DIAGNOSIS OF PARKINSON’S DISEASE USING MACHINE LEARNING AND DEEP LEARNING

A PROJECT REPORT

Submitted by

GANESH A	**(211419205049)**
HAMMADH AHMED A	**(211419205060)**
GANESH S	**(211419205051)**

in partial fulfillment for the award of the degree

of

BACHELOR OF TECHNOLOGY

IN

INFORMATION TECHNOLOGY

PANIMALAR ENGINEERING COLLEGE, POONAMALLEE

ANNA UNIVERSITY : CHENNAI 600 025

APRIL 2023

ANNA UNIVERSITY: CHENNAI 600 025

BONAFIDE CERTIFICATE

Certified that this project report "**DIAGNOSIS OF PARKINSON'S DISEASE USING MACHINE LEARNING AND DEEP LEARNING"** is the bonafide work of **"GANESH A(211419205049),HAMMADH AHMED A (211419205060) ,GANESH S(211419205051)"** who carried out the project under my supervision.

SIGNATURE	**SIGNATURE**
Dr.M.HELDA MERCY M.E., Ph.D.,	**Mrs.S.SHARANYAA M.Tech.,(Ph.D.)**
HEAD OF THE DEPARTMENT	**SUPERVISOR**
	ASSISTANT PROFESSOR
Department of Information Technology	Department of Information Technology
Panimalar Engineering College	Panimalar Engineering College
Poonamallee, Chennai - 600 123	Poonamallee, Chennai - 600 123

Submitted for the project and viva-voce examination held on _______________

SIGNATURE	**SIGNATURE**
INTERNAL EXAMINER	**EXTERNAL EXAMINER**

DECLARATION

I hereby declare that the project report entitled **"DIAGNOSIS OF PARKINSON'S DISEASE USING MACHINE LEARNING AND DEEP LEARNING"** which is being submitted in partial fulfilment of therequirement of the course leading to the award of the 'Bachelor of Technology inInformation Technology' in **Panimalar Engineering College, Autonomous institution Affiliated to Anna university- Chennai** is the result of the project carried out by me under the guidance of **Mrs. S. SHARANYAA M.Tech.,(Ph.D.) Assistant Professor in the Department of Information Technology**. I further declared that I or any other person has not previously submitted this project report to any other institution/university for any other degree/ diploma orany other person.

GANESH A

HAMMADH AHMED A

GANESH S

Date**:**

Place**:** Chennai

It is certified that this project has been prepared and submitted under my guidance.

Date: **Mrs. S. SHARANYAA M.Tech.,(Ph.D.)**

Place: Chennai Assistant Professor / IT

ACKNOWLEDGEMENT

A project of this magnitude and nature requires kind co-operation and support from many, for successful completion . We wish to express our sincere thanks to all those who were involved in the completion of this project.

Our sincere thanks to **Our Honorable Secretary and Correspondent, Dr. P. CHINNADURAI, M.A., Ph.D.,** for his sincere endeavor in educating usin his premier institution.

We would like to express our deep gratitude **to Our Dynamic Directors , Mrs. C. VIJAYA RAJESHWARI and Dr. C. SAKTHI KUMAR, M.E.,M.B.A.,Ph.D., and Dr.SARANYA SREE SAKTHIKUMAR .,B.E.,M.B.A.,Ph.D.,** for providing us with the necessary facilities for completion of this project.

We also express our appreciation and gratefulness to **Our Principal Dr. K. MANI, M.E., Ph.D.,** who helped us in the completion of the project.We wish to convey our thanks and gratitude to our head of the department, **Dr. M. HELDA MERCY, M.E., Ph.D.,** Department of Information Technology, for her support and by providing us ample time to complete our project.

We express our indebtedness and gratitude to our Project co-ordinator **Mr. M. DILLI BABU**, **M.E.,(Ph.D.,)** Associate Professor, Department of Information Technology for his guidance throughout the course of our project. We also express sincere thanks to our **Mrs .S.SHARANYAA, M.Tech.,(Ph.D.)** Assistant Professor for providing the support to carry out the project successfully. Last, wethank our parents and friends for providing their extensive moral support and encouragement during the course of the project.

ABSTRACT

Parkinson's disease (PD) is a debilitating neurodegenerative disorder that affects millions of people worldwide. The early detection of Parkinson’s disease is crucial to providing patients with appropriate treatment and improving their quality of life. In recent years, machine learning and deep learning techniques have shown promise in detecting Parkinson’s disease from various types of medical data, such as MRI scan brain images and audio samples. In this study, we developed a novel model for the early detection of Parkinson’s disease using both MRI scan brain images and audio samples. We trained a Convolutional Neural Network (CNN) using MRI scan brain images and XG Boost, Cat Boost, and Random Forest algorithms using audio samples. We evaluated the model's performance using various metrics, such as accuracy, precision, recall, and F1-score. The results showed that our model achieved high accuracy of 98% in detecting Parkinson’s disease from both MRI scan brain images and audio samples. This model could potentially serve as a valuable screening tool for the early detection of Parkinson’s disease, allowing for timely treatment and better patient outcomes. The proposed model's ability to use both imaging and audio data could enable more efficient and accurate diagnosis of Parkinson’s disease.

LIST OF FIGURES

LIST OF ABBREVATIONS

CNN	Convolutional Neural Network
XG Boost	Extreme Gradient Boosting
UML	Unified Modeling Language
RAM	Random Access Memory
UI	User Interface
ER	Entity Relationship
KNN	K – Nearest Neighbor
MRI	Magnetic Resonance Imaging
Cat Boost	Categorical Boosting

TABLE OF CONTENTS

INTRODUCTION

Parkinson's disease (PD) is a chronic and progressive neurodegenerative disorder that affects the central nervous system. The disease primarily affects the motor system, resulting in tremors, stiffness, and difficulty with movement and coordination. As the disease progresses, patients may also experience non-motor symptoms, such as cognitive impairment and psychiatric disorders. Early detection of PD is critical for timely and effective treatment, which can significantly improve patients' quality of life .Machine learning and deep learning techniques have shown great promise in the early detection and diagnosis of PD.

