ENHANCING DYSARTHRIA DIAGNOSIS WITH DEEP LEARNING TECHNIQUES

A PROJECT WORK I REPORT

Submitted by

NITHIKA K
21ALR029

DHANUSH T
21ALR013

ARVIND CB
21ALR007

in partial fulfilment of the requirements

for the award of the degree of

BACHELOR OF TECHNOLOGY

IN

ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING

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 **ENHANCING DYSARTHRIA DIAGNOSIS WITH DEEP LEARNING TECHNIQUE** is the bonafide the record of project work done by **NITHIKA K (21ALR029), DHANUSH T (21ALR013), ARVIND C B (21ALR007)** in partial fulfilment of the requirements for the award of Degree of Bachelor of Technology in Artificial Intelligence of Kongu Engineering College, Perundurai 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 **ENHANCING DYSARTHRIA DIAGNOSIS WITH DEEP LEARNING TECHNIQUES** 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 any other 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:

NITHIKA K

(21ALR029)

DHANUSH T

(21ALR013)

ARVIND C B

(21ALR007)

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

A complex speech disorder called dysarthria is brought on by a nerve ailment. The precise muscles implicated and the underlying cause can greatly influence the symptoms of dysarthria. Talk therapy is the typical course of treatment for dysarthria. Improved treatment outcomes and early intervention are possible when dysarthria is identified early. This research suggests using a model based on convolutional neural networks (CNNs) to detect dysarthria. The objective is to use audio data to construct a dysarthria detection system. To show how the speech patterns of people with dysarthria and those without it differ, a variety of audio visualization techniques are employed, along with the retrieval and investigation of audio data. Creating different neural network models, such as Convolutional Neural Networks (CNN), Long Short Term Memory (LSTM), Gated Recurrent Units (GRU), Bidirectional LSTM, SimpleRNN, and Deep Neural Networks (DNN), and extracting features using Mel-frequency cepstral coefficients (MFCC). Using the MFCC features that were retrieved, these models are trained to distinguish between dysarthric and non-dysarthric speech. The trained models are assessed for dysarthria detection using a variety of measures, including precision, recall, ROC-AUC scores, classification reports, and matrix matrices. Visualizations that shed light on the behavior and effectiveness of the models include confusion matrices, ROC curves, and loss and accuracy curves. All in all, this script embodies an all-inclusive pipeline for dysarthria identification, starting with data preprocessing and concluding with model training and assessment through cutting edge deep learning methodologies and audio feature extraction approaches. The ultimate objective is to develop a trustworthy dysarthria detection system that will help medical practitioners identify and treat the condition early.

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.KANIMOZHI SELVI ME., Ph.D.,** for his valuable guidance.

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

We extend our sincere gratitude to our beloved guide **Ms. S. KEERTHIKA M.E.,** 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 TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

DL	:	Deep Learning
CNN	:	Convolutional Neural Network
LSTM	:	Long Short-Term Memory
GRU	:	Gated Recurrent Unit
BiLSTM	:	Bidirectional LSTM
RNN	:	Recurrent Neural Network
DNN	:	Deep Neural Network
MFCC	:	Mel-frequency cepstral coefficients
TP	:	True Positive
TN	:	True Negative
FP	:	False Positive
FN	:	False Negative

CHAPTER 1

INTRODUCTION

1.1 INTRODUCTION

Effective communication is greatly impacted by dysarthria, a motor speech disease marked by poor articulation, phonation, and prosody. Millions of people worldwide are impacted by neurological disorders like stroke, Parkinson's disease, cerebral palsy, and traumatic brain injury. For effective intervention and rehabilitation measures to improve communication and quality of life for affected individuals, early detection and precise assessment of dysarthria are essential.

Conventional approaches to evaluating dysarthria frequently depend on the subjective assessments of speech-language pathologists (SLPs) using standardized assessment instruments and clinical observations. These techniques are useful, but they can also be laborious, subjective, and insensitive to minute variations in speech patterns. Furthermore, there may be a lack of access to qualified medical specialists for evaluation, especially in underprivileged areas or during emergencies.

Prospective paths for automating dysarthria assessment and identification are presented by recent developments in machine learning and signal processing. Automated systems can supplement conventional assessment methods by using computational approaches to analyze speech signals and extract essential information. This can result in faster, more objective, and scalable solutions.

This study employs deep learning models and audio feature extraction techniques to detect dysarthria in a novel way. Convolutional neural networks (CNNs), recurrent neural networks (RNNs), and dense neural networks (DNNs) are among the deep learning architectures that are trained and assessed using a dataset made up of audio recordings from people who have dysarthria and those who do not. Furthermore, the use of Mel-frequency cepstral coefficients (MFCCs) as feature representations of speech signals is carried out, collecting crucial attributes for the categorization of dysarthria.

Through testing the suggested framework's performance on a publicly accessible collection of speech samples from people with and without dysarthria, and contrasting it with baseline models. The outcomes indicate how well deep scalable and easily available methods for diagnosing and screening for dysarthria by applying learning methodologies to effectively identify dysarthria from speech signals.

This study aims to demonstrate the effectiveness of deep learning models in automated dysarthria detection and provide insights into the discriminatory power of different architectures and feature representations. By developing robust and accurate dysarthria detection systems, we aim to facilitate early intervention and personalized rehabilitation strategies for individuals affected by this debilitating condition.

1.2 OBJECTIVE

- Data collection: Gather a wide range of speech samples from both non-dysarthric controls and people with a diagnosis of dysarthria.
- Feature extraction : It involves taking out pertinent acoustic elements from the speech signals, such as pitch, zero-crossing rate, spectral centroid, and Mel-frequency cepstral coefficients (MFCCs).
- Model Development: Using the features that were collected, create and apply deep learning models to categorize speech samples as dysarthric or non-dysarthric.
- Model assessment: Use appropriate assessment metrics, such as accuracy, sensitivity, specificity, and area under the receiver operating characteristic curve (AUC-ROC), to evaluate the performance of the generated models.
- Contrast to baseline models: Examine how well our suggested models perform in comparison to baseline classifiers and other methods currently used for dysarthria detection.
- Clinical Validation: Examine and compare the automated dysarthria detection system's results with professional evaluations from speech-language pathologists in order to confirm its efficacy.

1.3 SCOPE

It works to build a dependable system for audio recording-based dysarthria detection. To determine which deep learning model is the most accurate, a number of them must be tested. In order to improve dysarthria diagnosis and therapy, the project will also examine which audio aspects are most crucial for detection and create a useful tool that can be applied in clinical or telemedicine contexts.

CHAPTER 2

SYSTEM ANALYSIS

2.1 LITERATURE REVIEW

In 2023, this study [1] tackles the problem of using deep learning techniques to identify dysarthria, a speech issue linked to neurological abnormalities. The suggested method outperforms conventional machine learning models in terms of accuracy by using pre-trained convolutional neural networks (CNNs) with transfer learning to convert time-domain speech data into scalogram images via wavelet transformation. The technique enables early intervention for individuals with dysarthria and improves classification performance by capturing spectral properties at several scales.

In 2023, dysarthria—a speech [2] problem marked by poor motor control—poses difficulties for early identification and frequently points to underlying medical conditions. In order to help with early diagnosis and enhance disability management, this research suggests a model for dysarthria identification based on Convolutional Neural Networks (CNNs). Using a variety of features, including zero crossing rates, Mel-frequency cepstral coefficients (MFCCs), spectral centroids, and spectral roll off, the suggested model analyzes speech data. Although dysarthria may indicate a number of underlying illnesses, such as brain injury or stroke, the suggested approach just considers dysarthria detection as a symptom. It seeks to ascertain whether a speech sample originates from a patient with dysarthria, offering a validation accuracy performance score. The suggested model makes use of the TORGO database, which includes samples from people who speak normally and those who have dysarthria. This could lead to better methods for managing impairments and better dysarthria identification.

In 2022 has seen an increase in the prevalence of dysarthria, a speech problem that affects people with disorders including Parkinson's disease and stroke. This is due to the growth in neurological diseases [3]. Although prompt diagnosis and treatment are essential, conventional diagnostic techniques are expensive and time-consuming.

This work presents a CNN-GRU model that detects dysarthria with an astounding 98.38% accuracy. This model outperforms earlier methods by utilizing deep learning techniques and STFT and MFCC speech signal processing. Its accomplishments highlight how deep learning may be used to efficiently and objectively assess dysarthria, enabling better patient care and disease management.

A comparison research on the categorization of dysarthria severity levels using various deep learning techniques and acoustic characteristics is presented in this paper [4] in 2022. The study uses fundamental speech features such as Mel-frequency cepstral coefficients (MFCCs) and constant-Q cepstral coefficients to evaluate different deep learning architectures, such as deep neural networks (DNNs), convolutional neural networks (CNNs), gated recurrent units (GRUs), and long short-term memory networks (LSTMs). Furthermore, prosody, articulation, phonation, and glottal functional features particular to speech disorders are investigated on DNN models. Additionally, the usefulness of utilizing subspace modeling to obtain i-vectors from low-dimensional feature representation is examined. The usual UA-Speech and TORGO databases are used for evaluation. According to the results, the DNN classifier that uses MFCC-based i-vectors outperforms other systems and gets promising accuracy scores.Dysarthric patients find it difficult to communicate since their speech is characterized by imprecise articulation, low audibility, abnormal prosody, and variable speech rate, even in cases when their syntax is perfect. Due to physical limitations, existing interactive apps based on keyboards or joysticks are less helpful; therefore, applications based on automated speech recognition (ASR) that are customized for dysarthric speakers are needed. However, when employed by dysarthric speakers, existing ASRs created for healthy speakers have substantial error rates, mostly because of the weak articulations.

The work [5] from 2022 discusses the problem of dysarthric speech recognition (DSR), which is still challenging since dysarthric speech is incomprehensible and has irregular phoneme articulation. Previous methods frequently rely on automated speech recognition (ASR) systems that have been trained on speech that is not affected, leading to subpar performance. In order to address this, the study compares a regular convolutional neural network (CNN) model with a deep convolutional recurrent neural network (CRNN) model for DSR.

A neuro-motor articulation ailment called dysarthria causes weakening of the speech muscles and is frequently linked to diseases including cerebral palsy, Parkinson's disease, and stroke. Slower and with erratic phoneme articulation, dysarthric speech makes communication difficult and fosters social isolation.

Acoustic and articulatory properties of traditional ASR systems are mismatched, making them inefficient for DSR. These systems are trained on unimpaired speech. Consequently, customized DSR systems that have been trained on datasets that include both normal and dysarthric speech are crucial. Previous research has mostly employed methods such as hidden Markov models (HMM) and Gaussian mixture models (GMM) to extract acoustic features; however, deep learning approaches have the advantage of modeling long-term dependencies in speech signals. The suggested CRNN model successfully captures the traits of dysarthric speech by combining CNNs for local feature extraction and RNNs for global feature aggregation. As opposed to the RNN, the CNN serves as a feature extractor that enhances DSR performance by integrating these characteristics.

