EMPLOYING ADVANCED MACHINE LEARNING TECHNIQUES FOR DETECTING EPILEPTIC SEIZURES THROUGH PHYSIOLOGICAL SIGNALS

A PROJECT WORK I REPORT

Submitted by

AARTHI B
21ADR001

HARISH K
21ADR017

SWETHA NALLAMANGAI K N
21ADR054

in partial fulfilment of the requirements

for the award of the degree

of

BACHELOR OF TECHNOLOGY

IN

ARTIFICIAL INTELLIGENCE AND DATA SCIENCE

DEPARTMENT OF ARTIFICIAL INTELLIGENCE

KONGU ENGINEERING COLLEGE
(Autonomous)

PERUNDURAI, ERODE-638 060

MAY 2024

DEPARTMENT OF ARTIFICIAL INTELLIGENCE

KONGU ENGINEERING COLLEGE
(Autonomous)

PERUNDURAI, ERODE – 638 060

MAY 2024

BONAFIDE CERTIFICATE

This is to certify that the Project Report titled **EMPLOYING ADVANCED MACHINE LEARNING TECHNIQUES FOR DETECTING EPILEPTIC SEIZURES USING PHYSIOLOGICAL SIGNALS** is the bonafide record of project work done by **AARTHI B (21ADR001), HARISH K (21ADR017)**, **SWETHA NALLAMANGAI K N (21ADR054)** in partial fulfilment of the requirements for the award of Degree of Bachelor of Technology in Artificial Intelligence and Data Science of Anna University, Chennai during the year 2023-2024.

SUPERVISOR **HEAD OF THE DEPARTMENT**
(Signature with seal)

Date:

Submitted for the end semester viva voce examination held on_____.

INTERNAL EXAMINER **EXTERNAL EXAMINER**

DEPARTMENT OF ARTIFICIAL INTELLIGENCE

KONGU ENGINEERING COLLEGE
(Autonomous)

PERUNDURAI, ERODE – 638 060

MAY 2024

DECLARATION

We affirm that the Project Report titled **EMPLOYING ADVANCED MACHINE LEARNING TECHNIQUES FOR DETECTING EPILEPTIC SEIZURES USING PHYSIOLOGICAL SIGNALS** being submitted in partial fulfilment of the requirements for the award of Bachelor of Technology is the original work carried out by us. It has not formed part of anyother project report or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

Date:

AARTHI B

(21ADR001)

HARISH K

(21ADR017)

SWETHA NALLAMANGAI K N

(21ADR054)

I certify that the declaration made by the above candidate is true to the best of my knowledge.

Name & Signature of the Supervisor with seal

Date:

ABSTRACT

Epilepsy is a neurological disorder characterized by unpredictable seizures, posing significant challenges for timely intervention and management. Traditional methods of seizure detection rely heavily on clinical observation and electroencephalogram (EEG) analysis, often requiring expert interpretation and limited by accessibility. In recent years, the integration of advanced machine learning techniques with physiological signal analysis has emerged as a promising approach for automated seizure detection. This study explores the application of advanced machine learning algorithms, such as deep learning and ensemble methods, for detecting epileptic seizures using physiological signals. Physiological signals EEG, offer valuable insights into the physiological changes associated with seizures. The proposed framework involves preprocessing of physiological signals to extract relevant features, followed by the development of machine learning models trained on labeled datasets. Feature selection techniques are employed to enhance model efficiency and interpretability. Additionally, ensemble methods are utilized to combine the strengths of multiple models, improving overall performance and robustness. The results demonstrate the feasibility and effectiveness of employing advanced machine learning techniques for detecting epileptic seizures through physiological signals. The proposed framework holds promise for the development of non-invasive, real-time seizure detection systems that can aid healthcare professionals in timely intervention and personalized treatment strategies for individuals with epilepsy.

ACKNOWLEDGEMENT

First and foremost, we acknowledge the abundant grace and presence of the almighty throughout different phases of the project and its successful completion.

We wish to express our sincere gratitude to our honorable Correspondent **Thiru.A.K.ILANGO B.Com.,M.B.A.,LLB.,** and other trust members for having provided us with all necessary infrastructures to undertake this project.

We extend our hearty gratitude to our honorable Principal **Dr.V.BALUSAMY B.E.(Hons)., MTech., Ph.D.,** for his consistent encouragement throughout our college days.

We would like to express our profound interest and sincere gratitude to our respected Head of the department **Dr.C.S.KANIMOZHISELVI ME., Ph.D.,** for her valuable guidance.

A special debt is owed to the project coordinator **Ms. S. SANTHIYA B.E., M.E.**. Assistant Professor, Department of Artificial Intelligence for their encouragement and valuable advice that made us to carry out project work successfully.

We extend our sincere gratitude to our beloved guide **Ms.N.ABINAYA M.Tech** Assistant Professor, Department of Artificial Intelligence for her ideas and suggestions, which have been very helpful to complete the project.

We are grateful to all the faculty and staff members of the Department of Artificial Intelligence and persons who directly and indirectly supported this project.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF ABBREVIATIONS

ML	:	Machine Learning
DL	:	Deep Learning
LSTM	:	Long Short Term Memory
BiLSTM	:	Bi directional Long Short Term Memory
GRU	:	Gated Recurrent Unit
BiGRU	:	Bi directional Gated Recurrent Unit
RNN	:	Recurrent Neural Network

CHAPTER 1

INTRODUCTION

1.1 INTRODUCTION

Millions of individuals worldwide suffer with epilepsy, a neurological condition marked by recurring seizures. These seizures, caused by abnormal electrical activity in the brain, can vary in severity and impact, leading to significant challenges in diagnosis, treatment, and management. Early detection of epileptic seizures plays a crucial role in mitigating their adverse effects and improving patient outcomes.

In recent years, advancements in machine learning (ML) techniques have shown promising results in automating the detection of epileptic seizures. Traditional methods often rely on manual observation or basic signal processing techniques, which can be subjective, time-consuming, and prone to error. However, with the advent of advanced ML algorithms and the availability of large-scale datasets, researchers have been able to develop more robust and accurate seizure detection systems.

This report aims to provide an overview of the state-of-the-art ML techniques employed in the detection of epileptic seizures. It will delve into the challenges associated with seizure detection, the importance of early diagnosis, and the role of ML in addressing these challenges. Additionally, the report will discuss various ML algorithms, including but not limited to deep learning, recurrent neural networks (RNNs), and ensemble methods, highlighting their strengths and limitations in seizure detection tasks.

Furthermore, this project will explore the significance of feature engineering, data preprocessing, and model optimization in enhancing the performance of seizure detection algorithms. It will also examine the integration of wearable devices, electroencephalogram (EEG) signals, and other physiological data sources into ML-based seizure detection systems, enabling real-time monitoring and personalized healthcare solutions.

Overall, by leveraging advanced ML techniques, researchers and healthcare professionals have the potential to revolutionize the diagnosis and management of epilepsy, ultimately improving the quality of life for individuals living with this condition. This project aims to contribute to the growing body of knowledge in this field and inspire further research and innovation in epileptic seizure detection.

