4.1.6 Comparison of models

Out of the five machine learning models evaluated, Random Forest exhibited the best accuracy in epileptic seizure detection. Its ensemble nature enabled it to treat intricate EEG features effectively and prevent overfitting, performing better than Logistic Regression, SVM, KNN, and Gradient Boosting. Although other models provided decent results, Random Forest emerged as the strongest and most consistent option for precise EEG signal classification.The **Accuracy** of Random forest model is 97%

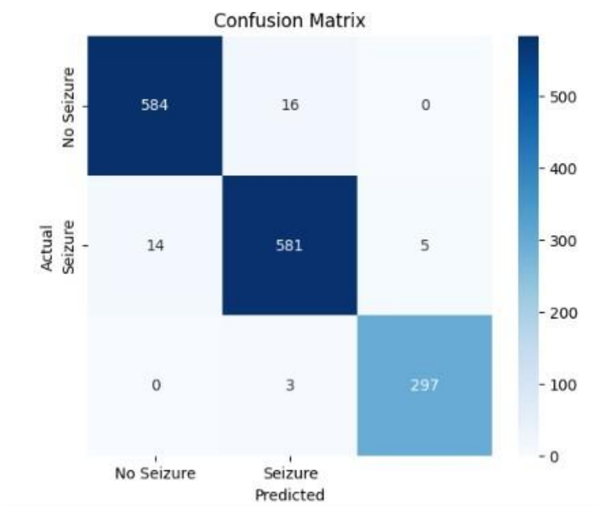

Fig 4.1.7 Confusion Matrix of Random Forest

Although Random Forest had the best accuracy among the machine learning models, the confusion matrix indicated some misclassifications between seizure and non-seizure classes. These misclassifications indicate poor generalization and minor overfitting during testing. Because of these issues, deep learning models were investigated to improve the capture of complex EEG patterns and enhance classification performance.

4.2 Results for Deep Learning

4.2.1 1D CNN(Convolutional Neural Network):

📊 Classification Report (Validation Set):

	precision	recall	f1-score	support
Seizure	1.00	1.00	1.00	60
Non-Seizure	1.00	0.97	0.99	120
Normal	0.98	1.00	0.99	120
accuracy			0.99	300
macro avg	0.99	0.99	0.99	300
weighted avg	0.99	0.99	0.99	300

Fig 4.2.1.1 1DCNN Classification Report

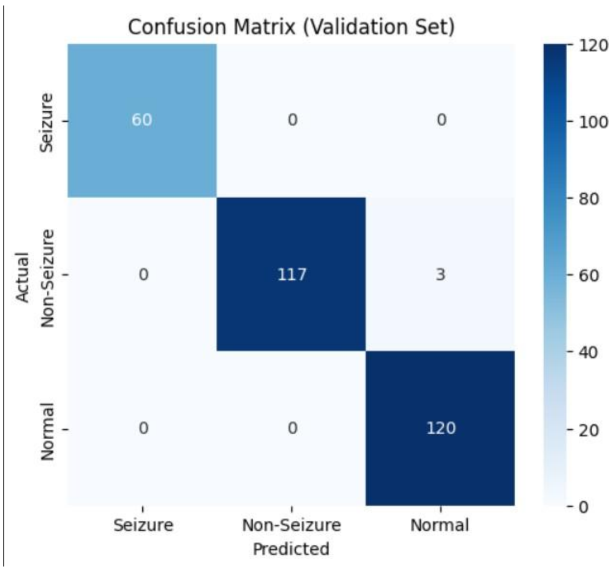

Fig 4.2.1.2 Confusion Matrix of 1DCNN

4.2.2 1D CNN+LSTM:

Classification Report:

	precision	recall	f1-score	support
Seizure	0.90	0.90	0.90	10
No Seizure	0.96	0.90	0.92	48
Normal	0.91	0.98	0.94	42
accuracy			0.93	100
macro avg	0.92	0.92	0.92	100
weighted avg	0.93	0.93	0.93	100