2.2 SUMMARY

The overall goal of these investigations is to enhance the identification and categorization of dysarthria, a speech impairment frequently linked to neurological diseases. They evaluate voice signals and extract pertinent features for precise evaluation using deep learning algorithms. Deep CRNN models for dysarthric speech recognition, CNN-GRU models, wavelet transformation, and CNN-based analysis of speech features are some of the suggested techniques. Through the utilization of sophisticated computational techniques, these investigations exhibit encouraging outcomes in terms of improving the precision of dysarthria identification and enabling prompt intervention, ultimately leading to improved patient care and management tactics. These studies address the difficulty of detecting dysarthria, a speech problem common in individuals with disorders including Parkinson's disease and stroke, in response to the growth in neurological diseases.

Because conventional diagnostic techniques are frequently expensive and time-consuming, cutting-edge deep learning strategies are being investigated.These studies produce outstanding accuracy rates in dysarthria diagnosis by utilizing approaches including CNN-based analysis, wavelet transformation, MFCC, and STFT. Additionally, they stress how crucial early identification and intervention are to bettering impairment management techniques and patient outcomes. These research provide a road towards more effective and objective dysarthria examinations through the incorporation of deep learning techniques, which will ultimately benefit patients and medical professionals.

CHAPTER 3

SYSTEM REQUIREMENTS

3.1 HARDWARE REQUIREMENTS

CPU type : Intel corei5

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 an easy-to-learn, powerful programming language. Its broad application of information structures and simple yet efficient method of managing object-oriented programming. Python's sophisticated phrase structure and imperative composition, which are close to its made sense character, make it an ideal language for prearranging and quick application development in numerous fields at the best stages. This essay covers Python 3's fundamental concepts and features. Python's standard library is made up of many of the functions that are installed along with the program. The Python programming language may experiment with a wide range of supplementary libraries to explore a greater variety of web content. These libraries provide it power and make it capable of performing a variety of tasks. A number of crucial libraries and techniques are used in the given Python code snippets to accomplish a variety of tasks. Data preprocessing is made possible by the Pandas package, which is used for data manipulation and analysis, especially when reading and processing data from CSV files. As a machine learning library, Scikit-Learn (sklearn) is used to help with model selection, data partitioning for training and testing, and accuracy score computation.

3.3.2 Google Colab Notebook

Google Colab, sometimes known as Google Colaboratory, is a cloud-based platform that makes it simple for users to execute and share code by giving free access to Jupyter notebooks. It provides a deep learning, machine learning, and data analysis environment powered by GPUs and TPUs. Users can keep their work on Google Drive and collaborate in real time. The sharing of Colab notebooks is similar to that of Sheets or Google Docs. Either follow these instructions for sharing files from Google Drive, or click the Share button in the top right corner of any Colab notebook. For Colab laptops, Google Drive search functions are provided. Clicking the Colab logo in the upper left corner of the notebook view will make all of the notebooks in Drive visible. You can also find notebooks that you have recently opened by choosing File > Open Notebook. Code is executed on your account-only virtual computer. The Colab service enforces a maximum lifetime for virtual computers, beyond which they are removed after a period of inactivity. You can access any Colab notebook you've made through the File menu in Colab, by following these instructions, or by downloading it from Google Drive.

CHAPTER 4

PROPOSED SYSTEM

4.1 SYSTEM ARCHITECTURE

The objective of the suggested system is to create a classifier model that successfully separates dysarthric speech from non-dysarthric speech, in addition to a robust model that can reliably detect speech impacted by dysarthria based on audio signal data. In order to do this, the first step is to compile a dataset containing important data regarding speech characteristics, temporal trends, and other relevant contextual elements. Preprocessing methods are used once the dataset is assembled to make sure it is appropriate for training the classification models. Essential temporal and spectral information are captured in frequency-based representations of raw audio signals by preprocessing techniques like Spectrogram, Mel Spectrogram, Zero Crossing Rate, and Spectral Rolloff. In simple terms, Mel Spectrogram visualizes the frequency content of the signal over time, Zero Crossing Rate measures changes in the signal's amplitude, and Spectral Rolloff estimates the frequency below which a certain percentage of the total spectral energy lies.

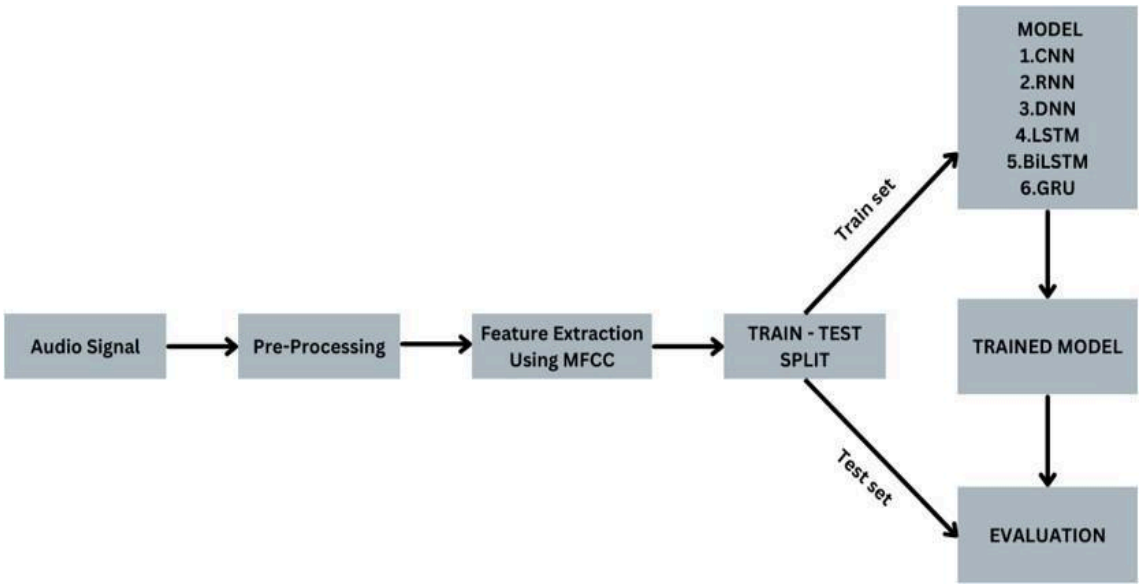

Fig. 4.1 Proposed model workflow

Then feature extraction is performed using Mel-frequency cepstral coefficients (MFCC) are widely used in audio signal processing for feature extraction due to their effectiveness in capturing key characteristics of the signal related to human perception.It involves dividing the audio signal into short frames, passing them through Mel filters to emphasize perceptually relevant frequencies, and then applying the Discrete CosineTransform (DCT) to obtain a set of coefficients. These coefficients capture essential characteristics such as pitch, timbre, and spectral envelope, providing a compact representation of the signal. MFCC is widely used in tasks like speech recognition and emotion detection due to its effectiveness in capturing relevant signal information.

After Feature extraction, six different deep learning techniques are employed to the audio signal dataset. These techniques include Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), Deep Neural Networks (DNNs), Long Short-Term Memory (LSTM), Bidirectional LSTM (BiLSTM), and Gated Recurrent Unit (GRU). Each of these methodologies presents unique advantages and limitations, and by employing a combination of techniques, the objective is to discern the most precise model for classifying dysarthric signals based on the extracted features. Through comprehensive evaluation using pertinent metrics such as classification reports, confusion matrices, and accuracy scores, we aim to ascertain the optimal classification model. This iterative approach allows us to leverage the strengths of each technique while mitigating their respective weaknesses, ultimately enhancing the robustness and reliability of the dysarthria classification system.

4.2 MODULE DESCRIPTION

4.2.1 Dataset Collection

The TORGO dysarthric articulation database includes matched controls from Kaggle and aligned acoustics and measured 3D articulatory features from speakers with cerebral palsy (CP) or amyotrophic lateral sclerosis (ALS), two of the most common causes of speech disability (Kent and Rosen, 2004). You may access the dataset at (https://www.kaggle.com/datasets/iamhungundji/dysarthria-detection). Two thousand samples for dysarthric males, dysarthric females, non-dysarthric males, and non-dysarthric females are included in this dataset.

Table 4.1 Folder description of the dataset

dysarthria_female	500 samples of dysarthric female audio recorded on different sessions.
dysarthria_male	500 samples of dysarthric male audio recorded on different sessions.
non _dysarthria _female	500 samples of non-dysarthric female audio recorded on different sessions.
non _dysarthria _male	500 samples of non-dysarthric male audio recorded on different sessions.

4.2.2 Data pre-processing

Spectrogram: A spectrogram is a graphic depiction of a signal's frequency spectrum that changes over time. It is produced by dividing the audio signal into brief parts and computing the Fourier Transform for each segment using the Short-Time Fourier Transform (STFT) method. After that, the spectrum is plotted across time, with color or grayscale used to indicate intensity. Spectrograms are crucial for applications like speech analysis, sound recognition, and music processing because they offer insightful information about the frequency content and temporal evolution of the signal.

Mel Spectrogram: A technique called a Mel Spectrogram can be used to transform unprocessed audio signals into a picture that shows the signal's frequency content over time. The audio stream is split up into brief frames, and the Discrete Fourier Transform (DFT) is calculated for each frame. After that, the spectrum is converted into the Mel scale, a pitch scale that is perceptually relevant. This transformation is useful for tasks like speech processing because it highlights key aspects of human auditory perception.

Zero Crossing Rate: A method for calculating the rate at which an audio signal changes sign is called zero crossing rate. It keeps track of how many times the signal crosses the zero-amplitude axis in a specific amount of time. The Zero Crossing Rate yields information on the signal's temporal properties, including its rate of amplitude change and the existence of voiced or silent periods. Tasks involving voice and audio processing, such as speaker diarization and speech recognition, frequently use it.

Spectral Rolloff: This measurement determines the frequency at which a specific proportion of the total spectral energy is present. It shows where the majority of the signal's energy is focused in the frequency spectrum. Spectral Rolloff is helpful in describing the signal's spectral shape and can reveal details about its timbral characteristics. It is frequently applied to jobs involving audio analysis, such as instrument recognition and sound classification.

4.2.3 Feature Extraction

Overview of the MFCC method:

Mel Filtering: The audio stream is split up into brief frames that overlap, usually lasting 20–40 milliseconds. After that, every frame is run through a succession of triangle filters that are uniformly spaced across the Mel frequency range. By highlighting frequencies that are more perceptually meaningful, these filters emulate the frequency response of the human hearing system.