1.2 OBJECTIVE

- Highlight the challenges associated with epileptic seizure detection, emphasizing the need for accurate and efficient automated systems.
- Discuss the importance of early seizure detection in improving patient outcomes and quality of life.
- Review the state-of-the-art ML algorithms, including deep learning models such as recurrent neural networks (RNNs), along with ensemble methods, for seizure detection tasks.
- Explore the role of feature engineering, data preprocessing, and model optimization in enhancing the performance of seizure detection algorithms.
- Investigate the integration of wearable devices, electroencephalogram (EEG) signals, and other physiological data sources into ML-based seizure detection systems for real-time monitoring and personalized healthcare solutions.
- Provide insights into current research trends, challenges, and future directions in the field of epileptic seizure detection using ML techniques.
- Ultimately, contribute to the advancement of knowledge and inspire further research and innovation in the development of effective and reliable seizure detection systems for individuals living with epilepsy.

1.3 SCOPE

- Perform a feature importance analysis to identify the Epileptic seizures with the generated EEG signals.

CHAPTER 2

SYSTEM ANALYSIS

2.1 LITERATURE REVIEW

In 2023, Sakorn Mekruksavanich et al. [1] provided the application of deep learning models, specifically Long Short-Term Memory (LSTM), Bidirectional LSTM (BiLSTM), Gated Recurrent Unit (GRU), and Bidirectional GRU (BiGRU), for the detection of epileptic seizures using EEG signals. The study compares the performance of these models in scenario I using the ESRD dataset through a 5-fold cross-validation approach. Additionally, the research introduces the ResNet-BiGRU-ECA model, which demonstrates superior accuracy in seizure recognition compared to other state-of-the-art deep learning networks. The findings highlight the potential of deep learning techniques in improving the accuracy and efficiency of epileptic seizure detection, paving the way for enhanced diagnostic tools in epilepsy management.

In 2020, Rahib Abiyev et al. [2] provided the identification of epileptic EEG signals using deep learning techniques. It discusses the use of convolutional neural networks (CNN) for enhancing the performance of identifying epileptic EEG signals, the challenges faced by neurologists in visually inspecting epileptic abnormalities in EEG signals, and various scientific approaches used for accurate detection of epileptic seizures from EEG signals. Additionally, the review includes references to studies on deep learning in medical imaging, fuzzy neural networks for breast cancer identification, machine learning techniques for breast tissue classification, diagnosing Parkinson's disease using fuzzy neural systems, head mouse control systems for people with disabilities, sign language translation using deep CNNs, improved spiking neural networks for EEG classification and epilepsy detection, and a new supervised learning algorithm for multiple spiking neural networks.

In 2016, Hadi Ratham Al Ghayab et al. [3] provided the classification of epileptic EEG signals using innovative techniques. Researchers have explored the use of simple random sampling (SRS) to extract features from EEG signals and sequential feature selection (SFS) to reduce data dimensionality. By applying these methods in conjunction with a least square support vector machine (LS_SVM) classifier, the study achieved high classification accuracy, sensitivity, and specificity for distinguishing between healthy individuals and epileptic patients. Comparisons with previous methods demonstrate the effectiveness of the proposed approach in analyzing and classifying EEG signals. The research highlights the potential of SRS and SFS techniques in biomedical signal analysis and opens up new possibilities for improving the diagnosis and treatment of neurological disorders like epilepsy.

In 2016, Hadi Ratham Al Ghayab et al. [4] focuses on the classification of epileptic EEG signals using a novel method that combines simple random sampling (SRS) and sequential feature selection (SFS) techniques. EEG signals play a crucial role in medical applications such as diagnosing and treating diseases like epilepsy. The proposed method aims to extract key features from EEG signals, reduce dimensionality through SFS, and classify the signals using a least square support vector machine (LS_SVM) classifier.

In 2021,Muhammed Bilal Qureshi et al. [5] discussed various aspects related to EEG signal processing and machine learning techniques for epileptic seizure detection. It covers the PhysioNet research resource for complicated physiologic signals, hypergraph neural networks, and fuzzy closest neighbor methods. The paper also shows a comparison between the suggested method and the most recent research on EEG categorization, emphasizing the suggested method's better performance. Furthermore, the suggested method's efficacy in EEG categorization is demonstrated by a comparison with earlier literature research. The responsibilities that each author played in conceptualization, supervision, data curation, analysis, and validation are detailed in their contributions. The paper also cites research on machine learning applications for epileptic seizure detection and nonlinear features in brain electrical activity.

In 2023, Marcin Kolodziej et at. [6] covers a range of topics related to the use of machine learning and deep learning techniques for the detection of epileptic seizures using electroencephalography (EEG) signals. Several key studies are referenced, highlighting the importance of psychosocial and economic challenges faced by parents of children with epilepsy , as well as reviews on machine learning for EEG signal processing , deep learning algorithms and architectures , and deep learning techniques for EEG signal applications .

2.2 SUMMARY

The enhanced automated model to identify epileptic seizures in a clinical context and notify medical personnel when a seizure occurs in an ambulance. The study employs fuzzy and conventional machine learning algorithms to classify EEG epochs into interictal and ictal classes. It also suggests a feature extraction method to identify discriminating features in EEG signals and empirically investigates the ideal EEG epoch size for improved signal interpretation. Using benchmark datasets, the classifiers' performance is assessed; the KNN and FRNN algorithms produce impressive results. Analyses that compare the suggested approach to previously published material show how successful it is at automatically detecting epileptic seizures.

CHAPTER 3

SYSTEM REQUIREMENTS

3.1 HARDWARE REQUIREMENTS

CPU type	:	Intel core i5 processors
Ram size	:	8 GB
Hard disk capacity	:	500 GB

3.2 SOFTWARE REQUIREMENTS

Operating System	:	Windows 10
Language	:	Python (version 3+)
Tool	:	Google Colab

3.3 SOFTWARE DESCRIPTION

3.3.1 Python

Python is a strong programming language that is easy to learn. It makes heavy use of information structures and takes a simplistic yet effective approach to object-oriented programming. Python is an excellent language for planning and rapid application development in a variety of disciplines due to its refined phrase structure and imperative composition, which are similar to its made sense character. This essay covers the essential principles and capabilities of Python 3. Python's standard library contains a huge number of functions that are included when the programme is installed. The Python language may utilise a variety of supplementary libraries to explore with more things available online. Several important libraries and techniques are used in the Python code snippets to conduct a variety of tasks. The Pandas library is used for data manipulation and analysis, namely reading and processing data from CSV files, which allows for data pretreatment. Scikit-Learn (sklearn) is a machine learning package that helps with model selection, data partitioning for training and testing, and calculating accuracy scores.

3.3.2 Google Colab Notebook

Google Colab, or Google Colaboratory, is a cloud-based platform that gives free access to Jupyter notebooks, making it simple for users to execute and exchange code. It provides a GPU and TPU-enabled environment for data processing, machine learning, and deep learning activities. Users may interact in real time and save their work in Google Drive. Colab notebooks, like Sheets and Google Docs, may be shared. Either use the Share button in the top right corner of any Colab notebook or follow these instructions for sharing files from Google Drive. Google Drive searches are now accessible for Colab notebooks. When you click the Colab logo in the upper left corner of the notebook view, you will see all of your Drive notebooks. You may also search for recently opened notebooks by going to File > Open Notebook. Your account-only virtual machine is utilised to execute programmes. The Colab service enforces a maximum lifetime for virtual machines, which are removed after a period of inactivity. Any Colab notebook you've produced may be downloaded from Google Drive or accessed through Colab's File menu by following these steps.