Fig 4.2.2.1 Classification Report of 1DCNN+LSTM

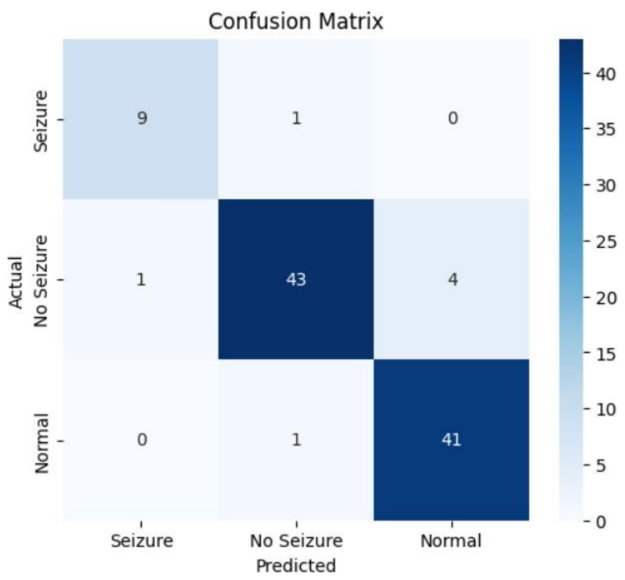

Fig 4.2.2.2 Confusion Matrix of 1DCNN+LSTM

4.2.3 1D CNN+GRU:

Classification Report:

	precision	recall	f1-score	support
0	0.97	0.97	0.97	40
1	0.91	0.97	0.94	40
2	1.00	0.85	0.92	20
accuracy			0.95	100
macro avg	0.96	0.93	0.94	100
weighted avg	0.95	0.95	0.95	100

Fig 4.2.3.1 Classification report of 1DCNN+GRU

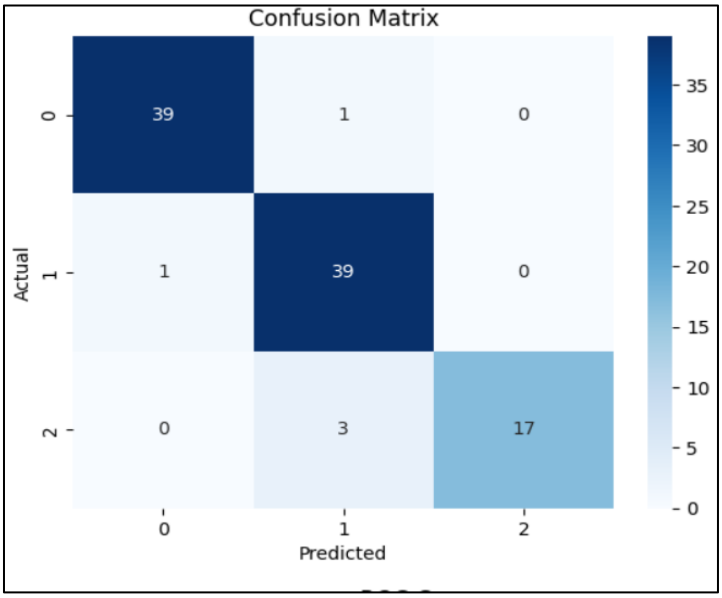

Fig 4.2.3.2 Confusion Matrix of 1DCNN+GRU

4.2.4 LSTM(Long Short Term Memory):

```
Confusion Matrix:
[[32  0  8]
 [ 4  9  7]
 [ 8  0 32]]

Classification Report:
                          precision    recall  f1-score   support

                  Normal       0.73      0.80      0.76        40
                 Seizure       1.00      0.45      0.62        20
Epilepsy Without Seizure       0.68      0.80      0.74        40

                accuracy                           0.73       100
               macro avg       0.80      0.68      0.71       100
            weighted avg       0.76      0.73      0.72       100

Validation Accuracy: 73.00%
```

Fig 4.2.4.1 Confusion matrix and Classification report of LSTM

4.2.5 GRU (Gated Recurrent Unit):

```
Confusion Matrix:
[[31  0  9]
 [ 3  9  8]
 [ 8  0 32]]
Classification Report:
                         precision    recall  f1-score   support

                 Normal       0.74      0.78      0.76        40
                Seizure       1.00      0.45      0.62        20
Epilepsy Without Seizure      0.65      0.80      0.72        40

               accuracy                           0.72       100
              macro avg       0.80      0.68      0.70       100
           weighted avg       0.76      0.72      0.71       100

Validation Accuracy: 72.00%
```

Fig 4.2.5.1 Confusion matrix and Classification report of GRU

4.2.6 Custom 2DCNN:

```
📊 Classification Report:
              precision    recall  f1-score   support

     Seizure       1.00      1.00      1.00        60
 Non-Seizure       0.99      0.97      0.98       120
      Normal       0.98      0.99      0.98       120

    accuracy                           0.99       300
   macro avg       0.99      0.99      0.99       300
weighted avg       0.99      0.99      0.99       300
```