Discrete Cosine Transform (DCT): To approximate the non-linear response of the human auditory system to sound intensity, the logarithm of the filter bank outputs is computed after filtering. The resulting filter bank energies are then subjected to the Discrete Cosine Transform (DCT), which produces a set of MFCC coefficients.

Coefficient Selection: A portion of the resulting MFCC coefficients are typically chosen for additional examination, with the succeeding coefficients—which represent the signal's spectral shape—being kept and the initial coefficient—which is connected to the signal's total energy—usually discarded. Depending on the application, different coefficients may be preserved at different times.

Feature Representation: The last set of MFCC coefficients provides a condensed picture of the audio signal, encapsulating crucial elements like spectral envelope, timbre, and pitch. These coefficients are appropriate for tasks like voice recognition because they are comparatively resilient to changes in speech and noise environments.

4.2.4 Implementation of DL techniques

In the implementation of deep learning techniques for dysarthria detection, we explore six distinct models:

4.2.4.1 Convolutional Neural Networks (CNNs)

- CNNs are useful for applications like dysarthria identification because they are good at extracting hierarchical features from input data.
- CNNs can be trained to recognize patterns and structures in spectrograms or other frequency-based representations of audio signals in order to diagnose dysarthria.
- CNNs may efficiently learn discriminative features for dysarthria classification by using pooling layers to down sample and retain critical information and convolutional filters to capture local characteristics.
- CNNs use spectrograms or other frequency-based representations of audio signals to find patterns and structures in order to detect dysarthria.

$$C_{i,j} = \sigma\left(\sum_{k,l} W_{k,l} \cdot I_{i+k,j+l} + b\right)$$

- $where\ C_{i,j}\ is\ the\ output\ at\ position\ (i,j)\ (i,j)$
- $W_{k,l}\ are\ the\ filter\ weights, I_{i+k,j+l}\ are\ the\ input\ values$
- $b\ is\ the\ bias\ term, \sigma\ is\ the\ activation\ function.$

4.2.4.2 Recurrent Neural Networks (RNNs):

- RNNs are helpful for jobs where temporal dependencies are crucial, like dysarthria identification, because they are well-suited for processing sequential data.
- RNNs may be trained to recognize long-term relationships and temporal patterns in audio inputs, which is useful for diagnosing dysarthria.
- Recurrent neural networks (RNNs) are able to accurately mimic the temporal dynamics of dysarthric speech by processing sequential input data recurrently and preserving internal states.

- Mathematically, the output of an RNN at time step *t* can be calculated as

$$h_t = \sigma(W_{ih}x_t + W_{hh}h_{t-1} + b_h)$$

- $where\ h_t\ is\ the\ hidden\ state\ at\ time\ t$
- $x_t\ is\ the\ input\ at\ time\ t$
- $W_{ih}\ and\ W_{hh}\ are\ weight\ matrices$
- $b_h\ is\ the\ bias\ term$
- $\sigma\ is\ the\ activation\ function.$

4.2.4.3 Deep Neural Networks (DNNs):

- DNNs are adaptable models that can figure out intricate relationships between input and output data.
- DNNs can be used to learn high-level representations of audio information taken from spectrograms or MFCCs in order to detect dysarthria.
- DNNs may accurately classify dysarthric speech signals by capturing complex patterns and correlations within the signals through the application of non-linear activation functions and the stacking of numerous layers of neurons.
- Mathematically, the output of a DNN layer can be calculated as:

$$y = \sigma(W \cdot x + b)$$

- $where\ y\ is\ the\ output$
- $W\ is\ the\ weight\ matrix$
- $x\ is\ the\ input$
- $b\ is\ the\ bias\ term$
- $\sigma\ is\ the\ activation\ function$

4.2.4.4 Long Short-Term Memory (LSTM):

- Specifically created to tackle the vanishing gradient issue, long-range dependencies in sequential data can be effectively modeled using LSTMs, a form of RNN.

- Long short-term memory (LSTM) models are able to acquire the temporal patterns and dependencies necessary for effectively categorizing dysarthric speech in the context of dysarthria detection.
- LSTMs can cope with input patterns of varying durations and preserve contextual information over time steps by utilizing memory cells and gating mechanisms.
- Mathematically, cell status updates and different gates (forget, output, and input) can be used to determine the output of an LSTM cell.

4.2.4.5 Bidirectional LSTM (BiLSTM)

- BiLSTMs handle input sequences both forward and backward, improving upon the capabilities of conventional LSTMs.
- Bidirectional context information is captured by BiLSTMs in dysarthria identification, which helps them comprehend the temporal dynamics and interdependence inherent in dysarthric speech.
- BiLSTMs can produce more thorough representations of audio signals and hence improve classification performance by simultaneously taking into account past and future information.

4.2.4.6 Gated Recurrent Unit (GRU):

- In contrast to LSTMs, GRUs are a different kind of RNN architecture that solves the vanishing gradient issue and simplifies the design.
- GRUs are comparable to LSTMs in dysarthria identification since they require less computational complexity and fewer parameters to capture temporal connections.
- GRUs are well-suited for dysarthria classification tasks because they can mimic long-range relationships in audio signals by including gating techniques to restrict the flow of information.