CHAPTER 4

PROPOSED SYSTEM

4.1 SYSTEM ARCHITECTURE

This project involves developing a ResNet based model with a BiGRU layer and an Efficient Channel Attention (ECA) module for epileptic seizure detection using EEG signals. ResNet, short for Residual Network, is a type of deep neural network architecture known for its effectiveness in training very deep networks. It introduces residual connections, which allow the model to learn residual mappings rather than directly learning the desired underlying mapping. This helps in mitigating the vanishing gradient problem and enables training of deeper networks more effectively. BiGRU stands for Bidirectional Gated Recurrent Unit. GRU is a type of recurrent neural network (RNN) architecture, which is effective for modeling sequential data due to its ability to capture temporal dependencies. The bidirectional aspect means that the GRU layer processes input sequences not only in the forward direction but also in the backward direction, allowing the model to capture information from both past and future contexts. Channel attention mechanisms focus on recalibrating feature maps by assigning different importance weights to different channels. The Efficient Channel Attention (ECA) module is a specific attention mechanism designed to efficiently capture interdependencies between channels in feature maps. It helps the model to dynamically adjust the importance of each channel according to the context of the input data, thereby enhancing the model's ability to capture relevant information. By combining these components, the proposed model aims to leverage the representational power of ResNet for feature extraction, the temporal modeling capabilities of BiGRU for capturing sequential dependencies in EEG signals, and the attention mechanism of the ECA module to enhance the model's ability to focus on relevant information for epileptic seizure detection. The synergy of these components is expected to result in a robust and effective model for this important medical application. The input given to the model is signals in the form of CSV file. The given input goes to the convolutional layer block and undergoes max pooling. Once this max pooling is done the output of the convolutional layer is then processed by the Hybrid Residual Block. Here batch normalisation takes place and the final output is obtained.

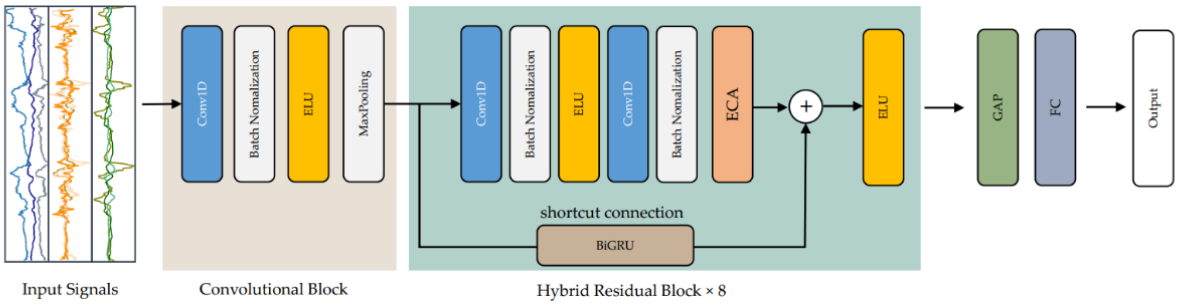

Fig. 4.1 Proposed model workflow

4.2 MODULE DESCRIPTION

4.2.1 Dataset Collection

The data collected here is from the University of Bonn. The data contains of 5 folders each with 100 text files. These 5 folders are classified as the class labels. Each text file has 4096 data points which are recorded EEG signals on the interval of 23.5 seconds.

It has five classes named Z,S,N,O,F.

5 (F) - eyes open indicates that the patient's eyes were open during the recording of the brain's EEG signal.

4 (O) - eyes closed indicates that the patient's eyes were closed during the EEG signal recording.

3 (N) - they locate the tumor's location in the brain while logging the EEG activity from the healthy brain region.

2 (S): The EEG was obtained from the region containing the tumor.

1(Z): Recording seizure activity is done.

4.2.2 Data pre-processing

```
X1      0
X2      0
X3      0
X4      0
X5      0
       ..
X175    0
X176    0
X177    0
X178    0
y       0
Length: 179, dtype: int64
```

Fig.4.2 Missing Values

The missing values have been checked and it was found that no missing values were there. The class distribution has been plotted to make sure that the distribution of data is equal among the five classes and to check if it is an imbalanced data. X values has been standardized and reshaped.

4.2.3 Implementation of Deep Learning Models

4.2.3.1 Long Short-Term Memory (LSTM)

One kind of RNN (recurrent neural network) that can handle and analyze sequential input is an LSTM (Long Short-Term Memory) network. An LSTM network is made up of a sequence of LSTM cells, each of which has a set of input, output, and forget gates to regulate the information entering and leaving the cell. The LSTM can preserve long-term dependencies in the input data by using the gates to selectively forget or keep information from earlier time steps. Additionally, the LSTM cell has a memory cell that uses data from earlier time steps to affect the cell's output at the present time step.

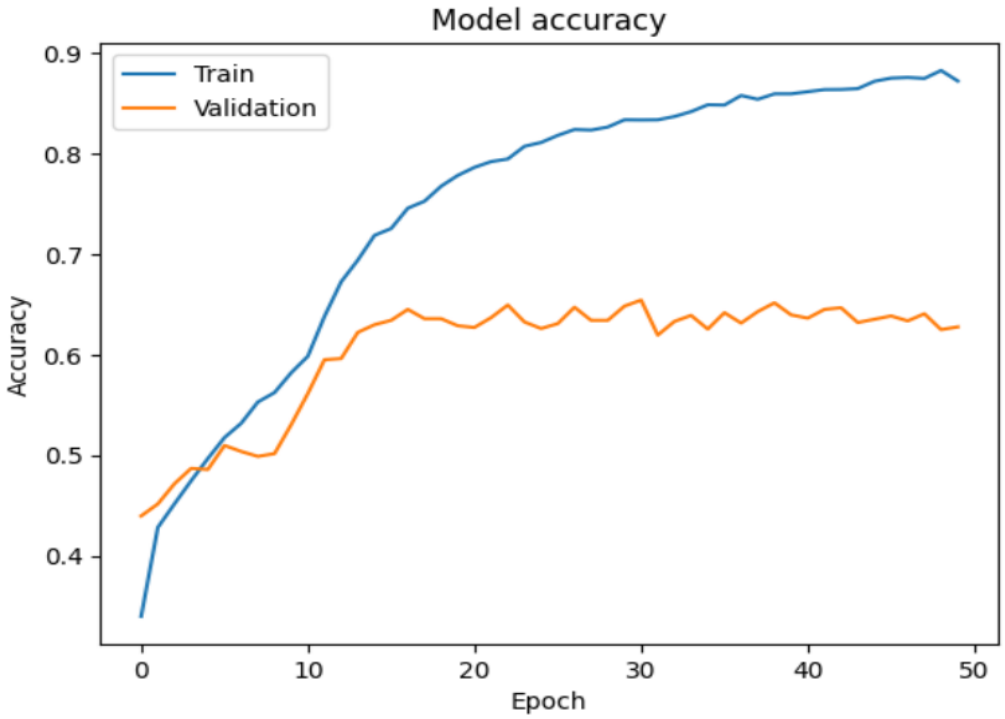

Fig 4.3 Model accuracy of LSTM

4.2.3.2 Bidirectional Long short-term Memory (BiLSTM)

A sequence processing model known as a bidirectional LSTM, or BiLSTM, is made up of two LSTMs, one of which takes input in a forward manner and the other in a backward way. BiLSTMs efficiently expand the network's information pool, enhancing the context that the algorithm has access to (e.g., knowing what words immediately before and follow a word in a sentence).