Fig 4.2.6.1 Classification Report of Custom 2DCNN

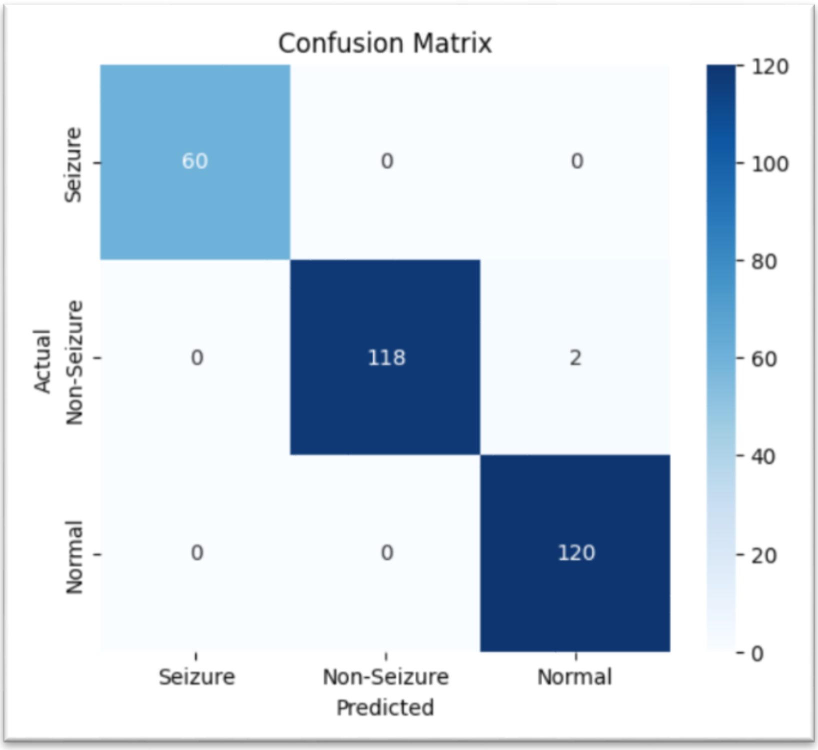

Fig 4.2.6.2 Confusion Matrix of Custom 2DCNN

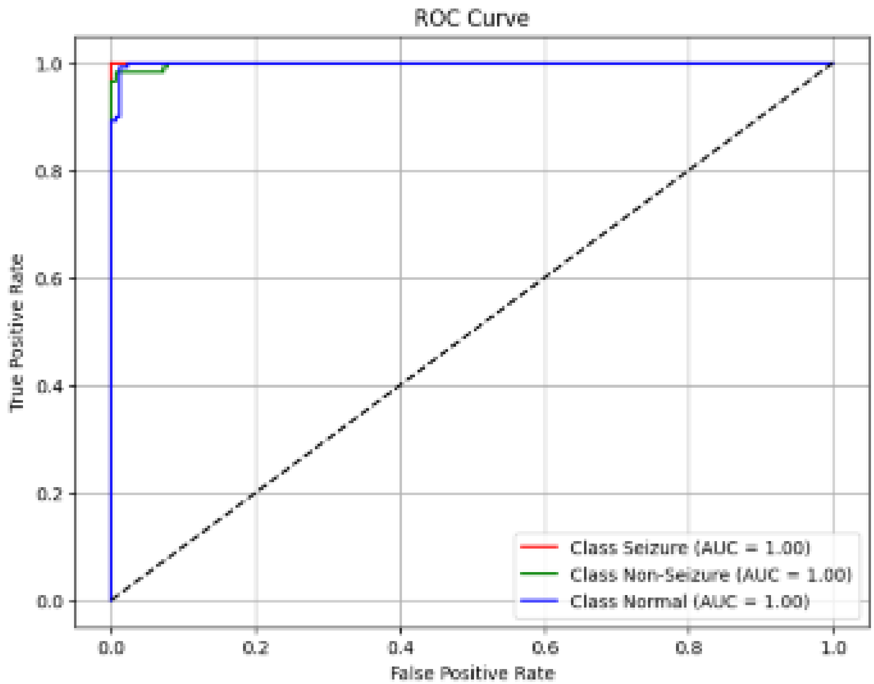

Fig 4.2.6.3 ROC Curve for Custom 2DCNN

4.2.7 ResNet18:

Classification Report:

	precision	recall	f1-score	support
0	1.00	1.00	1.00	40
1	0.97	0.97	0.97	40
2	0.95	0.95	0.95	20
accuracy			0.98	100
macro avg	0.97	0.97	0.97	100
weighted avg	0.98	0.98	0.98	100