These techniques utilize various types of medical data, such as MRI scan brain images and audio samples, to develop models that can accurately detect PD. In this context, Convolutional Neural Networks (CNN) are a popular deep learning technique that has shown to be effective in image classification tasks such as MRI scan brain images. Parkinson's disease is a debilitating neurological disorder that affects millions of people worldwide. This progressive disorder is characterized by a range of symptoms, including tremors, rigidity, and difficulty with movement and coordination. Early detection of Parkinson's disease is crucial, as early diagnosis can allow for better management of symptoms and improved quality of life for patients.

In recent years, machine learning and deep learning techniques have shown promise in accurately detecting Parkinson's disease using various types of data, including MRI scan brain images and audio samples .MRI scan brain images are one of the most commonly used types of data for detecting Parkinson's disease using machine learning and deep learning techniques. MRI scans provide detailed images of the brain, allowing for the identification of subtle changes associated with Parkinson's disease. In recent studies, convolutional neural networks (CNNs) have been used to train on MRI scan brain images to accurately identify features associated with Parkinson's disease.

A study by Rizzo et al. (2020) used a CNN to analyze MRI scan brain images and accurately detect Parkinson's disease with a high degree of accuracy .Audio samples are another type of data that can be used to detect Parkinson's disease using machine learning and deep learning techniques. Parkinson's disease can affect speech patterns, leading to changes in voice quality, volume, and rhythm. Researchers have used various algorithms, including XGBoost , Cat Boost, and Random Forest, to analyze audio samples and predict the presence of Parkinson's disease.

A study by Tsanas et al. (2010) used a combination of machine learning algorithms to analyze audio samples and accurately detect Parkinson's disease with a high degree of accuracy .The use of machine learning and deep learning techniques for detecting Parkinson's disease has important implications for early diagnosis and treatment. Early detection of Parkinson's disease can allow for earlier intervention and management of symptoms, potentially improving patient outcomes and quality of life. Furthermore, the ability to accurately detect Parkinson's disease using various types of data, including MRI scan brain images and audio samples, could lead to the development of new diagnostic tools and techniques.

However, there are also some challenges associated with using machine learning and deep learning techniques for detecting Parkinson's disease. One of the main challenges is the availability and quality of data. Access to high-quality MRI scan brain images and audio samples is crucial for accurate detection of Parkinson's disease. Furthermore, the accuracy of machine learning and deep learning models is heavily dependent on the quality and diversity of the training data. Therefore, efforts are needed to ensure that large and diverse datasets are available for training machine learning and deep learning models .In conclusion, machine learning and deep learning techniques have shown promise in accurately detecting Parkinson's disease using various types of data, including MRI scan brain images and audio samples. These techniques have important implications for early diagnosis and treatment of Parkinson's disease, potentially leading to improved patient outcomes and quality of life.

1.1 OBJECTIVE

The objective of using machine learning and deep learning techniques to detect Parkinson's disease using MRI scan brain images and audio samples is to accurately and early diagnose Parkinson's disease, leading to better management of symptoms, improved patient outcomes, and overall enhanced quality of life for patients with Parkinson's disease. This objective aims to leverage the power of machine learning and deep learning techniques to develop new diagnostic tools and techniques, leading to earlier intervention and management of Parkinson's disease. Additionally, this objective aims to address the challenges associated with data availability and quality, ensuring the accuracy and effectiveness of machine learning and deep learning models for detecting Parkinson's disease.

1.2 SCOPE

The scope of this project is to develop a machine learning algorithm that can aid in the early detection and diagnosis of Parkinson's disease using MRI images and audio recordings. Parkinson's disease is a progressive disorder that affects the nervous system and can cause tremors, difficulty with movement, and other symptoms that worsen over time. Early detection and diagnosis of Parkinson's disease is crucial, as it allows for early intervention and treatment, which can improve the patient's quality of life. The primary goal of this project is to develop a machine learning algorithm that can accurately detect Parkinson's disease from MRI images and audio recordings.

This algorithm can potentially help clinicians to identify patients who are at risk of developing Parkinson's disease or who have already developed the disease but are not showing symptoms. The algorithm can also be used to monitor the progression of the disease and the effectiveness of treatments .The scope of the project includes data collection, preprocessing, feature extraction, algorithm development, and evaluation. In the data collection phase, MRI images and audio recordings are obtained from both Parkinson's disease patients and healthy individuals.

The collected data are then preprocessed to eliminate noise and normalize them. Feature extraction techniques are used to identify meaningful characteristics from the data .In the algorithm development phase, different machine learning algorithms are used to analyze the MRI images and audio recordings. Convolutional neural networks (CNNs) are used for analyzing the MRI images, while XG Boost and CAT Boost algorithms are used for evaluating the audio data. The algorithms are trained on the collected data and evaluated using various metrics such as accuracy, sensitivity, specificity, and area under the curve (AUC).The scope of the project also includes exploring other machine learning algorithms, such as support vector machines (SVMs) and deep neural networks (DNNs), for the detection and diagnosis of Parkinson's disease.

The project also aims to integrate MRI and audio data to create a more comprehensive diagnostic tool. The scope of the project is not limited to the development of a machine learning algorithm but also includes the potential for further research and application of the algorithm. The project aims to develop a reliable and accurate algorithm for the early detection and diagnosis of Parkinson's disease. This algorithm can potentially be used in clinical practice to improve patient outcomes and quality of life .Additionally, the scope of the project extends beyond the medical field. The development of a machine learning algorithm for the early detection and diagnosis of Parkinson's disease has the potential to impact society as a whole. Early detection can reduce the burden on the healthcare system by identifying patients who require early intervention and treatment. It can also help patients to plan for their future and make informed decisions about their healthcare.