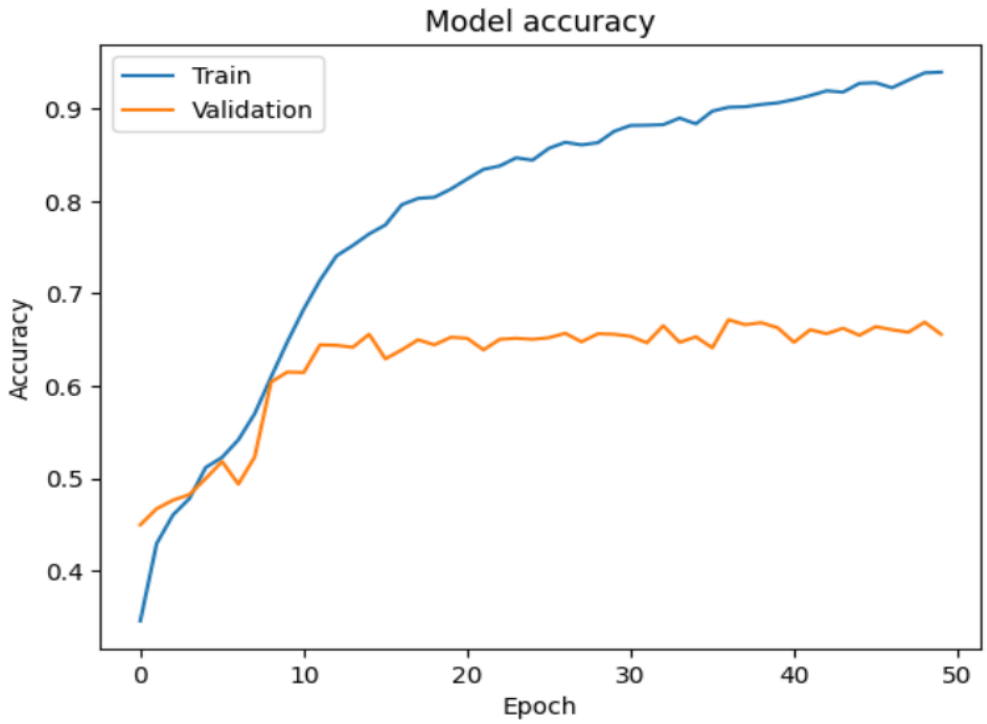

Fig 4.4 Model accuracy of Bi-LSTM

4.2.3.3 Gated recurrent unit(GRU)

Gated Recurrent Units, or GRUs for short, are a kind of recurrent neural network (RNN) architecture that resembles Long Short-Term Memory, or LSTM. Similar to LSTM, GRU uses the concept of selective memory retention and forgetting to describe sequential data. But compared to LSTM, GRU has a simpler, more computationally efficient architecture with fewer parameters, making it easier to train.

The manner that GRU and LSTM handle the memory cell state is the primary distinction between them. Three gates—the input, output, and forget gates—are used in LSTM to update the memory cell state, which is kept apart from the hidden state. A "candidate activation vector," which is updated via the reset and update gates, takes the place of the memory cell state in GRU. While the update gate chooses how much of the candidate activation vector to add to the new hidden state, the reset gate chooses how much of the previous hidden state to ignore. All things considered, GRU is a well-liked substitute for LSTM in the modeling of sequential data, particularly when fewer computational resources or a more straightforward design are required.

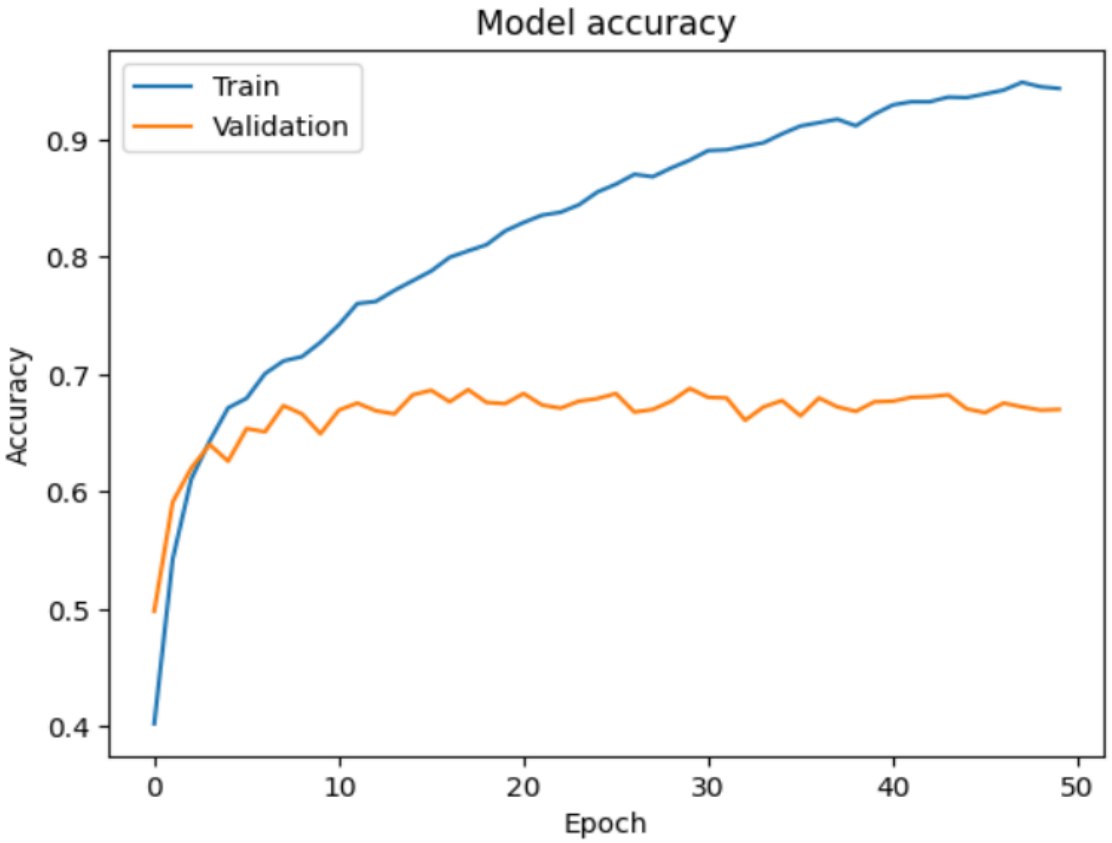

Fig 4.5 Model accuracy of GRU

4.2.3.4 Bidirectional Gated Recurrent Units (BiGRU)

Two GRUs make up a Bidirectional GRU, or BiGRU, which is a model for sequence processing. One is processing the input forward, while the other is processing it backward. This neural network is bidirectional and recurrent, utilizing simply input and forget gates.