4.3 VISUALIZATION

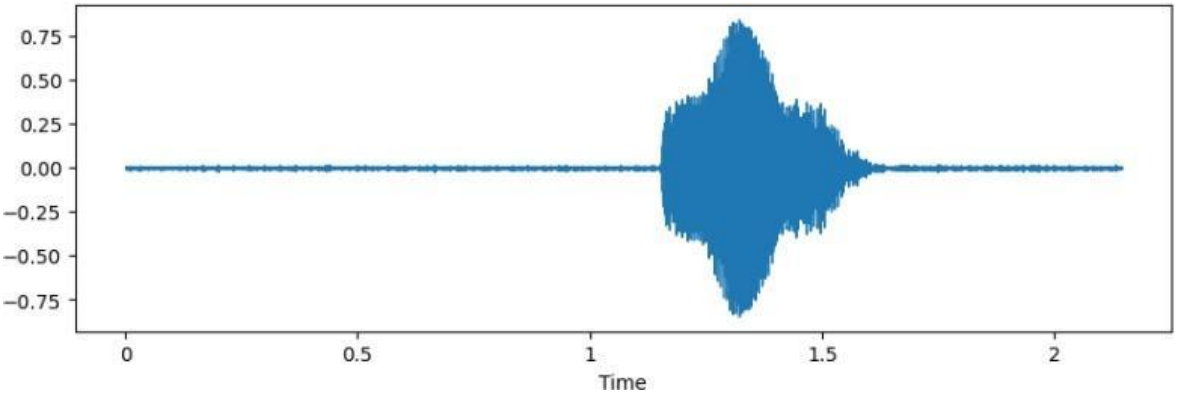

Fig 4.2 Wave plot of audio

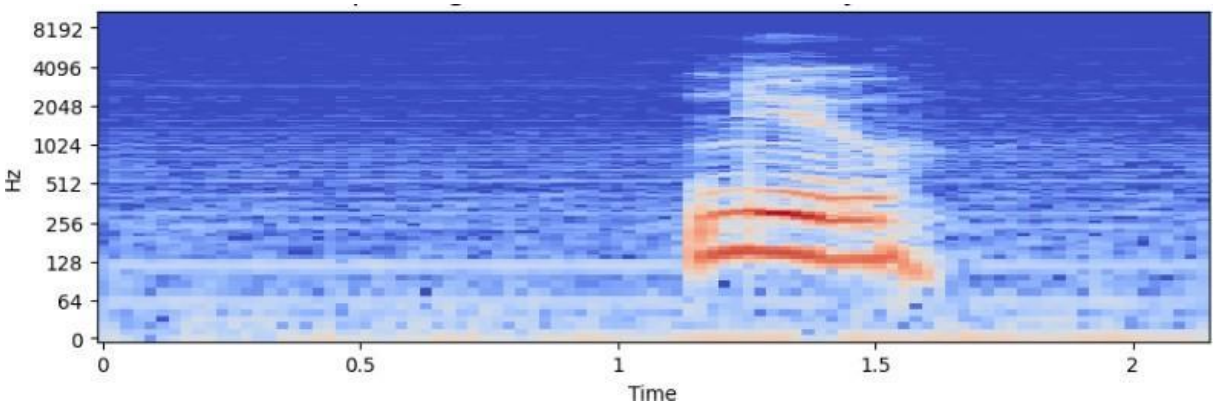

Fig 4.3 Spectrogram of audio

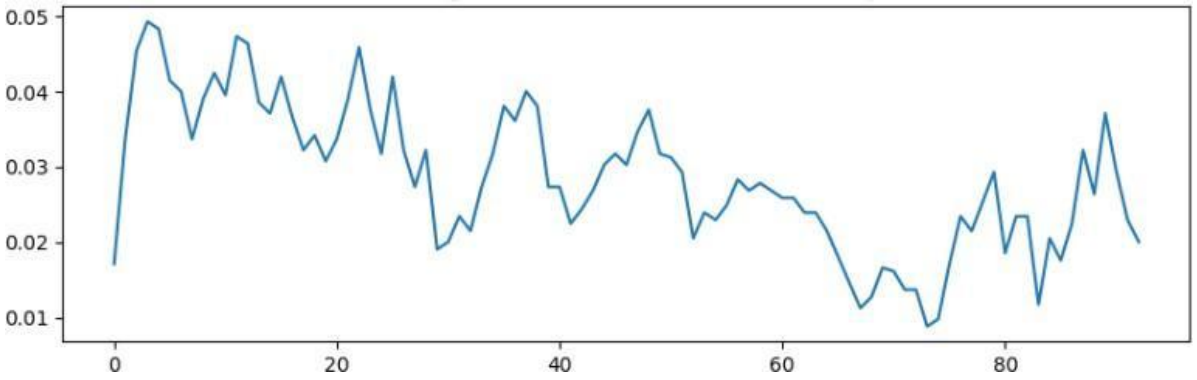

Fig 4.4 Zero Crossing Rate of audio

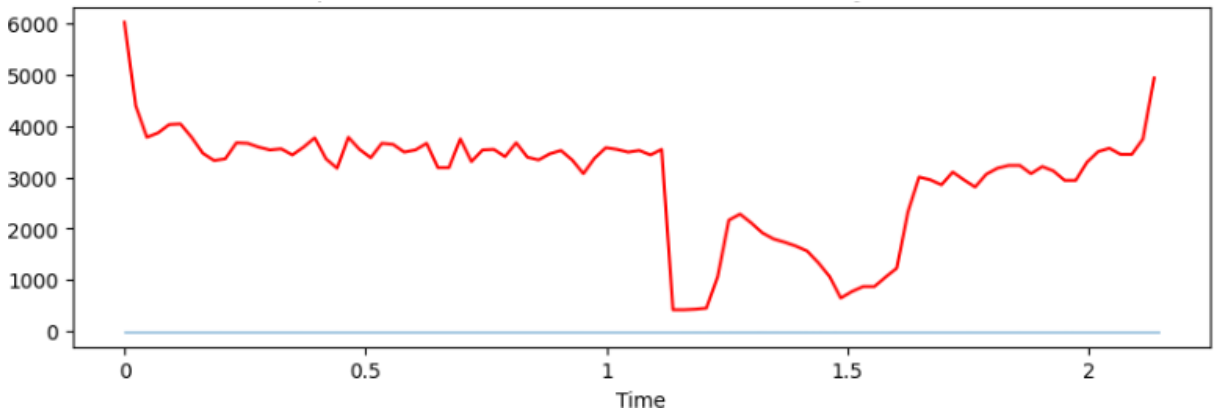

Fig 4.5 Spectral Rolloff of audio

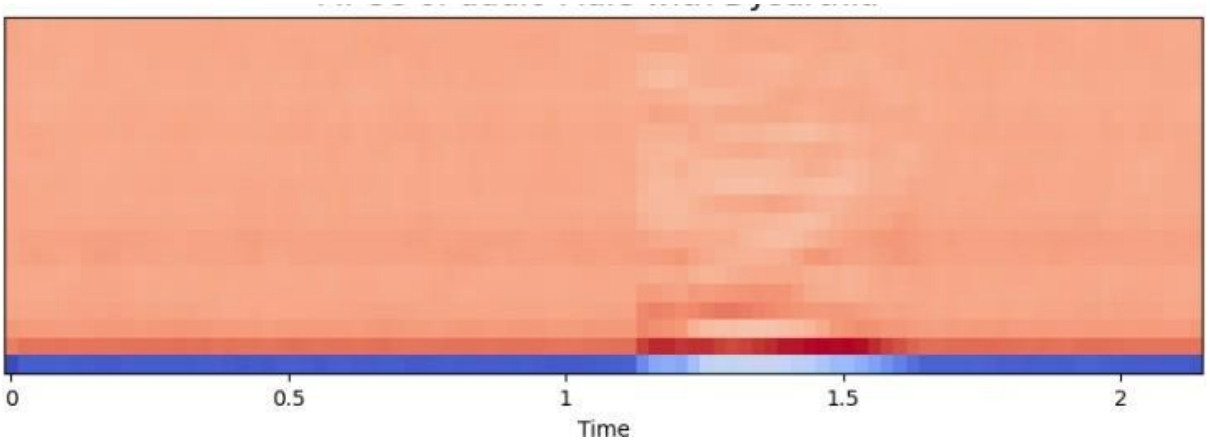

Fig 4.6 MFCC of audio

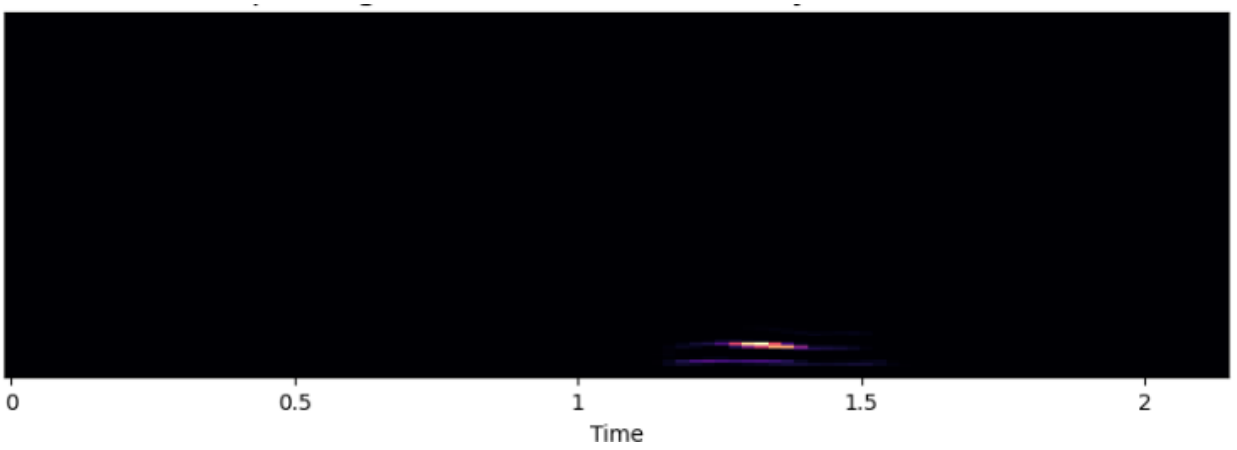

Fig 4.7 Mel Spectrogram of audio

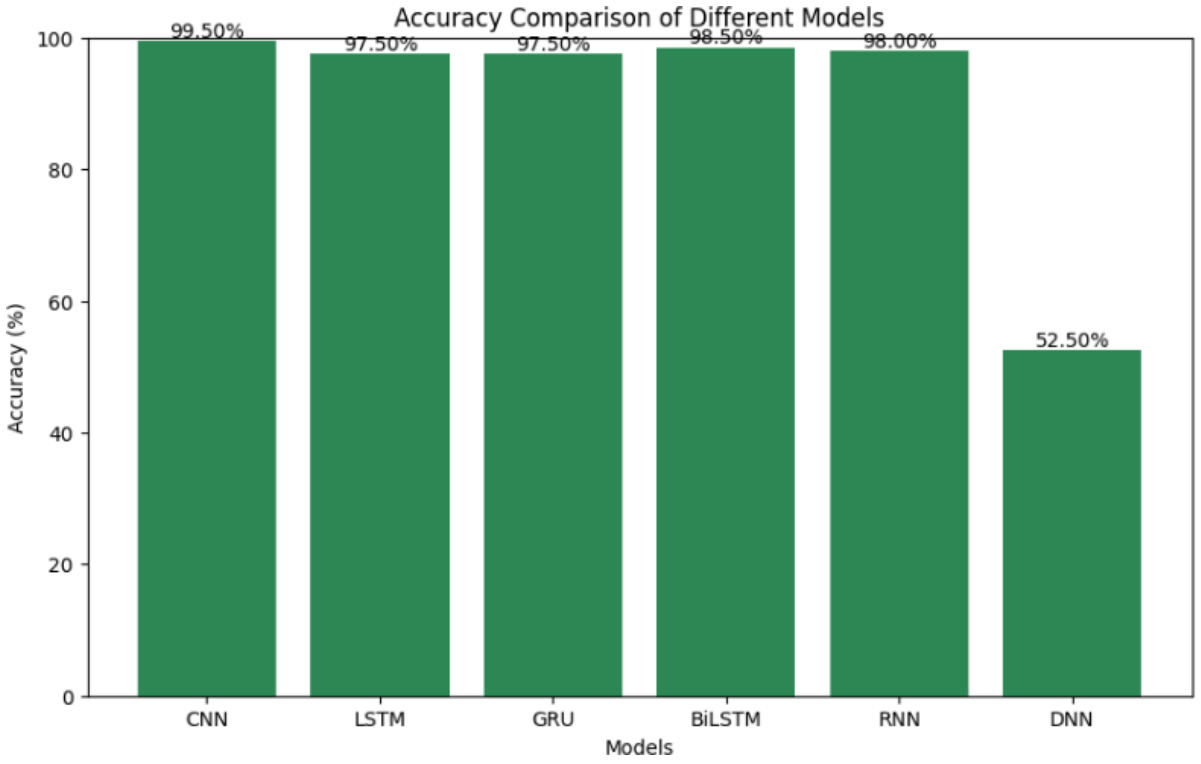

Fig 4.8 Accuracy of Different Models

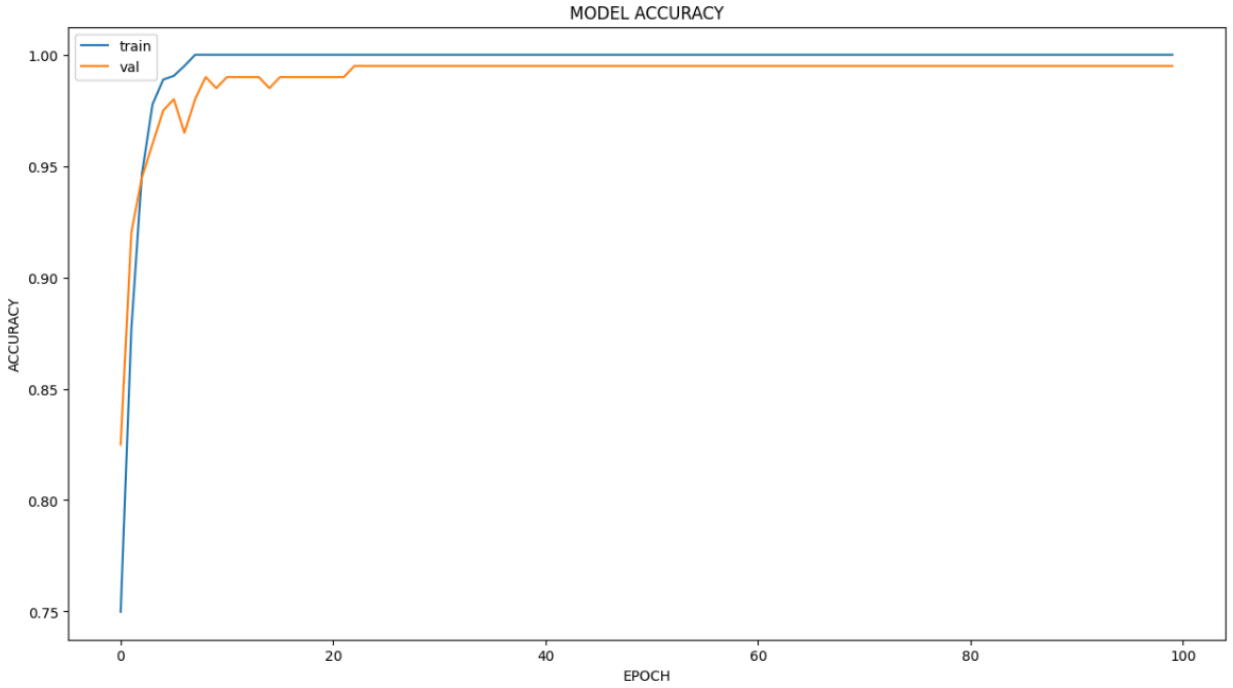

Fig 4.9 Accuracy curve of CNN

CHAPTER 5

PERFORMANCE ANALYSIS

The effectiveness of a classification model can be assessed using metrics like Accuracy, precision, recall, F-1 score, Classification report, Confusion matrix.