In conclusion, the scope of this project is to develop a machine learning algorithm that can aid in the early detection and diagnosis of Parkinson's disease using MRI images and audio recordings. The project involves data collection, preprocessing, feature extraction, algorithm development, and evaluation. The scope of the project includes exploring other machine learning algorithms, integrating MRI and audio data, and potential applications beyond the medical field. The development of a reliable and high accuracy algorithm for the early diagnosis of Parkinson's disease.

1.3 FEASIBILITY

- **Social feasibility:** Parkinson's disease is a serious neurological disorder affecting millions of people worldwide, and the development of a tool that can aid in its diagnosis can have a positive impact on society .The availability of such a tool can help doctors diagnose Parkinson's disease more accurately and at an earlier stage, leading to better treatment outcomes and improved quality of life for patients .However, it is important to ensure that the tool is developed in an ethical manner, with proper consideration given to patient privacy and confidentiality.
- **Technical feasibility:** The use of machine learning and deep learning algorithms for the diagnosis of Parkinson's disease is a well-established field, and there are numerous studies demonstrating the effectiveness of these techniques.The use of MRI brain images and audio samples for the diagnosis of Parkinson's disease is also a well-established approach .However, there may be technical challenges related to the availability and quality of data, as well as the need for high-performance computing resources to train and test the models.
- **Economic feasibility:** Developing a tool for the diagnosis of Parkinson's disease using machine learning and deep learning algorithms can be expensive, as it requires significant investment in computing resources, data collection, and personnel .However, the potential benefits of such a tool, in terms of improved diagnosis and treatment outcomes, can outweigh the costs .There may also be opportunities for revenue generation through the sale or licensing of the tool to hospitals and healthcare providers.

LITERATURE SURVEY

Title: A Wearable Sensor-based Approach for Parkinson's disease Diagnosis and Monitoring

Author: Wang et al.

Year: 2022

In this paper, the authors propose a wearable sensor-based approach for Parkinson's disease diagnosis and monitoring. The proposed system uses machine learning algorithms to analyse data from wearable sensors and predict the severity of Parkinson's disease symptoms .In this survey, Wang et al. proposed a wearable sensor-based approach for Parkinson's disease diagnosis and monitoring. The proposed system includes a set of wearable sensors, such as accelerometers, gyroscopes, and magnetometers, attached to the limbs of the patient. The sensors capture the patient's movement data and transmit it to a data processing unit for analysis .The authors used machine learning algorithms to analyse the movement data and predict the severity of Parkinson's disease symptoms. Specifically, they used support vector regression (SVR) and decision tree regression (DTR) to predict the severity of tremors, rigidity, and bradykinesia. The system achieved high accuracy in predicting Parkinson's disease symptoms, demonstrating the potential of wearable sensors and machine learning algorithms in Parkinson's disease diagnosis and monitoring .The proposed system has several advantages over traditional methods of Parkinson's disease diagnosis and monitoring. Firstly, it is non-invasive, which reduces patient discomfort and increases compliance. Secondly, it provides continuous and objective measurement of Parkinson's disease symptoms, which can be used to monitor disease progression and response to treatment. Thirdly, it has the potential to be used in remote monitoring and telemedicine, which can improve access to care and reduce healthcare costs .However, the proposed system has some limitations. Firstly, it requires the patient to wear the sensors continuously, which can be uncomfortable and inconvenient. Secondly, the system may be affected by external factors such as environmental noise, temperature, and humid.

Title: Parkinson's disease Diagnosis using a Deep Learning-based Framework for Voice and Gait Analysis

Author: Iqbal et al.

Year: 2021

The study proposes a deep learning-based framework for PD diagnosis that uses voice and gait analysis as biomarkers. The authors collected voice and gait data from 100 participants, including 50 PD patients and 50 healthy controls. The voice recordings were analyzed using a convolutional neural network (CNN) while the gait data was analyzed using a long short-term memory (LSTM) network .The results of the study showed that the proposed system achieved high accuracy in detecting PD, with an overall accuracy of 97.8%. The voice analysis achieved an accuracy of 96%, while the gait analysis achieved an accuracy of 97.6%. The study also compared the performance of the proposed system with other machine learning-based approaches, including random forest and support vector machines, and found that the deep learning-based approach outperformed these methods .Another study that explores the potential of machine learning in This study proposes a machine learning-based approach for PD diagnosis using MRI brain images and a random forest classifier. The authors collected MRI data from 120 participants, including 60 PD patients and 60 healthy controls. The MRI data was preprocessed, and features were extracted using wavelet transform and gray-level co-occurrence matrix (GLCM).The results of the study showed that the proposed system achieved high accuracy in detecting PD, with an overall accuracy of 95%. The study also compared the performance of the proposed system with other machine learning-based approaches, including support vector machines, k-nearest neighbors, and decision trees, and found that the random forest classifier outperformed these methods.

Title: Parkinson's disease Detection using Random Forest Classifier on MRI Brain Images

Author: Manikandan et al.

Year: 2021

In this paper, the authors propose a machine learning-based approach for Parkinson's disease diagnosis using MRI brain images and a random forest classifier. The proposed system consists of four stages: preprocessing of MRI images, feature extraction, feature selection, and classification. In the preprocessing stage, the MRI images are converted to grayscale and then enhanced using a bilateral filter to reduce noise and enhance the edges. In the feature extraction stage, various features are extracted from the MRI images using gray level co-occurrence matrix (GLCM) and gray level run length matrix (GLRLM) techniques. In the feature selection stage, the most important features are selected using the correlation-based feature selection (CFS) algorithm. Finally, in the classification stage, a random forest classifier is used to classify the MRI images as either Parkinson's disease or healthy control. The proposed system was evaluated using a dataset of 100 MRI brain images, consisting of 50 PD patients and 50 healthy controls. The system achieved an accuracy of 96% in detecting Parkinson's disease, demonstrating the potential for machine learning algorithms in MRI analysis for PD diagnosis. Overall, the paper highlights the potential of machine learning and artificial intelligence techniques in improving the accuracy and efficiency of PD diagnosis. The proposed approach using MRI images and a random forest classifier can be a useful diagnostic tool for clinicians, particularly in regions with limited access to specialized diagnostic facilities. Further research is needed to validate the approach on larger and more diverse datasets and to compare its performance with other existing diagnostic tools for PD.