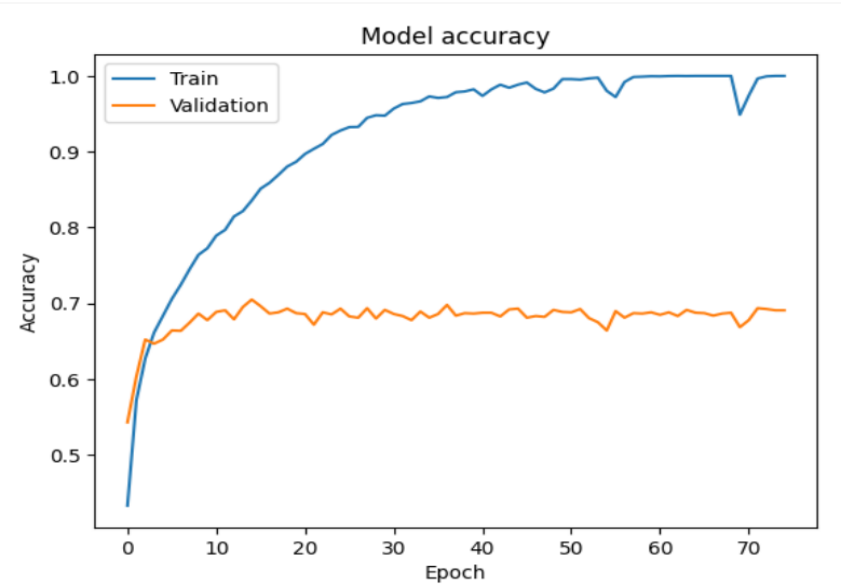

Fig 4.6 Model accuracy of BiGRU

4.2.3.5 Recurrent Neural Network (RNN)

A deep learning model trained to interpret and translate a sequential data input into a specific sequential data output is called a recurrent neural network (RNN). Data with complicated semantics and syntactic rules governing the relationships between sequential components, such as words, phrases, or time series data, is referred to as sequential data. An RNN is a software system made up of numerous interconnected parts that functions similarly to how people convert sequential data, such text from one language to another. Large language models (LLM) and transformer-based artificial intelligence (AI) are replacing RNNs in large part because they are significantly more effective at processing sequential input