Fig 4.2.7.1 Classification Report of ResNet18

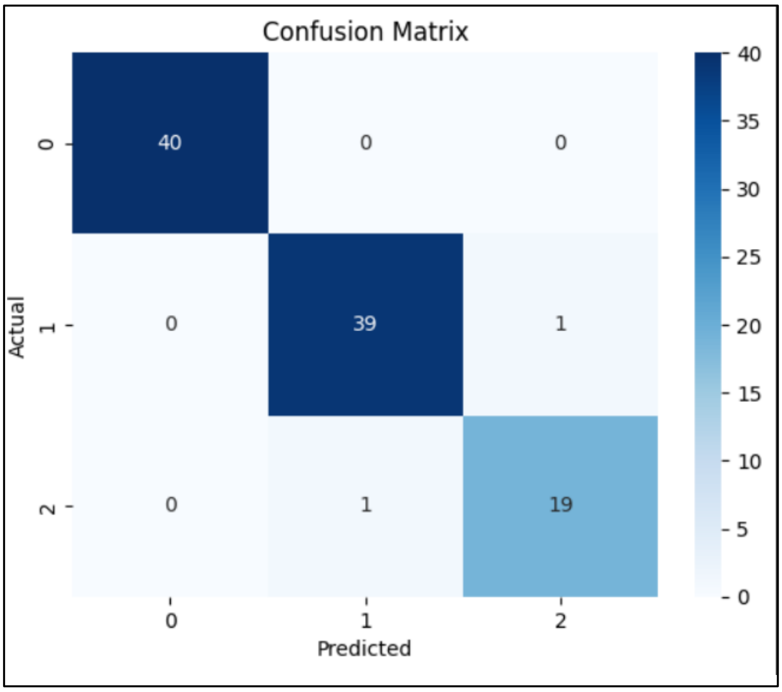

Fig 4.2.7.2 Confusion Matrix of ResNet18

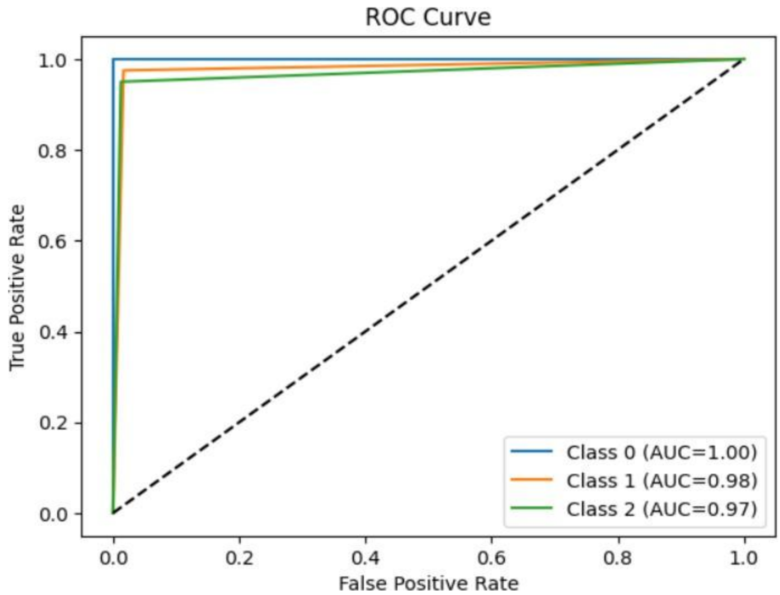

Fig 4.2.7.3 ROC Curve of ResNet18

4.2.8 DISCUSSION:

This project aimed to develop an accurate and automated epileptic seizure detection system using EEG data and deep learning methods. Among the various models explored—1D CNN, 1D+LSTM,1D+GRU,LSTM, GRU, ResNet18, and a custom 2D CNN—the custom 2D CNN and ResNet18 significantly outperformed others in terms of classification accuracy. The custom 2D CNN achieved 99% accuracy, while ResNet18 reached 98%, making them highly effective for EEG signal classification.