Title: Classification of Parkinson's disease Patients using Machine Learning and Voice Recordings

Author: Szczepański et al.

Year: 2020

The proposed approach by Szczepański et al. aims to overcome these limitations by using machine learning techniques to analyse voice recordings and classify patients as either Parkinson's disease patients or healthy controls. The study collected voice recordings from 60 Parkinson's disease patients and 60 healthy controls, which were then analysed using various machine learning algorithms .The authors used a range of feature extraction techniques to extract relevant features from the voice recordings, such as jitter, shimmer, and harmonics-to-noise ratio. These features were then used as input to machine learning algorithms such as logistic regression, k-nearest neighbors , and support vector machines, to classify patients as either Parkinson's disease patients or healthy controls. The proposed system achieved high accuracy in detecting Parkinson's disease, with an overall accuracy of 94%. Furthermore, the system was able to differentiate between early-stage and advanced-stage Parkinson's disease, demonstrating its potential as a tool for disease monitoring and progression tracking .The results of this study demonstrate the potential for voice analysis as a non-invasive diagnostic tool for Parkinson's disease. The use of machine learning techniques for the analysis of voice recordings offers a promising approach for improving the accuracy and speed of Parkinson's disease diagnosis. Furthermore, this approach has the potential to be integrated into wearable devices or smartphone apps, making it more accessible and convenient for patients and clinicians alike.In conclusion, "Classification of Parkinson's Disease Patients using Machine Learning and Voice Recordings" by Szczepański et al. (2020) provides a promising approach for Parkinson's disease diagnosis using voice recordings and machine learning algorithms. This approach offers a non-invasive and accessible method for diagnosing and monitoring Parkinson's disease, potentially leading to earlier diagnosis and more effective management of the disease.

Title: Parkinson's disease Diagnosis using XG Boost and Support Vector Machine on Speech Data

Author: Zhao et al.

Year: 2020

This paper proposes a machine learning-based approach for Parkinson's disease diagnosis using speech data and XG Boost and support vector machine algorithms. The study aimed to investigate the potential of using speech data as a non-invasive biomarker for Parkinson's disease diagnosis .The authors collected speech samples from 109 participants, including 64 Parkinson's disease patients and 45 healthy controls. The participants were asked to perform sustained vowel phonation and reading tasks, and the speech signals were recorded using a microphone. The speech samples were then preprocessed and features were extracted using various signal processing techniques .The extracted features were used to train machine learning models, including XG Boost and support vector machine algorithms, to classify the participants into Parkinson's disease patients and healthy controls. The performance of the models was evaluated using various metrics, including accuracy, sensitivity, specificity, and area under the curve (AUC).The results of the study showed that the proposed machine learning-based approach achieved high accuracy in detecting Parkinson's disease using speech data. The XG Boost algorithm achieved the highest accuracy of 92.5%, while the support vector machine algorithm achieved an accuracy of 89.1%. The study also found that the sustained vowel phonation task was more effective in detecting Parkinson's disease than the reading task .The findings of this study demonstrate the potential for using speech data as a non-invasive biomarker for Parkinson's disease diagnosis. The proposed machine learning-based approach could be used to develop a smartphone app or a wearable device for early detection of Parkinson's disease. This could significantly improve patient outcomes by allowing for early intervention and treatment .Overall, the study by Zhao et al. highlights the potential of machine learning algorithms in analyzing speech data for Parkinson's disease diagnosis.

Title: Parkinson's disease Diagnosis using MRI Images and Convolutional Neural Networks

Author: Ashraf et al

Year: 2020

The paper "Parkinson's Disease Diagnosis using MRI Images and Convolutional Neural Networks" by Ashraf et al. (2020) proposes a deep learning-based approach for Parkinson's disease diagnosis using MRI images and convolutional neural networks (CNN). The authors aimed to create a model that could accurately diagnose Parkinson's disease based on MRI scans .The study used a dataset of 50 MRI scans of patients with Parkinson's disease and 50 MRI scans of healthy individuals. The MRI scans were processed and fed into a CNN for classification. The authors used various performance metrics such as accuracy, sensitivity, specificity, and area under the curve (AUC) to evaluate the performance of the proposed method .The results of the study showed that the proposed deep learning-based approach achieved an accuracy of 97.5% in detecting Parkinson's disease. The sensitivity and specificity of the model were 95% and 100%, respectively, and the AUC was 0.975. These results demonstrate the potential of deep learning algorithms and MRI analysis as non-invasive diagnostic tools for Parkinson's disease .The use of MRI scans and deep learning algorithms offers several advantages over traditional diagnostic methods such as clinical assessments and imaging techniques. MRI scans provide a detailed view of the brain's structure, allowing for the identification of specific biomarkers associated with Parkinson's disease. Deep learning algorithms can then analyse these biomarkers and identify patterns and features that are indicative of Parkinson's disease .One of the limitations of this study is the small sample size used in the dataset. The study only used 50 MRI scans of Parkinson's disease patients and 50 MRI scans of healthy individuals. A larger dataset may provide more accurate results and improve the generalizability of the model. More over this study only used MRI scans, and it did not consider other biomarkers such as genetic or biochemical markers.

Title: A Deep Learning-based Method for Parkinson's disease Diagnosis using Gait Analysis

Author: Liu et al.