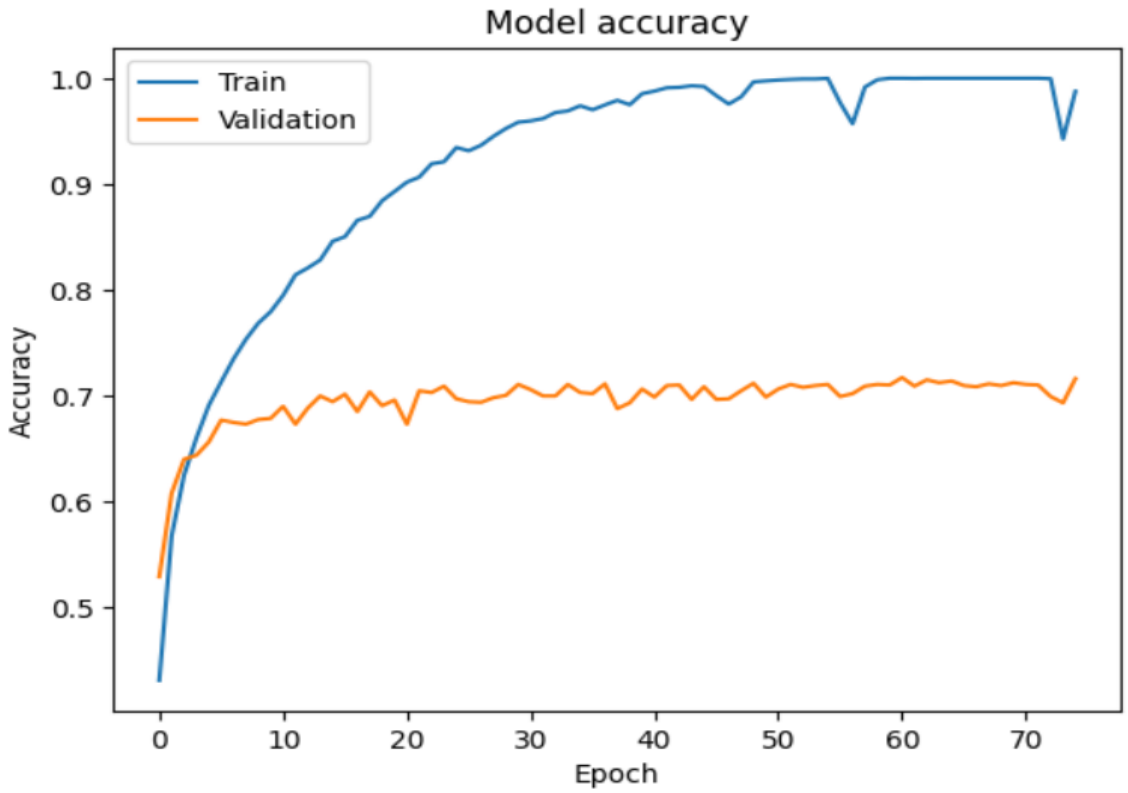

Fig 4.7 Model accuracy of RNN

4.3 VISUALIZATION

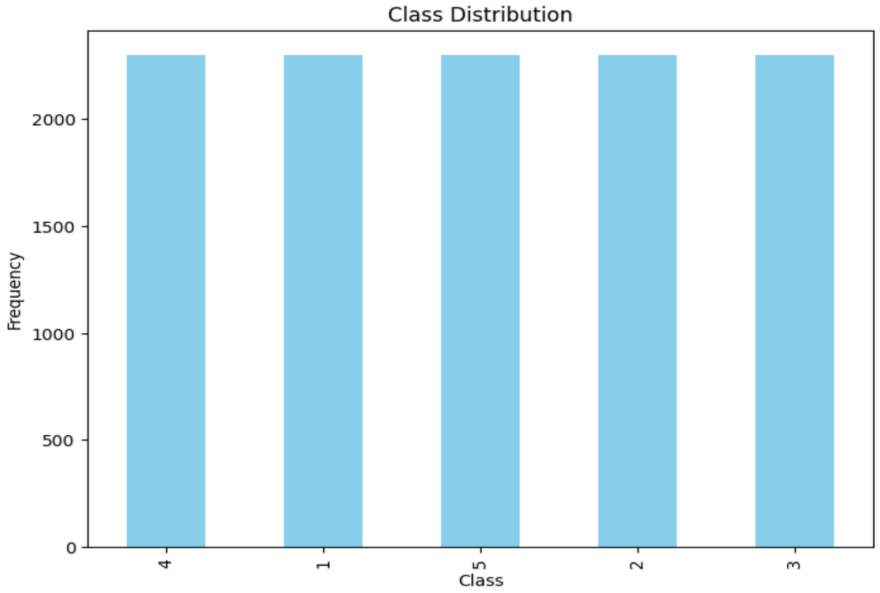

Fig 4.8 Number of instances in each class

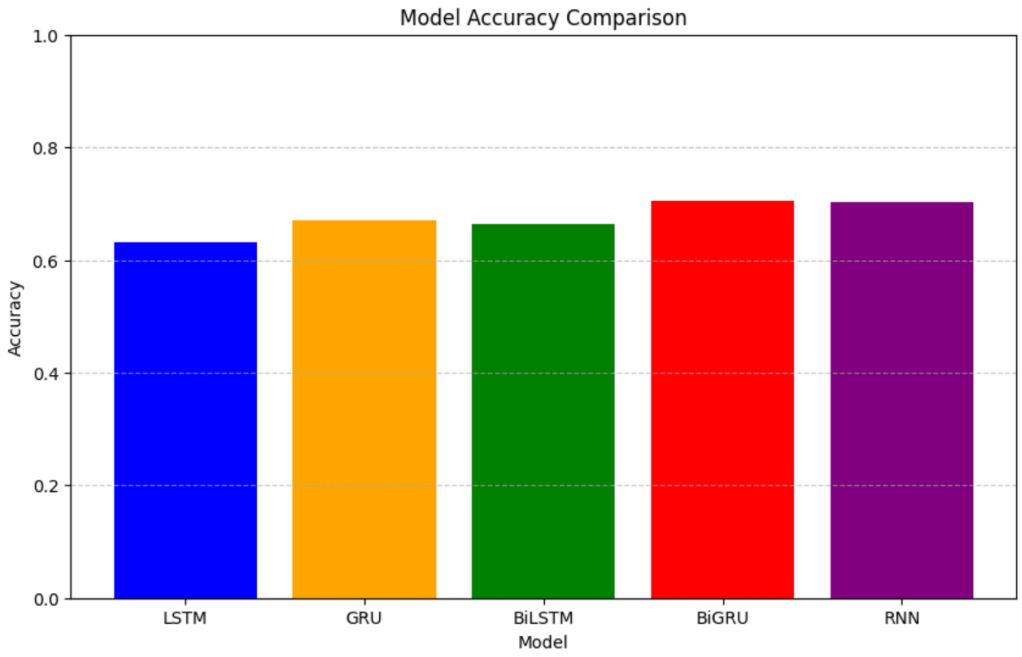

Fig 4.9 Accuracy Comparison for each model

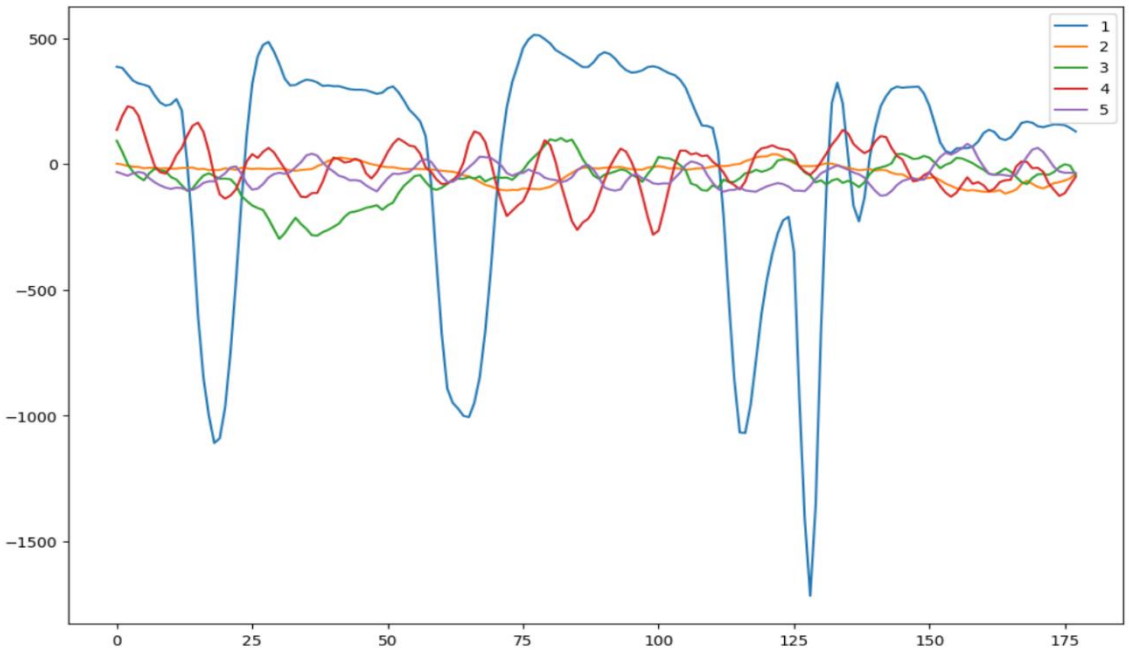

Fig 4.10 Range value for each class

CHAPTER 5

PERFORMANCE ANALYSIS

The effectiveness of this deep learning model can be assessed using the accuracy metric. One parameter used to assess classification models is accuracy. Roughly speaking, accuracy is the percentage of correct predictions our model made. The following is the formal definition of accuracy:

$$Accuracy = \frac{Correct\ Predictions}{All\ Predictions}$$

Accuracy for binary classification can also be computed as follows in terms of positives and negatives:

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN}$$

Where *TP* = True Positives, *TN* = True Negatives, *FP* = False Positives, and *FN* = False Negatives.

CHAPTER 6

RESULTS AND DISCUSSION

In summary, the analysis of the epileptic seizure detection using advanced machine learning techniques when compared with LSTM, BiLSTM, GRU, BiGRU, RNN gave out the best accuracy of all the compared results. The LSTM model gave an accuracy of 67%, the GRU model gave an accuracy of 69.02%, the BiLSTM model gave an accuracy of 66%, the BiGRU model gave an accuracy of 70% and the RNN gave an accuracy of 70.2%.