PRECISION:

The precision of a model is determined by dividing its total number of true positive predictions by its total number of false positive predictions.

$$Precision = \frac{TP}{TP + FP} \qquad (5.1)$$

RECALL:

Recall, sometimes referred to as sensitivity, illustrates a model's capacity to identify positive cases by calculating the percentage of genuine positive predictions among all real positive occurrences.

$$Recall = \frac{TP}{TP + FN} \qquad (5.2)$$

F1-SCORE:

The F1-Score is a balanced statistic that takes into account both false positives and false negatives in a classification task. It is calculated as the harmonic mean of precision and recall.

$$F1 - Score = 2\,x\,\frac{Precision\,x\,Recall}{Precision + Recall} \qquad (5.3)$$

ACCURACY:

The ratio of correctly predicted cases to the total instances is used to calculate accuracy, which is a measure of overall correctness in a classification task.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \qquad (5.4)$$

Model	Precision	Recall	F1-score	Accuracy (%)
CNN	1.00	0.99	0.99	99.5
LSTM	0.98	0.97	0.97	97.5
GRU	0.98	0.97	0.97	97.5
BiLSTM	0.98	0.99	0.98	98.5
RNN	0.98	0.98	0.98	98
DNN	0.26	0.50	0.34	53

Table 5.1. Comparison of different DL model

CHAPTER 6

RESULTS AND DISCUSSION

To summarise, the audio dataset from the Torgo database was analyzed, and features were extracted using Mel-frequency cepstral coefficients (MFCC) in order to create prediction models that could identify dysarthria. Based on these retrieved data, six alternative deep learning models were used to predict dysarthria. The study concentrated on determining the efficacy of using MFCC features for dysarthria detection. The models were trained, validated, and assessed during this process to determine how well they classified speech samples into dysarthric and non-dysarthric categories.

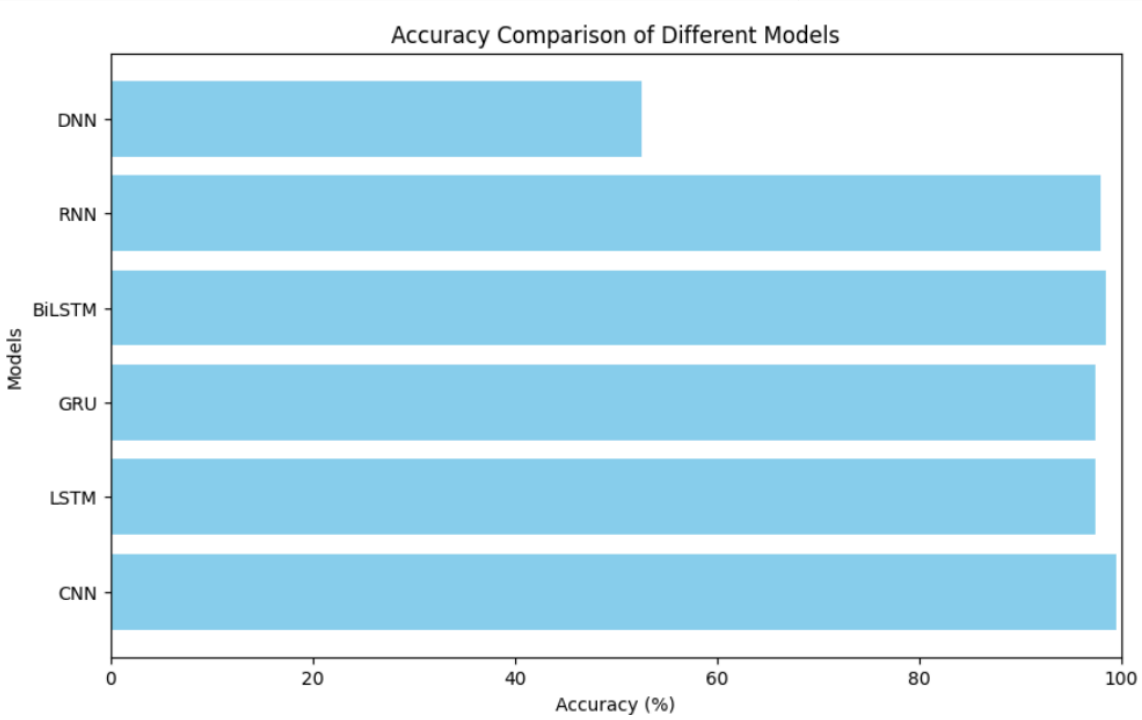

Fig.6.1 Accuracy comparison of DL models

CNN and LSTM are two examples of the classification models used for the analysis of the audio dataset from the Torgo database. Using mfcc methods, the audios were initially preprocessed. It was discovered that the CNN model's maximum accuracy of 99.5% was attained when it came to dividing the audio files into two categories: dysarthria and non-dysarthria.

CHAPTER 7

CONCLUSION

The dysarthria identification study demonstrated how well deep learning models and sophisticated signal processing techniques work together to reliably identify dysarthria in audio recordings. The system shown strong performance in diagnosing dysarthria through an extensive investigation of multiple model architectures, such as CNN, LSTM, GRU, Bidirectional LSTM, Simple RNN, and DNN, in conjunction with feature extraction techniques including MFCCs, zero-crossing rate, and spectral Rolloff. The system's ability to accurately detect dysarthria has important practical implications as it provides medical professionals with a dependable means of detecting speech abnormalities. Its speedy analysis of audio recordings has the potential to improve patient outcomes by streamlining the diagnosis procedure. The project pinpointed a number of directions for future investigation, such as more study into feature extraction methods and model architecture optimization.

It was also emphasized that validation on bigger and more varied datasets will improve the system's functionality and increase its application. Metrics like accuracy, recall, and ROC AUC score showed how accurate the models were. A thorough evaluation of the model's performance was provided by the addition of visualizations such as confusion matrices, ROC curves, and loss curves to the evaluation metrics. The dysarthria detection system described in this work is a major advancement in the use of technology to enhance dysarthria diagnosis and treatment. With the potential to improve patient treatment and further our understanding of speech-related illnesses, the system provides a valuable tool for researchers and clinicians by fusing deep learning models with sophisticated signal processing techniques.

CHAPTER 8

APPENDICES

8.1 APPENDIX – 1 CODING

```
from IPython.display import Audio
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import re
import
librosa
import librosa.display
from sklearn.preprocessing import minmax_scale
import IPython.display as ipd

from tqdm import tqdm
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split
from sklearn.metrics import classification_report, confusion_matrix, roc_auc_score, roc_curve, recall_score
import tensorflow as tf
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import InputLayer, Conv2D, MaxPooling2D, Dense, Flatten, LSTM, GRU,
Bidirectional, SimpleRNN
from tensorflow.keras.callbacks import EarlyStopping, ModelCheckpoint
from sklearn.metrics import accuracy_score,log_loss,
confusion_matrix,ConfusionMatrixDisplay import warnings
warnings.filterwarnings("ignore")

pip install opendatasets

import opendatasets as od

od.download("https://www.kaggle.com/datasets/iamhungundji/dysarthria-detection/data"

) dirname='/content/dysarthria-detection'

import os
d = pd.read_csv("/content/dysarthria-detection/torgo_data/data.csv")
d['filename'] = d['filename'].apply(lambda x: os.path.join(dirname,x))
d
```

```
sns.displot( d.gender)
plt.title('Gender')
plt.ylabel('Count', size=12)
plt.xlabel('Gender', size=12)
plt.show()

sns.displot( d.is_dysarthria)
plt.title('Dysarthria')
plt.ylabel('Count', size=12)
plt.xlabel('Gender', size=12)


plt.show()

def create_waveplot(data, sr, i):
    plt.figure(figsize=(10, 3))
    plt.title('Waveplot for audio with {}'.format(i), size=15)
    librosa.display.waveshow(data, sr=sr)
    plt.show()

def create_mel_Spectogram(data, sr, i):
    plt.figure(figsize=(10, 3))
    plt.title('Mel Spectrogram of audio {} emotion'.format(i), size=15)
    mel_spec = librosa.feature.melspectrogram(y=data, sr=sr)
    librosa.display.specshow(mel_spec, sr=sr, x_axis='time')
    plt.show()
def create_Spectogram(data, sr, i):

    plt.figure(figsize=(10, 3))
    plt.title('Spectrogram of audio {} '.format(i), size=15)
    X = librosa.stft(data)

    Xdb = librosa.amplitude_to_db(abs(X))
    librosa.display.specshow(Xdb,sr = sr, x_axis="time",y_axis = "log")
    plt.show()


def create_Zero_crossing_rate(data, sr, i):

    zero_crossings = librosa.zero_crossings(data)
    print("Sum of zero crossing ", zero_crossings.sum())
    plt.figure(figsize=(10, 3))
    plt.title('Zero Crossing Rate of audio {} '.format(i), size=15)
    zcrs = librosa.feature.zero_crossing_rate(data)
    plt.plot(zcrs[0])
    plt.show()
```

```
def create_mfcc(data, sr, i):

    plt.figure(figsize=(10, 3))
    plt.title('MFCC of audio {} '.format(i), size=15)
    mfccs = librosa.feature.mfcc(y=data, sr=sr)
    librosa.display.specshow(mfccs, sr=sr,
    x_axis='time') plt.show()
def create_Spectral_rolloff(data, sr, i):
    plt.figure(figsize=(10, 3))
    plt.title('Spectral Rolloff of audio {} '.format(i), size=15)
    spectral_rolloff = librosa.feature.spectral_rolloff(y=data, sr=sr)[0]
    frames = range(len(spectral_rolloff))
    t = librosa.frames_to_time(frames)
    librosa.display.waveshow(data, sr=sr, alpha=0.4)
    plt.plot(t, spectral_rolloff, color='r')
    plt.show()

    dysarthricMALE = d[(d['gender']=='male') & (d['is_dysarthria']=='dysarthria') ]
    DM = dysarthricMALE.sample()
    x= DM.iloc[0].filename info= "Male with Dysarthia"
    data, sampling_rate = librosa.load(x)
    create_waveplot(data, sampling_rate, info)
    create_Spectogram(data, sampling_rate, info)
    create_Zero_crossing_rate(data, sampling_rate, info)
    create_Spectral_rolloff(data, sampling_rate, info)
    create_mfcc(data, sampling_rate, info)
    create_mel_Spectogram(data, sampling_rate, info)
    Audio(x)


    dysarthricFemale = d[(d['gender']=='female') & (d['is_dysarthria']=='dysarthria')]
    DF = dysarthricFemale.sample()
    x= DF.iloc[0].filename
    info= "Female with Dysarthia"
    data, sampling_rate = librosa.load(x)
     create_waveplot(data, sampling_rate, info)
    create_Spectogram(data, sampling_rate, info)
    create_Zero_crossing_rate(data, sampling_rate, info)
     create_Spectral_rolloff(data, sampling_rate, info)
    create_mfcc(data, sampling_rate, info)
    create_mel_Spectogram(data, sampling_rate, info)
     Audio(x)
```

```
non_dysarthricMale = d[(d['gender']=='male') & (d['is_dysarthria']=='non_dysarthria')]
NDM = non_dysarthricMale.sample()
v= NDM.iloc[0].filename
info= "Male not with Dysarthia"
data, sampling_rate =
librosa.load(v)
create_waveplot(data, sampling_rate, info)
create_Spectogram(data, sampling_rate, info)
create_Zero_crossing_rate(data, sampling_rate, info)
create_Spectral_rolloff(data, sampling_rate, info)
create_mfcc(data, sampling_rate, info)
create_mel_Spectogram(data, sampling_rate, info)
Audio(v)


non_dysarthricFemale = d[(d['gender']=='female') &
(d['is_dysarthria']=='non_dysarthria')] NDF= non_dysarthricFemale.sample()
w= NDF.iloc[0].filename
info= "Female not with Dysarthia"
data, sampling_rate = librosa.load(w)
create_waveplot(data, sampling_rate, info)
create_Spectogram(data, sampling_rate, info)
create_Zero_crossing_rate(data, sampling_rate, info)
create_Spectral_rolloff(data, sampling_rate, info)
create_mfcc(data, sampling_rate, info)
create_mel_Spectogram(data, sampling_rate, info)
Audio(w)
```