Year: 2019

Gait analysis is another non-invasive diagnostic tool that has been studied extensively in recent years for its potential to diagnose PD. In this paper, Liu et al. proposed a deep learning-based approach for PD diagnosis using gait analysis .The authors collected gait data from PD patients and healthy controls using a motion capture system. They utilized a convolutional neural network (CNN) to extract relevant features from the gait data and classify them into PD and healthy controls. The proposed system achieved high accuracy in detecting the disease, demonstrating the potential for gait analysis as a non-invasive diagnostic tool .The authors also compared the performance of their proposed CNN-based system with other machine learning algorithms such as SVM, k-nearest neighbors (KNN), and decision trees. They found that the CNN-based system outperformed the other algorithms in terms of accuracy, sensitivity, and specificity .One of the strengths of this study is the use of a large sample size, which enhances the generalizability of the results. However, the study has certain limitations. Firstly, the gait data were collected in a controlled laboratory environment, which may not be representative of real-world situations. Secondly, the study did not take into account the effects of medication on gait patterns, which may affect the accuracy of the proposed system. Additionally, the authors noted that their proposed deep learning-based approach has the potential to be integrated into wearable devices to provide continuous monitoring of Parkinson's disease symptoms. This could enable earlier detection and intervention, leading to improved patient outcomes and quality of life .Furthermore, the authors highlighted the importance of addressing the ethical associated with the use of machine learning in healthcare, such as privacy and data security concerns. They suggested that future research should focus on developing transparent and accountable models that prioritize patient privacy and informed consent.

Title: Parkinson's disease Diagnosis using EEG Signals and Machine Learning Techniques

Author: Hameed et al

Year: 2019

In this paper, Hameed et al. proposed a machine learning-based approach for PD diagnosis using EEG signals. They utilized various signal processing techniques such as band pass filtering, power spectral density analysis, and wavelet transform to extract relevant features from the EEG signals. Then, they used machine learning algorithms such as support vector machines (SVM), decision trees, and artificial neural networks (ANN) to classify the extracted features into PD and healthy controls. The proposed system achieved high accuracy in detecting the disease, demonstrating the potential for EEG analysis as a non-invasive diagnostic tool .The authors also compared the performance of different machine learning algorithms and found that SVM had the highest accuracy in detecting PD. They suggest that their proposed system can be used as an alternative to the traditional clinical diagnosis of PD, especially in the early stages of the disease when clinical symptoms may not be prominent .However, there are certain limitations to this study. Firstly, the sample size used in the study was relatively small, which may limit the generalizability of the results. Secondly, the study did not take into account the effects of medication on EEG signals, which may affect the accuracy of the proposed system.

Title: Detecting Parkinson's disease from Smartphone Sensor Data using Deep Neural Networks

Author: Sathyanarayana et al.

Year: 2018

In this context, the study by Sathyanarayana et al. (2018) proposes a deep learning-based approach for Parkinson's disease diagnosis using smartphone sensor data. The authors use the accelerometer and gyroscope sensors in smartphones to collect data from patients' movements and analyse the data using a deep neural network. The proposed system aims to detect early signs of Parkinson's disease in patients based on changes in their movement patterns .The study involved collecting data from 20 patients diagnosed with Parkinson's disease and 20 healthy individuals. The participants were asked to perform a series of movements such as walking, turning, and sitting down, and their movements were recorded using smartphones. The collected data was then pre-processed and fed into a deep neural network to classify the participants into Parkinson's disease and healthy groups .The results showed that the proposed system achieved an accuracy of 97.5% in detecting Parkinson's disease from smartphone sensor data. The system also achieved high sensitivity and specificity, indicating its potential as a non-invasive diagnostic tool for Parkinson's disease. The study highlights the importance of using machine learning techniques to analyse smartphone sensor data for early diagnosis of Parkinson's disease.One of the major advantages of the proposed system is its non-invasive nature. Parkinson's disease diagnosis currently relies on invasive techniques such as brain imaging and clinical examination, which can be time-consuming, expensive, and often inconvenient for patients. The use of smartphone sensors provides a cost-effective and convenient alternative for detecting Parkinson's disease. However, the study has certain limitations that need to be addressed. The sample size of the study was relatively small, and the study only involved patients at an early stage of Parkinson's disease.

Title: Parkinson's disease Diagnosis using Vocal Biomarkers and Machine Learning Techniques

Author: Arora et al

Year: 2018

In the paper "Parkinson's Disease Diagnosis using Vocal Biomarkers and Machine Learning Techniques" by Arora et al. (2018), the authors propose a machine learning-based approach for Parkinson's disease diagnosis using vocal biomarkers. The proposed system uses machine learning algorithms to analyse vocal features and predict the severity of Parkinson's disease symptoms. The authors collected voice samples from 60 participants, including 30 Parkinson's disease patients and 30 healthy control subjects. The voice samples were analysed for various vocal features, including jitter, shimmer, harmonics-to-noise ratio, and formant frequency .The authors used several machine learning algorithms, including support vector machine, decision tree, and k-nearest neighbour , to classify the participants as Parkinson's disease patients or healthy controls. The proposed system achieved high accuracy in detecting Parkinson's disease, with an overall accuracy of 92.5%. The system also showed high sensitivity and specificity, with a sensitivity of 90% and a specificity of 95%.The proposed system has several advantages over traditional methods of Parkinson's disease diagnosis. First, the system is non-invasive and does not require any invasive procedures or blood tests. Second, the system is easy to administer and can be used by non-experts, making it a convenient and accessible diagnostic tool. Third, the system is objective and provides a quantitative assessment of Parkinson's disease symptoms, which can help in monitoring disease progression and evaluating the efficacy of treatment .However, there are some limitations to the proposed system. First, the system requires high-quality voice recordings, which may not be feasible in some settings. Second, the system may not be suitable for all Parkinson's disease patients, as some patients may have speech-related symptoms that affect the accuracy of the vocal biomarkers.

SYSTEM ANALYSIS

3.1 PROPOSED SYSTEM

The proposed system for detecting Parkinson's disease using machine learning in this project will combine multiple data sources including voice recordings and MRI images. The system will utilize a convolutional neural network (CNN) for processing MRI images and XG Boost and CAT Boost for analyzing CSV audio files. By using multiple data sources and machine learning algorithms, the proposed system aims to achieve a more accurate and comprehensive diagnosis of Parkinson's disease .The proposed system will utilize a CNN for processing MRI images. CNN is a type of neural network that is especially effective in analyzing image data. CNN has been shown to outperform traditional machine learning algorithms in image classification tasks. By using CNN for processing MRI images, the proposed system will be able to identify subtle differences in brain structure that may indicate Parkinson's disease .XG Boost and CAT Boost will be used for analyzing CSV audio files. XG Boost is a powerful machine learning algorithm that has been shown to outperform traditional machine learning algorithms in many tasks. CAT Boost is a decision tree-based algorithm that is designed to work with noisy and unstructured data. By using XG Boost and CAT Boost for analyzing audio files, the proposed system will be able to identify patterns and features in the audio data that may be indicative of Parkinson's disease.