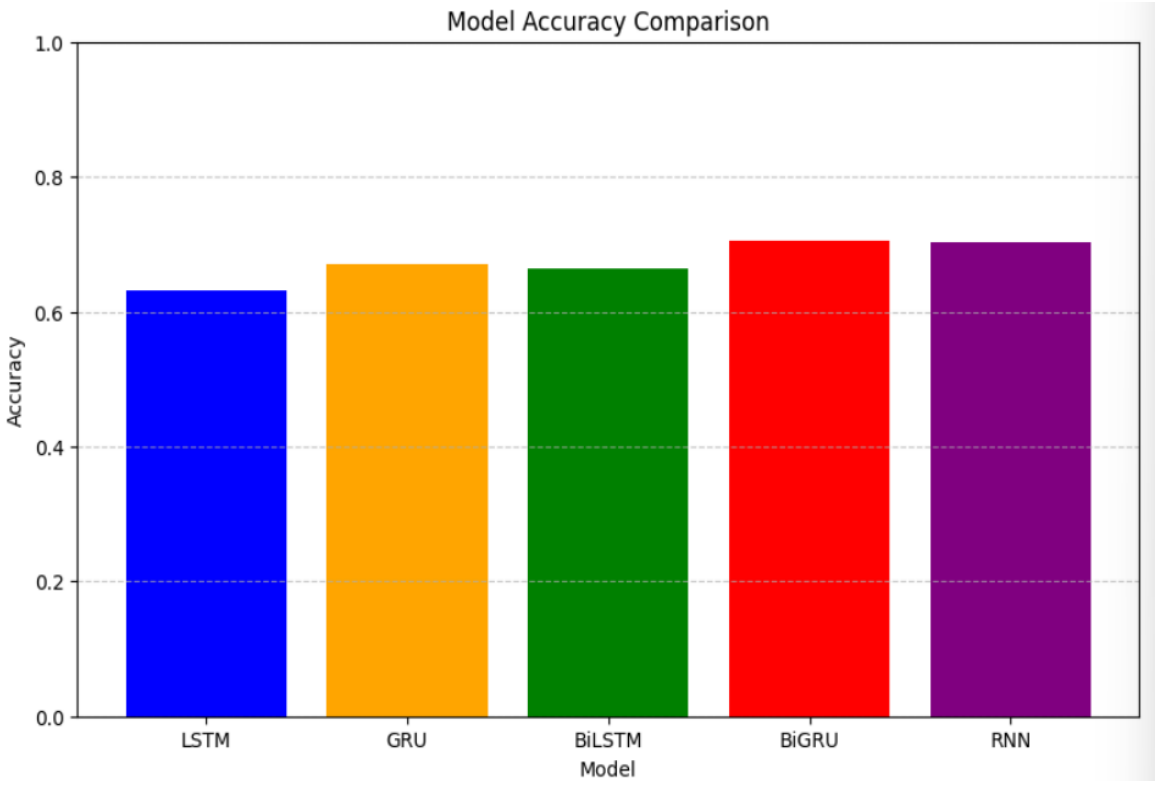

Fig.6.1 Comparison of different DL models

CHAPTER 7

CONCLUSION

The utilization of advanced machine learning techniques for epileptic seizure detection holds significant promise in revolutionizing the management and treatment of epilepsy. Through the analysis of electroencephalogram (EEG) signals, these techniques enable accurate and real-time detection of seizures, providing timely interventions and improving patient outcomes. The implementation of machine learning models such as Long Short-Term memory, Bidirectional Gated recurrent units, Gated recurrent units has demonstrated impressive performance in identifying seizure events with high sensitivity and specificity. These models leverage intricate patterns and features extracted from EEG data, allowing for robust classification of seizure and non-seizure states. However, challenges such as data variability, model generalization across diverse patient populations, and real-world deployment remain areas of active research and development. Addressing these challenges requires interdisciplinary collaboration between clinicians, data scientists, and engineers to refine existing algorithms and optimize their integration into clinical practice. The integration of advanced machine learning techniques for epileptic seizure detection represents a significant advancement in epilepsy management, offering the potential to transform patient care by providing timely interventions and personalized treatment strategies. Continued research and innovation in this field hold the key to further enhancing the accuracy, reliability, and accessibility of seizure detection systems, ultimately improving the lives of individuals living with epilepsy.

CHAPTER 8

APPENDICES

8.1 APPENDIX – 1 CODING

Uploading the dataset

```
import os
import pandas as pd

# Define the source folder containing the subfolders
source_folder = "/content/drive/MyDrive/Dataset_epilepsy"

# Initialize an empty list to store EEG data
eeg_data_list = []

# Initialize an empty list to store target values
target_list = []

# Iterate over the subfolders
for folder_name in ["Z", "F", "N", "O", "S"]:
    folder_path = os.path.join(source_folder, folder_name)
    # Iterate over the files in the subfolder
    for file_name in os.listdir(folder_path):
        file_path = os.path.join(folder_path, file_name)
        # Read EEG data from the text file
        with open(file_path, 'r') as file:
            lines = file.readlines()
            eeg_data = [float(value) for line in lines for value in line.strip().split()]
            # Ensure the EEG data has 4096 elements
            eeg_data = eeg_data[:4096] if len(eeg_data) >= 4096 else eeg_data + [0.0] * (4096 -
len(eeg_data))
            # Append EEG data to the list
            eeg_data_list.append(eeg_data)
```

```
            # Append target value to the list
            target_list.append(folder_name)

# Create DataFrame from EEG data
eeg_df = pd.DataFrame(eeg_data_list)

# Add target column
eeg_df['target'] = target_list
```

```
# Write DataFrame to CSV
csv_file_path = "combined_eeg_data.csv"
eeg_df.to_csv(csv_file_path, index=False)

print("CSV file saved to:", csv_file_path)

df=pd.read_csv(csv_file_path)
df
```

Checking for missing values

```
missing_values=df.isnull().sum()
missing_values
```

Checking the class distribution

```
class_counts = df['y'].value_counts()
plt.figure(figsize=(8, 6))
class_counts.plot(kind='bar', color='skyblue')
plt.title('Class Distribution')
plt.xlabel('Class')
plt.ylabel('Frequency')
plt.show()
```

Standard Scaling and encoding

```
from sklearn.preprocessing import StandardScaler
X = df.drop(columns=['y'])
y = df['y']

scaler = StandardScaler()
X_standardized = scaler.fit_transform(X)
```

```
from sklearn.preprocessing import LabelEncoder
label_encoder = LabelEncoder()
y_encoded = label_encoder.fit_transform(y)
```

LSTM Model

```
from tensorflow.keras import layers, Model
def LSTM(input_shape, num_classes):
    inputs = layers.Input(shape=input_shape)
    x = layers.LSTM(64, return_sequences=True)(inputs)  # First LSTM layer
    x = layers.LSTM(64, return_sequences=True)(x)  # Second LSTM
    x = layers.LSTM(64, return_sequences=True)(x)  # Third LSTM layer
    x = layers.LSTM(64, return_sequences=True)(x)  # Fourth LSTM layer
    x = layers.LSTM(64)(x)  # Fifth LSTM layer
    outputs = layers.Dense(num_classes, activation='softmax')(x)
    model = Model(inputs, outputs)
    return model

lstm_model = LSTM(input_shape, num_classes)
lstm_model.compile(optimizer='adam', loss='sparse_categorical_crossentropy',
metrics=['accuracy'])
```

```
lstm_model.summary()
lstm_history = lstm_model.fit(X_train_reshaped, y_train, epochs=50, batch_size=32,
validation_split=0.2)
lstm_loss, lstm_accuracy = lstm_model.evaluate(X_test_reshaped, y_test)
print("LSTM Model Accuracy:", lstm_accuracy)
```

BiLSTM Model

```
from tensorflow.keras.layers import Bidirectional

def BiLSTM(input_shape, num_classes):
    inputs = layers.Input(shape=input_shape)
    x = Bidirectional(layers.LSTM(64, return_sequences=True))(inputs)  # Bidirectional LSTM
layer
    x = layers.LSTM(64, return_sequences=True)(x)  # Second LSTM layer
    x = layers.LSTM(64, return_sequences=True)(x)  # Third LSTM layer
    x = layers.LSTM(64, return_sequences=True)(x)  # Fourth LSTM layer
    x = layers.LSTM(64)(x)  # Fifth LSTM layer
    outputs = layers.Dense(num_classes, activation='softmax')(x)
    model = Model(inputs, outputs)
    return model

bilstm_model = BiLSTM(input_shape, num_classes)
bilstm_model.compile(optimizer='adam', loss='sparse_categorical_crossentropy',
metrics=['accuracy'])
bilstm_model.summary()
bilstm_history = bilstm_model.fit(X_train_reshaped, y_train, epochs=50, batch_size=32,
validation_split=0.2)
bilstm_loss, bilstm_accuracy = bilstm_model.evaluate(X_test_reshaped, y_test)



print("BiLSTM Model Accuracy:", bilstm_accuracy)
```

GRU Model

```
def GRUModel(input_shape, num_classes):
    inputs = layers.Input(shape=input_shape)
    x = layers.GRU(64, return_sequences=True)(inputs)  # First GRU layer
    x = layers.GRU(32, return_sequences=True)(x)  # Second GRU layer
    x = layers.GRU(64)(x)  # Third GRU layer
    outputs = layers.Dense(num_classes, activation='softmax')(x)
    model = Model(inputs, outputs)
    return model


gru_model = GRUModel(input_shape, num_classes)
gru_model.compile(optimizer='adam', loss='sparse_categorical_crossentropy',
metrics=['accuracy'])
gru_model.summary()

gru_history = gru_model.fit(X_train_reshaped, y_train, epochs=50, batch_size=32,
validation_split=0.2)
gru_loss, gru_accuracy = gru_model.evaluate(X_test_reshaped, y_test)
print("GRU Model Accuracy:", gru_accuracy)
```