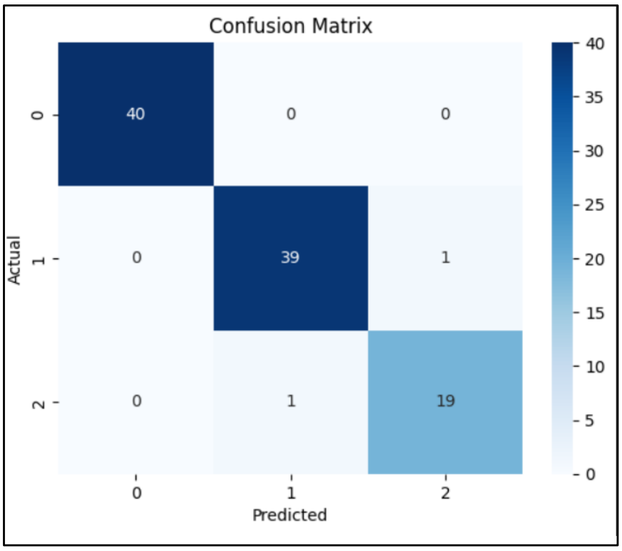

Fig 4.2.7.2 Confusion Matrix of ResNet18

Total correct predictions: 40 (Seizure) + 39 (Non Seizure) + 19 (Normal) = **98**

Total misclassifications: 2

Overall accuracy: 98%

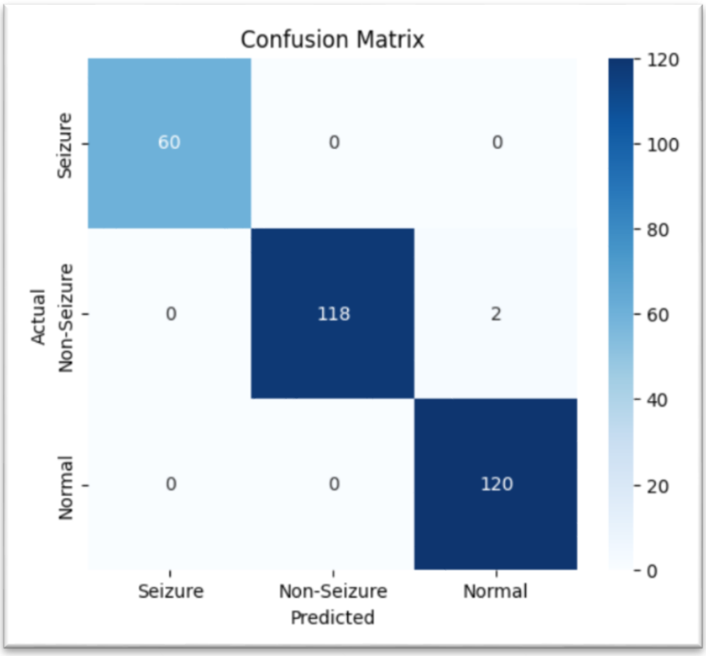

Fig 4.2.6.2 Confusion Matrix of Custom 2DCNN

Total samples evaluated: 60 (Seizure) + 120 (Non Seizure) + 120 (Normal) = **300**

Total correct predictions: 298

Total misclassifications: 2

Overall accuracy: 99.33%

CHAPTER 5

CONCLUSION

This project effectively proved the capability of both machine learning and deep learning methods for epileptic seizure detection from EEG signals. By systematic experimentation, conventional machine learning models like Random Forest showed strong performance with 97% accuracy, which validated that well-engineered feature-based models are still very much relevant. But deep learning models, especially Custom 2D CNN and ResNet18, pushed the performance even further, recording accuracies of 99% and 98% respectively. These findings demonstrate the impressive capacity of deep architectures to learn and extract intricate features automatically from EEG data.

Additionally, although individual LSTM and GRU models performed fairly well, their usage in association with 1D CNN improved their performance, stressing the significance of hybrid models. Generally, this research confirms that deep learning, particularly CNN-based algorithms, is promising for constructing highly reliable and accurate seizure detection machines.

Future research might concentrate on real-time operation, optimization of the model for edge machines, and validation on more extensive, more varied EEG datasets to achieve greater clinical relevance

CHAPTER 6

REFERENCES

[1] **S. A. Siddiqui, H. Menendez, G. Huang, and F. K. Hussain,** "*A reviewon epileptic seizure detection using machine learning classifiers*" Expert Systems with Applications, vol. 149, Art. no. 113319, 2020.

[2] **A. Guerrero, R. Parada, and C. Espitia**, *"Epileptic seizure detection using EEG sound analysis and machine learning techniques*" in Proc. IEEE Colombian Conf. Applications in Computational Intelligence (ColCACI), Bogotá, Colombia, pp. 1–6, 2021.

[3] **A. Gramacki and J. Gramacki**, "*Detection of neonatal seizures using deep convolutional neural networks*" Biomed. Signal Process. Control, vol. 71, Art. no. 103154, 2022.

[4] **H. Kode, K. Elleithy, and L. Almazaydeh**, "*A hybrid machine learning and deep learning approach for epileptic seizure detection*" Biomed. Signal Process. Control, vol. 86, Art. no. 104227, 2024.