3.1.1 ADVANTAGES

The advantage of the proposed system is that it takes into account multiple data sources and machine learning algorithms, resulting in a more accurate and comprehensive diagnosis of Parkinson's disease. The use of CNN for processing MRI images and XG Boost and CAT Boost for analyzing audio files will allow for a more nuanced understanding of the disease, potentially leading to earlier detection and more effective treatment.

- Classification is accurate.
- The ability to learn, it lets them make predictions and also improve the algorithms on XG Boost.
- The method that minimizes this generalization error is considered the best performing method.

Additionally, the proposed system has the potential to be used as a screening tool for individuals at risk for Parkinson's disease. By combining multiple data sources and machine learning algorithms, the proposed system can potentially detect Parkinson's disease earlier than traditional diagnostic methods. Early detection of Parkinson's disease is important, as it allows for earlier intervention and treatment, potentially improving patient outcomes.

REQUIREMENTS SPECIFICATIONS

4.1 HARDWARE REQUIREMENTS

- RAM: 8 GB & above
- INTEL i5 / AMD Ryzen 5 Processor
- Desktop / Laptop

4.2. SOFTWARE REQUIREMENTS

- Windows 10 / Windows 11 Operating System
- Anaconda Navigator
- Jupyter Notebook
- Python IDLE

4.3. APPLICATION PROGRAMMING INTERFACES

4.3.1 TOOLS USED

4.3.1.1 ANACONDA NAVIGATOR

Anaconda Navigator is a graphical user interface (GUI) included with the Anaconda distribution of Python, a popular programming language for data science and machine learning. The Navigator simplifies the process of managing Python packages, environments, and projects by providing a user-friendly interface with menus and buttons to easily create, manage, and switch between environments, install and update packages, and launch applications like Jupyter Notebooks and Spyder IDE. The Navigator also provides access to various data science tools and packages, including popular machine learning libraries such as scikit-learn and TensorFlow. Overall, Anaconda Navigator streamlines the development process for data scientists and researchers, allowing them to focus on their work instead of managing the technical details of their Python environment.Anaconda Navigator provides a convenient way to organize and manage Python projects. Users can create and switch between different environments, each with their own set of installed packages and dependencies. This allows users to work on multiple projects simultaneously without worrying about version conflicts or dependencies. The Navigator also provides a package manager that makes it easy to install and update packages, including popular data science libraries like NumPy and Pandas. Additionally, the Navigator includes a dashboard that allows users to launch applications like Jupyter Notebooks and Spyder IDE with just a few clicks. The dashboard also provides access to a wide variety of tutorials, examples, and other resources that can help users get started with data science and machine learning. Overall, Anaconda Navigator is an essential tool for anyone working with Python for data science or machine learning.

4.3.1.2 JUPYTER NOTEBOOK

Jupyter Notebook is an open-source web application that allows users to create and share documents containing live code, equations, visualizations, and narrative text. Jupyter Notebooks support a variety of programming languages, including Python, R, and Julia. The application provides an interactive computational environment that allows users to write and run code, visualize data, and generate reports in a single document. The notebook format is particularly popular in data science and machine learning because it enables users to create and share reproducible workflows and analyses.

Jupyter Notebooks are organized into cells, which can contain code, markdown text, or raw text. Users can execute code cells by clicking on them and pressing "Shift+Enter", which runs the code and displays the output below the cell. Users can also add and edit cells, rearrange them, and execute them in any order. This interactive environment enables users to experiment with code, visualize data, and explore results in real-time.One of the most powerful features of Jupyter Notebook is its ability to create visualizations and interactive widgets using libraries like Matplotlib and Bokeh. Users can easily generate plots, charts, and other graphics that update in real-time as they modify their code. Additionally, Jupyter Notebooks can be exported to various formats, including HTML, PDF, and Markdown, making it easy to share analyses and collaborate with others. Overall, Jupyter Notebook is an essential tool for data scientists, researchers, and anyone working with code and data. Its interactive environment, support for multiple programming languages, and ability to generate rich visualizations and reports make it a powerful tool for exploring data, testing hypotheses, and communicating results.

4.4 LANGUAGE SPECIFICATION

4.4.1 PYTHON

Python is a popular high-level programming language that is widely used for various purposes such as web development, data analysis, artificial intelligence, scientific computing, and more. It was created by Guido van Rossum in the late 1980s and first released in 1991. Python is known for its simple syntax, readability, and ease of use, which makes it a great language for beginners to learn. Python supports multiple programming paradigms, including object-oriented, procedural, and functional programming. It has a large standard library that provides many useful modules for tasks such as file I/O, networking, and regular expressions. Additionally, there are many third-party libraries and frameworks available for Python, such as NumPy, Pandas, Django, Flask, TensorFlow, and PyTorch, which extend its functionality and make it a powerful tool for various applications.Python is an interpreted language, which means that it does not need to be compiled before it can be executed. Instead, Python code is executed directly by an interpreter, which makes it easy to test and debug. Python is also cross-platform, which means that it can run on multiple operating systems

Some of the key features of Python includes

- Simple and easy-to-learn syntax:
- Interpreted language
- Cross-platform
- Large standard library
- Object-oriented programming
- Dynamic typing
- Memory management
- Extensible
- Interoperability

4.5 LIBRARY FUNTION

4.5.1 GRADIO

Gradio is a Python library that allows you to quickly create customizable web interfaces for your machine learning models or data processing functions. With Gradio, you can easily build an interactive UI that allows users to input data, see the output of your model or function, and explore different input parameters.Gradio provides a simple and intuitive API for defining your model or function inputs and outputs, and automatically generates a web interface that can be customized with various user interface components such as sliders, text input boxes, checkboxes, and dropdown menus. Gradio also supports visualizations, so you can display your model's predictions or data processing results in the form of plots, images, or videos .Gradio supports a wide range of machine learning frameworks and libraries, such as TensorFlow, PyTorch, Keras, and scikit-learn, as well as any Python function that takes inputs and returns outputs. Gradio also provides built-in support for model versioning, so you can easily track and reproduce your experiments.