BiGRU Model

```
def BiGRUModel(input_shape, num_classes, use_dropout=False, dropout_rate=0.2,
use_l1_l2=False, l1=1e-5, l2=1e-5):
    inputs = layers.Input(shape=input_shape)
    x = layers.Bidirectional(layers.GRU(64, return_sequences=True),
merge_mode='concat')(inputs)
    if use_dropout:
        x = layers.Dropout(dropout_rate)(x)
    if use_l1_l2:
        kernel_regularizer = regularizers.l1_l2(l1=l1, l2=l2)
    else:
        kernel_regularizer = None
```

```
    x = layers.GRU(64, return_sequences=False, kernel_regularizer=kernel_regularizer)(x)
    outputs = layers.Dense(num_classes, activation='softmax')(x)
    model = Model(inputs, outputs)
    return model

bigru_model = BiGRUModel(input_shape, num_classes)
bigru_model.compile(optimizer='adam', loss='sparse_categorical_crossentropy',
metrics=['accuracy'])

bigru_model.summary()
bigru_history = bigru_model.fit(X_train_reshaped, y_train, epochs=75, batch_size=32,
validation_split=0.2)
bigru_loss, bigru_accuracy = bigru_model.evaluate(X_test_reshaped, y_test)
print("BiGRU Model Accuracy:", bigru_accuracy)
```

RNN Model

```
from tensorflow.keras.layers import Input, GRU, Dense, Bidirectional, Dropout
from tensorflow.keras.models import Model
from sklearn.model_selection import train_test_split
import numpy as np


def create_rnn_model(input_shape, num_classes, use_dropout=False, dropout_rate=0.2,
use_l1_l2=False, l1=1e-5, l2=1e-5):
    inputs = Input(shape=input_shape)
    x = Bidirectional(GRU(64, return_sequences=True), merge_mode='concat')(inputs)
    if use_dropout:
        x = Dropout(dropout_rate)(x)
    if use_l1_l2:
        kernel_regularizer = tf.keras.regularizers.l1_l2(l1=l1, l2=l2)
    else:
        kernel_regularizer = None
```

```
    x = GRU(64, return_sequences=False, kernel_regularizer=kernel_regularizer)(x)
    outputs = Dense(num_classes, activation='softmax')(x)
    model = Model(inputs, outputs)
    return model
rnn_model = create_rnn_model(input_shape, num_classes)
rnn_model.compile(optimizer='adam', loss='sparse_categorical_crossentropy',
metrics=['accuracy'])
rnn_model.summary()


rnn_history = rnn_model.fit(X_train_reshaped, y_train, epochs=75, batch_size=32,
validation_split=0.2)
rnn_loss, rnn_accuracy = rnn_model.evaluate(X_test_reshaped, y_test)
print("RNN Model Accuracy:", rnn_accuracy)
```

8.2 APPENDIX – 2 SCREENSHOTS

```
Model: "model_1"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 input_2 (InputLayer)        [(None, 1, 178)]          0

 lstm_18 (LSTM)              (None, 1, 64)             62208

 lstm_19 (LSTM)              (None, 1, 64)             33024

 lstm_20 (LSTM)              (None, 1, 64)             33024

 lstm_21 (LSTM)              (None, 1, 64)             33024

 lstm_22 (LSTM)              (None, 64)                33024

 dense_14 (Dense)            (None, 5)                 325

=================================================================
Total params: 194629 (760.27 KB)
Trainable params: 194629 (760.27 KB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
```

Fig A2.1: Summary of LSTM Model

```
Model: "model_3"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 input_4 (InputLayer)        [(None, 1, 178)]          0

 bidirectional_1 (Bidirecti  (None, 1, 128)            124416
 onal)

 lstm_29 (LSTM)              (None, 1, 64)             49408

 lstm_30 (LSTM)              (None, 1, 64)             33024

 lstm_31 (LSTM)              (None, 1, 64)             33024

 lstm_32 (LSTM)              (None, 64)                33024

 dense_16 (Dense)            (None, 5)                 325

=================================================================
Total params: 273221 (1.04 MB)
Trainable params: 273221 (1.04 MB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
```

Fig A2.2: Summary of BiLSTM Model

```
Model: "model_4"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 input_5 (InputLayer)        [(None, 1, 178)]          0

 gru (GRU)                   (None, 1, 64)             46848

 gru_1 (GRU)                 (None, 1, 32)             9408

 gru_2 (GRU)                 (None, 64)                18816

 dense_17 (Dense)            (None, 5)                 325

=================================================================
Total params: 75397 (294.52 KB)
Trainable params: 75397 (294.52 KB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
```

Fig A2.3: Summary of GRU Model

```
Model: "model_6"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 input_7 (InputLayer)        [(None, 1, 178)]          0

 bidirectional_3 (Bidirecti  (None, 1, 128)            93696
 onal)

 gru_6 (GRU)                 (None, 64)                37248

 dense_19 (Dense)            (None, 5)                 325

=================================================================
Total params: 131269 (512.77 KB)
Trainable params: 131269 (512.77 KB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
```

Fig A2.4: Summary of BiGRU Model

```
Model: "model_7"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 input_8 (InputLayer)        [(None, 1, 178)]          0

 bidirectional_4 (Bidirecti  (None, 1, 128)            93696
 onal)

 gru_8 (GRU)                 (None, 64)                37248

 dense_20 (Dense)            (None, 5)                 325

=================================================================
Total params: 131269 (512.77 KB)
Trainable params: 131269 (512.77 KB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
```

Fig A2.5: Summary of RNN Model

REFERENCES

1. Mekruksavanich, S., & Jitpattanakul, A. (2023). Effective Detection of Epileptic Seizures through EEG Signals Using Deep Learning Approaches. *Machine Learning and Knowledge Extraction*, *5*(4), 1937-1952.
2. Abiyev, R., Arslan, M., Bush Idoko, J., Sekeroglu, B., & Ilhan, A. (2020). Identification of epileptic EEG signals using convolutional neural networks. *Applied sciences*, *10*(12), 4089.
3. Acharya, U. R., Oh, S. L., Hagiwara, Y., Tan, J. H., & Adeli, H. (2018). Deep convolutional neural network for the automated detection and diagnosis of seizure using EEG signals. *Computers in biology and medicine*, *100*, 270-278.
4. Ghayab, H. R. A., Li, Y., Abdulla, S., Diykh, M., & Wan, X. (2016). Classification of epileptic EEG signals based on simple random sampling and sequential feature selection. *Brain informatics*, *3*, 85-91.
5. Aayesha, Qureshi, M. B., Afzaal, M., Qureshi, M. S., & Fayaz, M. (2021). Machine learning-based EEG signals classification model for epileptic seizure detection. *Multimedia Tools and Applications*, *80*(12), 17849-17877.
6. Kołodziej, M., Majkowski, A., & Rysz, A. (2023). Implementation of machine learning and deep learning techniques for the detection of epileptic seizures using intracranial electroencephalography. *Applied Sciences*, *13*(15), 8747.