FEATURE EXTRACTION USING MFCC

```
def mfcc_features_extract(x):
    features_mfcc = []
    try:
        x , sr = librosa.load(x)
        features_mfcc = librosa.feature.mfcc(y=x, sr=sr, n_mfcc=128)
        features_mfcc = np.mean(features_mfcc.T,axis=0)
    except:
        print('Error reading audio')
    return features_mfcc


N= d['filename'].apply(lambda a: mfcc_features_extract(a))

N= N.tolist()
N=pd.DataFrame(N)

N.head(10)

N['Dysarthia'] = d['is_dysarthria']
```

```
N.loc[N['Dysarthia']=='non_dysarthria','Dysarthia'] = 0
N.loc[N['Dysarthia']=='dysarthria','Dysarthia'] = 1

N["Dysarthia"].value_counts()

N.isna().sum().sum()

N = N.dropna()

N["Dysarthia"].value_counts()

X = N.drop(['Dysarthia'],axis=1)
y = N['Dysarthia'].astype(float)

print("Shape of X and y: ", X.shape, y.shape)
X_train, X_val, y_train, y_val = train_test_split(X, y, test_size=0.1)
sc = StandardScaler()
X_train_scaled =
sc.fit_transform(X_train) X_val_scaled =
sc.transform(X_val)

import tensorflow as tf
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import InputLayer, Conv2D, MaxPooling2D, Dense, Flatten
from tensorflow.keras.callbacks import EarlyStopping, ModelCheckpoint

X_train_scaled =
X_train_scaled.reshape(-1,16,8,1) X_val_scaled =
X_val_scaled.reshape(-1,16,8,1)
print("Shape of X_train and X_test: ", X_train_scaled.shape, X_val_scaled.shape)
```

CNN

```
model_1 = Sequential([
   InputLayer(input_shape=(16, 8, 1)),
   Conv2D(filters=16, kernel_size=(3, 3), activation='relu', padding = "same"),
   MaxPooling2D(2, 2),
   Conv2D(filters=64, kernel_size=(3, 3), activation='relu', padding = "same"),
   MaxPooling2D(2, 2),
   Flatten(),
   Dense(32, activation='relu'),
   Dense(1, activation='sigmoid')
])

model_1.summary()

model_1.compile(optimizer='adam', loss='binary_crossentropy', metrics=['accuracy'])
history_1 = model_1.fit(X_train_scaled, y_train, epochs=100,
validation_data=(X_val_scaled,y_val)) plt.figure(figsize=(15,8))
plt.plot(history_1.history['loss']
```

```
plt.plot(history_1.history['val_loss'])
plt.xlabel('EPOCH')
plt.ylabel('LOSS')
plt.title('MODEL
LOSS')
plt.legend(['train', 'val'])
plt.show()

plt.figure(figsize=(15,8))
plt.plot(history_1.history['accuracy'])
plt.plot(history_1.history['val_accuracy'])
plt.xlabel('EPOCH')
plt.ylabel('ACCURACY')
plt.title('MODEL ACCURACY')
plt.legend(['train', 'val'])
plt.show()
ypred_1 =
model_1.predict(X_val_scaled)
roc_auc_score(y_val,ypred_1)
plt.figure(figsize=(15,8))
f, t, _ = roc_curve(y_val, ypred_1)
AUC_score = roc_auc_score(y_val, ypred_1)
plt.plot(f,t,label=" Model, AUC is "+str(AUC_score))
plt.plot([0, 1], [0, 1], color="red")
plt.legend(loc=4
) plt.show()

ypred_1[ypred_1>=0.5] = 1
ypred_1[ypred_1<0.5] = 0

print("Recall Score

",recall_score(y_val,ypred_1))

print(classification_report(y_val,ypred_1))

print("Testing Accuracy = ",accuracy_score(y_val,ypred_1)*100)
sns.heatmap(confusion_matrix(y_val, ypred_1),cmap='viridis',annot=True,fmt='.3g',
        xticklabels=['Non Dysarthria','Dysarthria'],yticklabels=['Non Dysarthria','Dysarthria'])
plt.xlabel('Predicted Class')
plt.ylabel('Actual Class')
plt.show()
```

LSTM

```
model_2 = Sequential([
   LSTM(units=64,
   input_shape=(16,8)), Dense(64,
   activation='relu'),
   Dense(1, activation='sigmoid')
])
```

```
model_2.compile(optimizer='adam', loss='binary_crossentropy', metrics=['accuracy'])

model_2.summary()

history_2 = model_2.fit(X_train_scaled, y_train, epochs=100,

validation_data=(X_val_scaled,y_val)) ypred_2 = model_2.predict(X_val_scaled)

roc_auc_score(y_val,ypred_2)

ypred_2[ypred_2>=0.5] = 1


ypred_2[ypred_2<0.5] = 0

print("Recall Score

",recall_score(y_val,ypred_2))

print(classification_report(y_val,ypred_2))

print("Testing Accuracy = ",accuracy_score(y_val,ypred_2)*100)
sns.heatmap(confusion_matrix(y_val, ypred_2),cmap='viridis',annot=True,fmt='.3g',
        xticklabels=['Non Dysarthria','Dysarthria'],yticklabels=['Non Dysarthria','Dysarthria'])
plt.xlabel('Predicted Class')
plt.ylabel('Actual Class')
plt.show()
```

GRU

```
model_3 = Sequential([
   GRU(64, input_shape=(16, 8)),
   Dense(64, activation='relu'),
   Dense(1, activation='sigmoid')
])

model_3.compile(optimizer='adam', loss='binary_crossentropy', metrics=['accuracy'])

model_3.summary()

history_3 = model_3.fit(X_train_scaled, y_train, epochs=100,

validation_data=(X_val_scaled,y_val)) ypred_3 = model_3.predict(X_val_scaled)

roc_auc_score(y_val,ypred_3)

ypred_3[ypred_3>=0.5] = 1
ypred_3[ypred_3<0.5] = 0
```

```
print("Recall Score",recall_score(y_val,ypred_3))

print(classification_report(y_val,ypred_3))

print("Testing Accuracy = ",accuracy_score(y_val,ypred_3)*100)
sns.heatmap(confusion_matrix(y_val, ypred_3),cmap='viridis',annot=True,fmt='.3g',
        xticklabels=['Non Dysarthria','Dysarthria'],yticklabels=['Non Dysarthria','Dysarthria'])
plt.xlabel('Predicted Class')
plt.ylabel('Actual Class')
plt.show()
```

BiLSTM

```
model_4 = Sequential([
   Bidirectional(LSTM(64), input_shape=(16, 8)),
   Dense(64, activation='relu'),
   Dense(1, activation='sigmoid')
])

model_4.compile(optimizer='adam', loss='binary_crossentropy', metrics=['accuracy'])

model_4.summary()

history_4 = model_4.fit(X_train_scaled, y_train, epochs=100,

validation_data=(X_val_scaled,y_val)) ypred_4 = model_4.predict(X_val_scaled)


roc_auc_score(y_val,ypred_4)

ypred_4[ypred_4>=0.5] = 1
ypred_4[ypred_4<0.5] = 0

print("Recall Score

",recall_score(y_val,ypred_4))

print(classification_report(y_val,ypred_4))

print("Testing Accuracy = ",accuracy_score(y_val,ypred_4)*100)
sns.heatmap(confusion_matrix(y_val, ypred_4),cmap='viridis',annot=True,fmt='.3g',
        xticklabels=['Non Dysarthria','Dysarthria'],yticklabels=['Non Dysarthria','Dysarthria'])
plt.xlabel('Predicted Class')
plt.ylabel('Actual Class')
plt.show()
```

RNN

```
model_5 = Sequential([
   SimpleRNN(64, input_shape=(16, 8)),
   Dense(64, activation='relu'),
   Dense(1, activation='sigmoid')
])

model_5.compile(optimizer='adam', loss='binary_crossentropy', metrics=['accuracy'])

model_5.summary()

history_5 = model_5.fit(X_train_scaled, y_train, epochs=100,

validation_data=(X_val_scaled,y_val)) ypred_5 = model_5.predict(X_val_scaled)

roc_auc_score(y_val,ypred_5)

ypred_5[ypred_5>=0.5] = 1
ypred_5[ypred_5<0.5] = 0
print("Recall Score

",recall_score(y_val,ypred_5))

print(classification_report(y_val,ypred_5))

print("Testing Accuracy = ",accuracy_score(y_val,ypred_5)*100)
sns.heatmap(confusion_matrix(y_val, ypred_5),cmap='viridis',annot=True,fmt='.3g',
        xticklabels=['Non Dysarthria','Dysarthria'],yticklabels=['Non Dysarthria','Dysarthria'])
plt.xlabel('Predicted Class')
plt.ylabel('Actual Class')
plt.show()
```