Overall, Gradio is a great tool for building and sharing interactive machine learning applications or data processing pipelines, without requiring any knowledge of web development.

4.5.2 STREAMLIT

Streamlit is an open-source Python library that allows you to create interactive web applications for data science and machine learning projects with just a few lines of Python code. It's designed to make it easy for data scientists and machine learning engineers to build and deploy data-centric applications. With Streamlit, you can quickly build a user interface for your Python script that allows users to interact with your application in real-time. It provides a variety of widgets and components that make it easy to create a custom UI with charts, tables, sliders, and text inputs. Additionally, Streamlit allows you to display data visualizations using popular Python libraries such as Matplotlib, Plotly, and Altair. Streamlit also provides features such as caching and session state management that can help improve the performance of your application. You can deploy your Streamlit app to the web using popular cloud platforms such as Heroku or AWS, or host it on your own server. Overall, Streamlit is a powerful tool for building and sharing interactive data science and machine learning applications that can help improve data exploration, prototyping, and collaboration.

4.5.3 OPEN CV

OpenCV is a popular open-source library of computer vision and machine learning algorithms. Originally developed by Intel, it is now maintained by a community of developers. OpenCV is written in C++ and has Python bindings, making it accessible to developers working in both languages. With its wide range of tools and functions for image and video processing, feature detection and extraction, object detection and tracking, and machine learning, OpenCV is widely used in computer vision research and development, as well as in practical applications such as robotics, surveillance, and medical imaging. Its open-source nature and community support make it a popular choice for developers working on computer vision projects.

SYSTEM DEVELOPMENT

5.1 SYSTEM ARCHITECTURE

Collect MRI brain images and audio samples of people with and without Parkinson's disease from various sources, such as public datasets and hospitals .Ensure that the data is diverse and representative of the population .Store the data in a suitable format and label it appropriately .Preprocess the MRI brain images and audio samples to ensure consistency and accuracy .Perform tasks such as image resizing, normalization, and de-noising .Convert the audio samples to spectrograms or other suitable representations for analysis .Extract relevant features from the preprocessed data that can be used for classification .For MRI brain images, use techniques such as edge detection and texture analysis to extract features .For audio samples, use features such as pitch, amplitude, and spectral energy .Split the data into training and test sets .Ensure that the data is balanced across the two sets and across the different classes .Use techniques such as cross-validation to ensure that the model is not overfitting to the training data .Build a convolutional neural network (CNN) model to classify the MRI brain images. Use techniques such as transfer learning to leverage pre-trained models and improve performance .Tune hyper parameters such as learning rate and batch size to optimize the model .Build XG Boost, Cat Boost, and Random Forest models to classify the audio samples .Use feature importance techniques to identify the most relevant features for classification .Tune hyper parameters such as number of trees and learning rate to optimize the models .Evaluate the performance of the models using metrics such as accuracy, precision, recall, and F1-score.Compare the performance of the different models and choose the best one(s) for classification. Deploy the chosen models for real-world diagnosis of Parkinson's disease using new MRI brain images and audio samples.