DNN

```
model_6 = Sequential([
   Flatten(input_shape=(16, 8)),
   Dense(64, activation='relu'),
   Dense(64, activation='relu'),
   Dense(1, activation='softmax')
])
model_6.compile(optimizer='adam', loss='binary_crossentropy', metrics=['accuracy'])
model_6.summary()

history_6 = model_6.fit(X_train_scaled, y_train, epochs=100, validation_data=(X_val_scaled,y_val))
ypred_6 = model_6.predict(X_val_scaled)
roc_auc_score(y_val,ypred_6)
```

```
ypred_6[ypred_6>=0.5] = 1
ypred_6[ypred_6<0.5] = 0
print("Recall Score

",recall_score(y_val,ypred_6))

print(classification_report(y_val,ypred_6))

print("Testing Accuracy = ",accuracy_score(y_val,ypred_6)*100)
sns.heatmap(confusion_matrix(y_val, ypred_6),cmap='viridis',annot=True,fmt='.3g',
          xticklabels=['Non Dysarthria','Dysarthria'],yticklabels=['Non Dysarthria','Dysarthria'])
plt.xlabel('Predicted Class')
plt.ylabel('Actual Class')
plt.show( )

models = ['CNN', 'LSTM', 'GRU', 'BiLSTM', 'RNN', 'DNN']

accuracies = [accuracy_score(y_val,ypred_1)*100, accuracy_score(y_val,ypred_2)*100,
accuracy_score(y_val,ypred_3)*100,
           accuracy_score(y_val,ypred_4)*100, accuracy_score(y_val,ypred_5)*100,
accuracy_score(y_val,ypred_6)*100]

plt.figure(figsize=(10,6))
plt.bar(models, accuracies,
color='seagreen') plt.xlabel('Models')
plt.ylabel('Accuracy (%)')
plt.title('Accuracy Comparison of Different Models')
plt.ylim([0, 100])

for i in range(len(models)):
   plt.text(i, accuracies[i], f"{accuracies[i]:.2f}%", ha='center', va='bottom')

plt.show()
```

8.1 APPENDIX – 2 SCREENSHOTS

	precision	recall	f1-score	support
0.0	1.00	0.99	0.99	95
1.0	0.99	1.00	1.00	105
accuracy			0.99	200
macro avg	1.00	0.99	0.99	200
weighted avg	1.00	0.99	0.99	200

Fig A2.1: Classification report of CNN

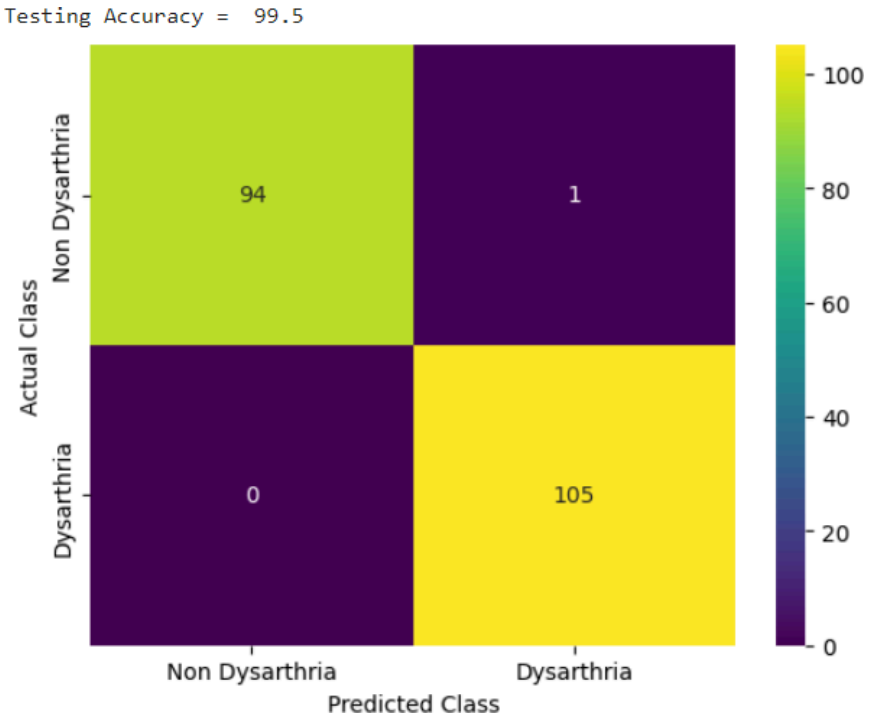

Fig A2.2: Confusion matrix of CNN

	precision	recall	f1-score	support
0.0	0.99	0.96	0.97	95
1.0	0.96	0.99	0.98	105
accuracy			0.97	200
macro avg	0.98	0.97	0.97	200
weighted avg	0.98	0.97	0.97	200

Fig A2.3: Classification report of LSTM

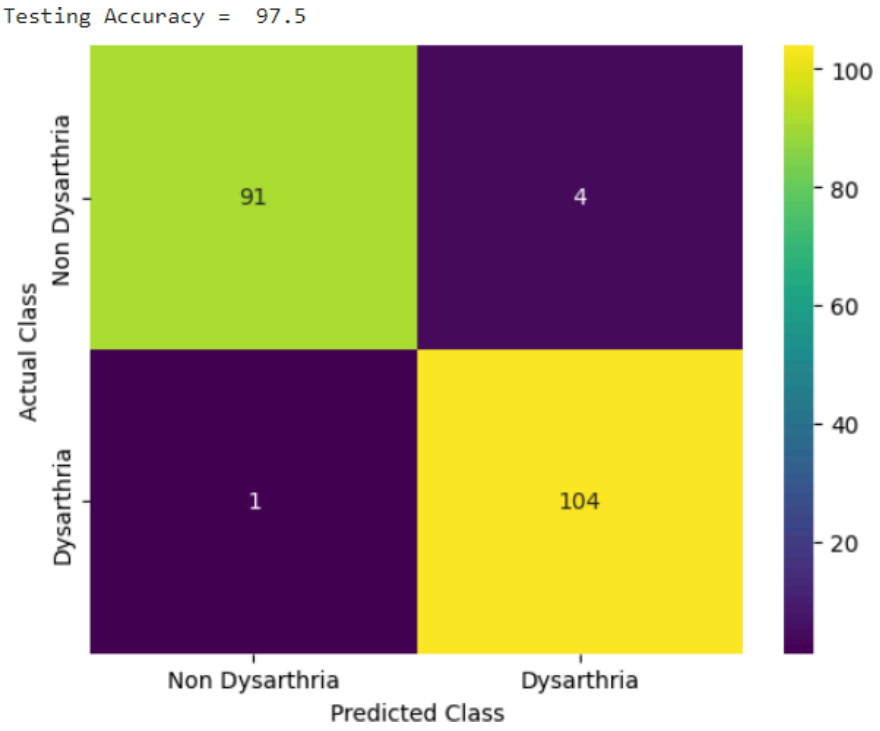

Fig A2.4: Confusion matrix of LSTM

	precision	recall	f1-score	support
0.0	0.98	0.97	0.97	95
1.0	0.97	0.98	0.98	105
accuracy			0.97	200
macro avg	0.98	0.97	0.97	200
weighted avg	0.98	0.97	0.97	200

Fig A2.5: Classification report of GRU

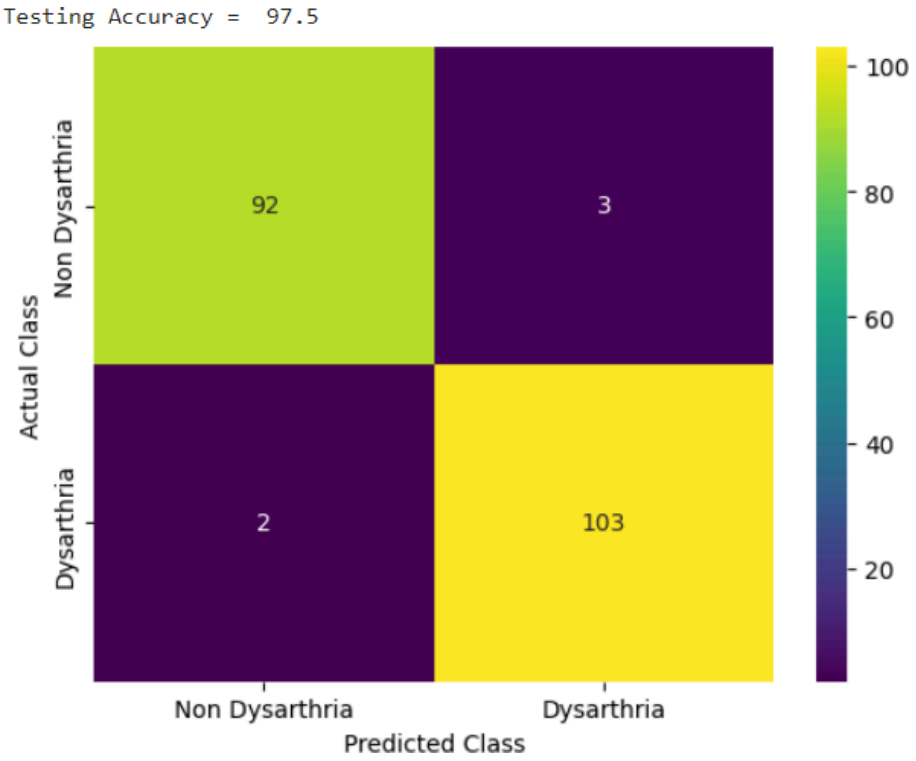

Fig A2.6: Confusion matrix of GRU

	precision	recall	f1-score	support
0.0	0.98	0.99	0.98	95
1.0	0.99	0.98	0.99	105
accuracy			0.98	200
macro avg	0.98	0.99	0.98	200
weighted avg	0.99	0.98	0.99	200

Fig A2.7: Classification report of BiLSTM

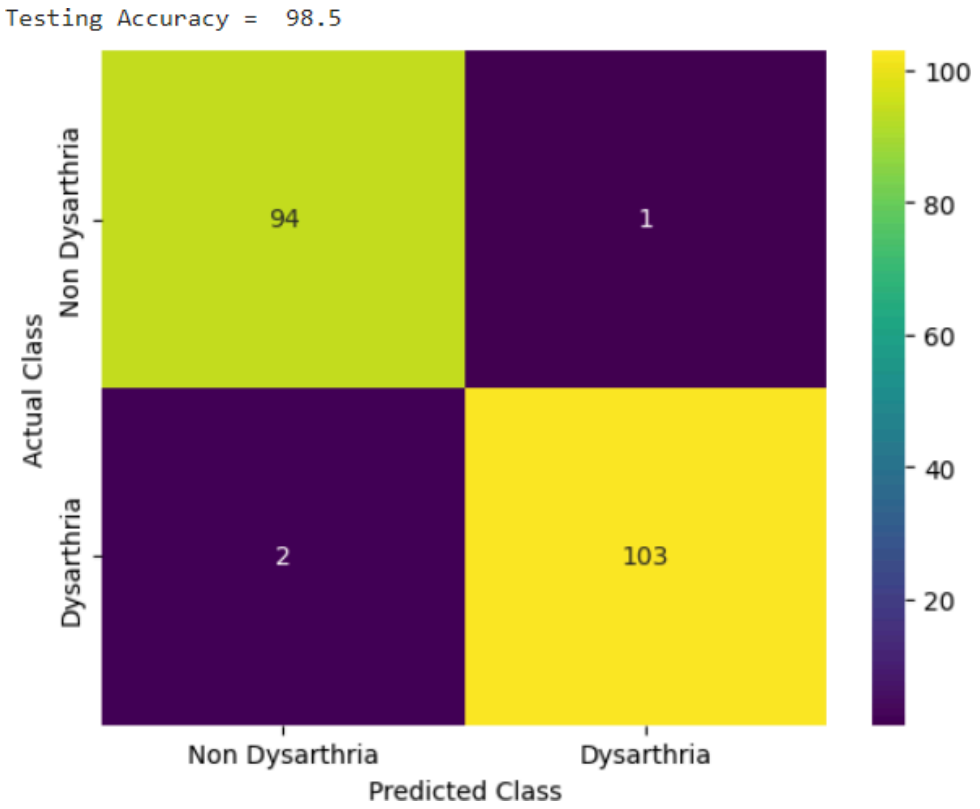

Fig A2.8: Confusion matrix of BiLSTM

	precision	recall	f1-score	support
0.0	0.98	0.98	0.98	95
1.0	0.98	0.98	0.98	105
accuracy			0.98	200
macro avg	0.98	0.98	0.98	200
weighted avg	0.98	0.98	0.98	200

Fig A2.9: Classification report of RNN

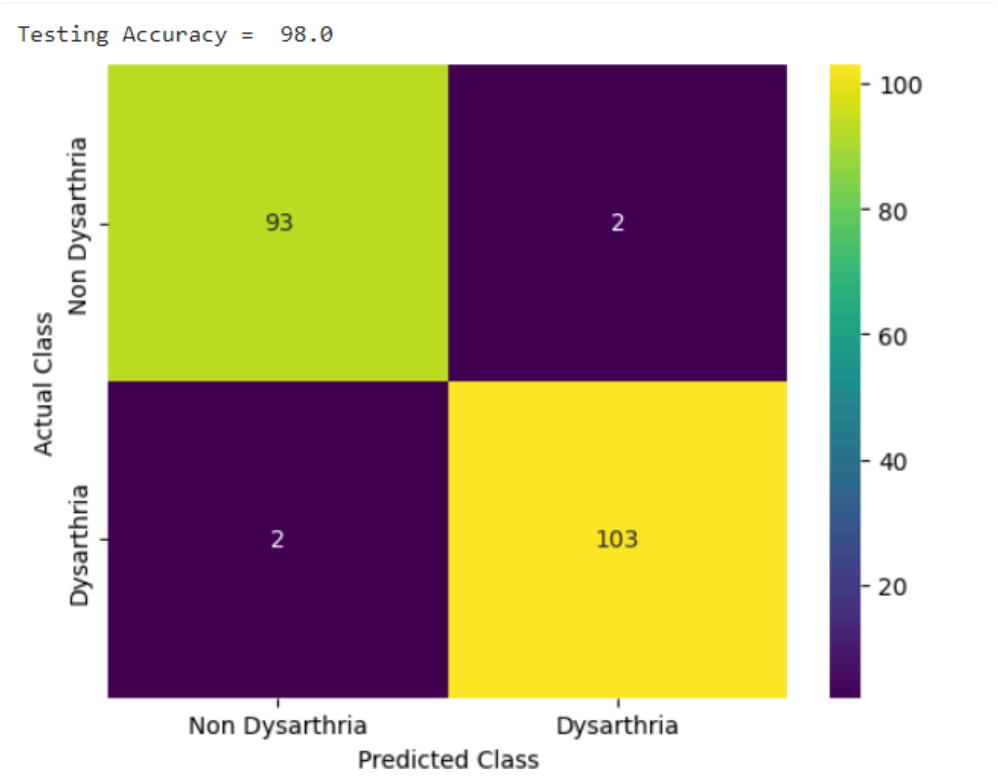

Fig A2.10: Confusion matrix of RNN

	precision	recall	f1-score	support
0.0	0.00	0.00	0.00	95
1.0	0.53	1.00	0.69	105
accuracy			0.53	200
macro avg	0.26	0.50	0.34	200
weighted avg	0.28	0.53	0.36	200

Fig A2.11: Classification report of DNN

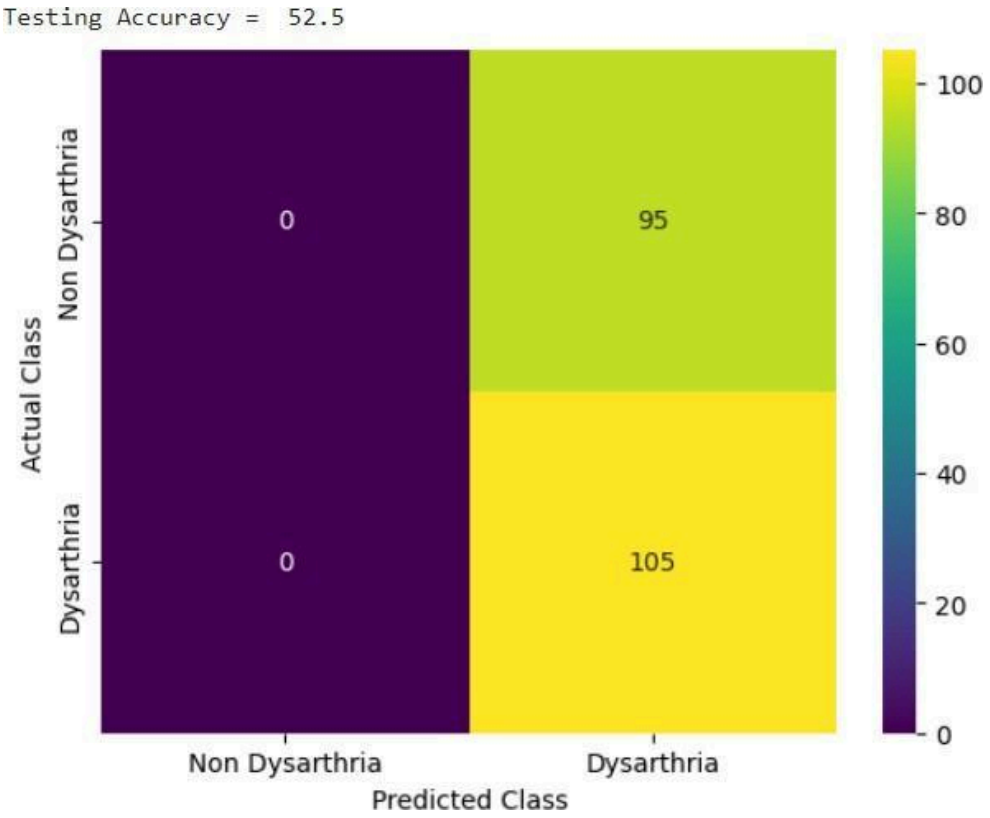

Fig A2.12: Confusion matrix of DNN

REFERENCES

1. S. Padmanaban, “Comparative Analysis of Deep Learning Models for Dysarthric Speech Detection Comparative Analysis of Deep Learning Models for Dysarthric Speech Detection,” 2023. Accessed: Apr. 23, 2024. [Online]. Available: https://assets.researchsquare.com/files/rs-1916239/v1_covered_136b2fdf-0bcf-4cf0-8f99-99c575c53b56.pdf?c=1699888127
2. M. Mahendran, R. Visalakshi, and S. Balaji, “Dysarthria detection using convolution neural network,” *Measurement: Sensors*, vol. 30, p. 100913, Dec. 2023, doi: https://doi.org/10.1016/j.measen.2023.100913.
3. D.-H. Shih, C.-H. Liao, T.-W. Wu, X.-Y. Xu, and M.-H. Shih, “Dysarthria Speech Detection Using Convolutional Neural Networks with Gated Recurrent Unit,” *Healthcare*, vol. 10, no. 10, p. 1956, Oct. 2022, doi: https://doi.org/10.3390/healthcare10101956.
4. A. A. Joshy and R. Rajan, “Automated Dysarthria Severity Classification: A Study on Acoustic Features and Deep Learning Techniques,” *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 30, pp. 1147–1157, 2022, doi: https://doi.org/10.1109/tnsre.2022.3169814.
5. H. Albaqshi and A. Sagheer, “Dysarthric Speech Recognition using Convolutional Recurrent Neural Networks,” *International Journal of Intelligent Engineering and Systems*, vol. 13, no. 6, pp. 384–392, Dec. 2020, doi: https://doi.org/10.22266/ijies2020.1231.34
6. P. Janbakhshi, I. Kodrasi, and H. Bourlard, “AUTOMATIC DYSARTHRIC SPEECH DETECTION EXPLOITING PAIRWISE DISTANCE-BASED CONVOLUTIONAL NEURAL NETWORKS.” Accessed: Apr. 28, 2024. [Online]. Available: https://publications.idiap.ch/attachments/papers/2021/Janbakhshi_ICASSP_2021.pdf

7. J. Millet and N. Zeghidour, “LEARNING TO DETECT DYSARTHRIA FROM RAW SPEECH.” Accessed: Apr. 28, 2024. [Online]. Available: https://scontent.fmaa12-2.fna.fbcdn.net/v/t39.2365-6/154874650_13089683559141 9_2037808512467710111_n.pdf?_nc_cat=110&ccb=1-7&_nc_sid=3c67a6&_nc_o hc=xihMx8VzARUAb6RFWoT&_nc_ht=scontent.fmaa12-2.fna&oh=00_AfDt0q9 qRPCFhzhvpR2KSPbG1Xe6ZOXXUUQt-Phl8xCwCQ&oe=6633E8D7
8. T. Ijitona, J. Soraghan, A. Lowit, G. Di-Caterina, and H. Yue, “Automatic Detection of Speech Disorder in Dysarthria using Extended Speech Feature Extraction and Neural Networks Classification.” Accessed: Apr. 28, 2024. [Online]. Available: https://pure.strath.ac.uk/ws/portalfiles/portal/70445057/Ijitona_etal_ICISP_2017_A utomatic_detection_of_speech_disorder_in_dysarthria.pdf
9. S. Khaled, A. Hassan, A. Salah, D. Mohamed, M. Mustafa, and E. Mohamed, “Re-Talk: Automated Speech Assistance for People with Dysarthria,” vol. 5, 2023, Available: https://fcihib.journals.ekb.eg/article_284717_685c28e1fe9e2af6712e53ba9d536091 .pdf
10. A. A. Joshy and R. Rajan, “Automated Dysarthria Severity Classification: A Study on Acoustic Features and Deep Learning Techniques,” IEEE Transactions on Neural Systems and Rehabilitation Engineering, vol. 30, pp. 1147–1157, 2022, doi: https://doi.org/10.1109/tnsre.2022.3169814.