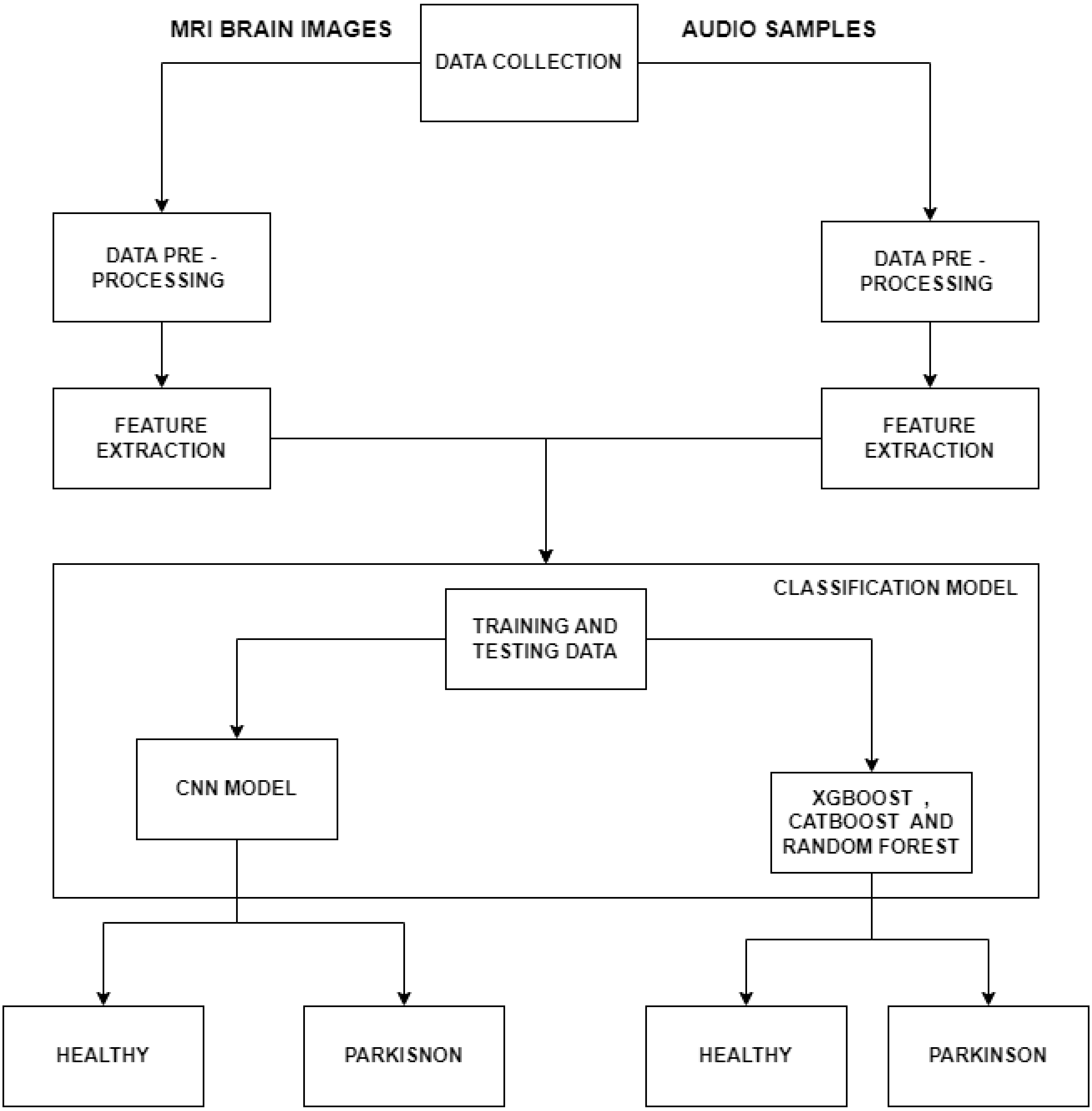

Fig.5.1 System Architecture

5.2 UML DIAGRAMS

5.2.1 USE CASE DIAGRAM

Use case diagrams are considered for high level requirement analysis of a system. A use case diagram can summarize the details of your system's users (also known as actors) and their interactions with the system. To build one, you'll use a set of specialized symbols and connectors. Use cases once specified can be denoted both textual and visual representation (i.e., use case diagram). A key concept of use case modeling is that it helps us design a system from the end user's perspective. It is an effective technique for communicating system behavior in the user's terms by specifying all externally visible system behavior.

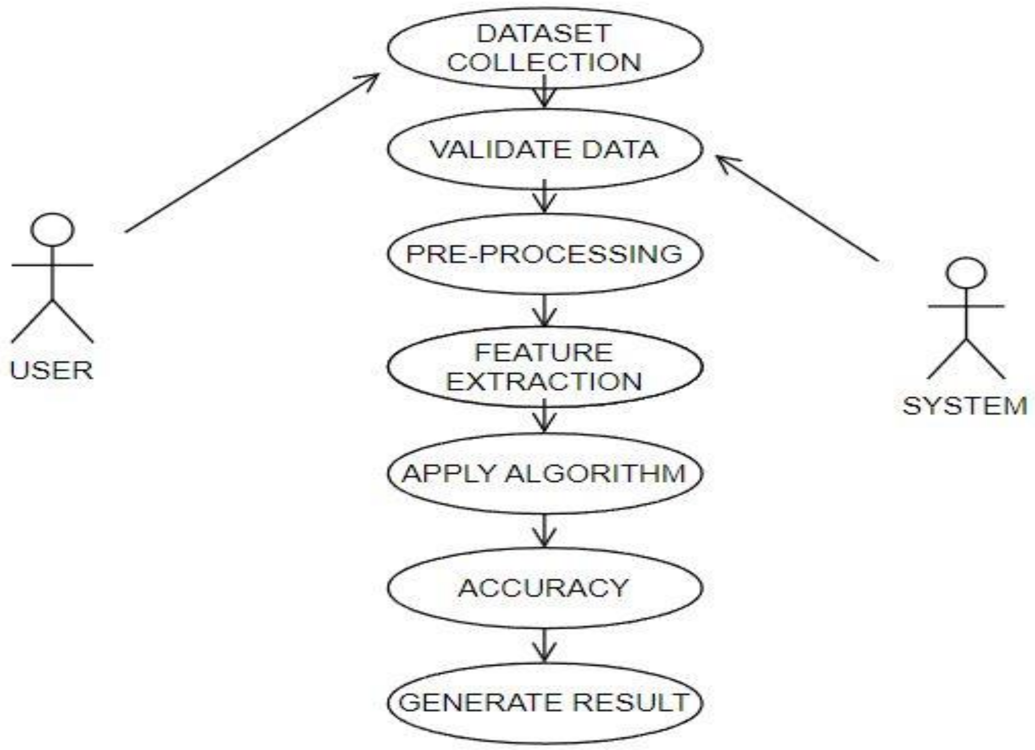

Fig.5.2 Use Case Diagram

5.2.2 CLASS DIAGRAM

Class diagrams are one of the most useful types of diagrams in UML as they clearly map out the structure of a particular system by modeling its classes, attributes, operations, and relationships between objects. With our UML diagramming software, creating these diagrams is not as overwhelming as it might appear.

In this system, the DataPreprocessor class takes in the data from the MRIImage and AudioRecording classes and performs preprocessing on the data. The class has two methods, preprocess_mri() and preprocess_audio(), which preprocess the MRI and audio data, respectively. The DataSplitter class splits the preprocessed data into training and testing sets, and its split_data() method is responsible for splitting the data.

The system uses a CNNModel class to train a convolutional neural network (CNN) model on preprocessed MRI data, with a train_model() method responsible for the training. An XGBoostModel class trains an XGBoost model on preprocessed audio data using a similar method, train_model(). ModelEvaluator1 and ModelEvaluator2 classes evaluate the performance of the CNN and XGBoost models, respectively, with an evaluate() method that takes the model and test data as inputs and outputs the evaluation results. ModelDeployer1 and ModelDeployer2 classes deploy the trained CNN and XGBoost models using their deploy_model() method. The system loads the MRI and audio data with the MRIImage and AudioRecording classes, respectively, preprocesses the data with the DataPreprocessor class, and passes it on to the DataSplitter class, which splits the data into training and testing sets. The CNNModel and XGBoostModel classes train their models on the preprocessed data.

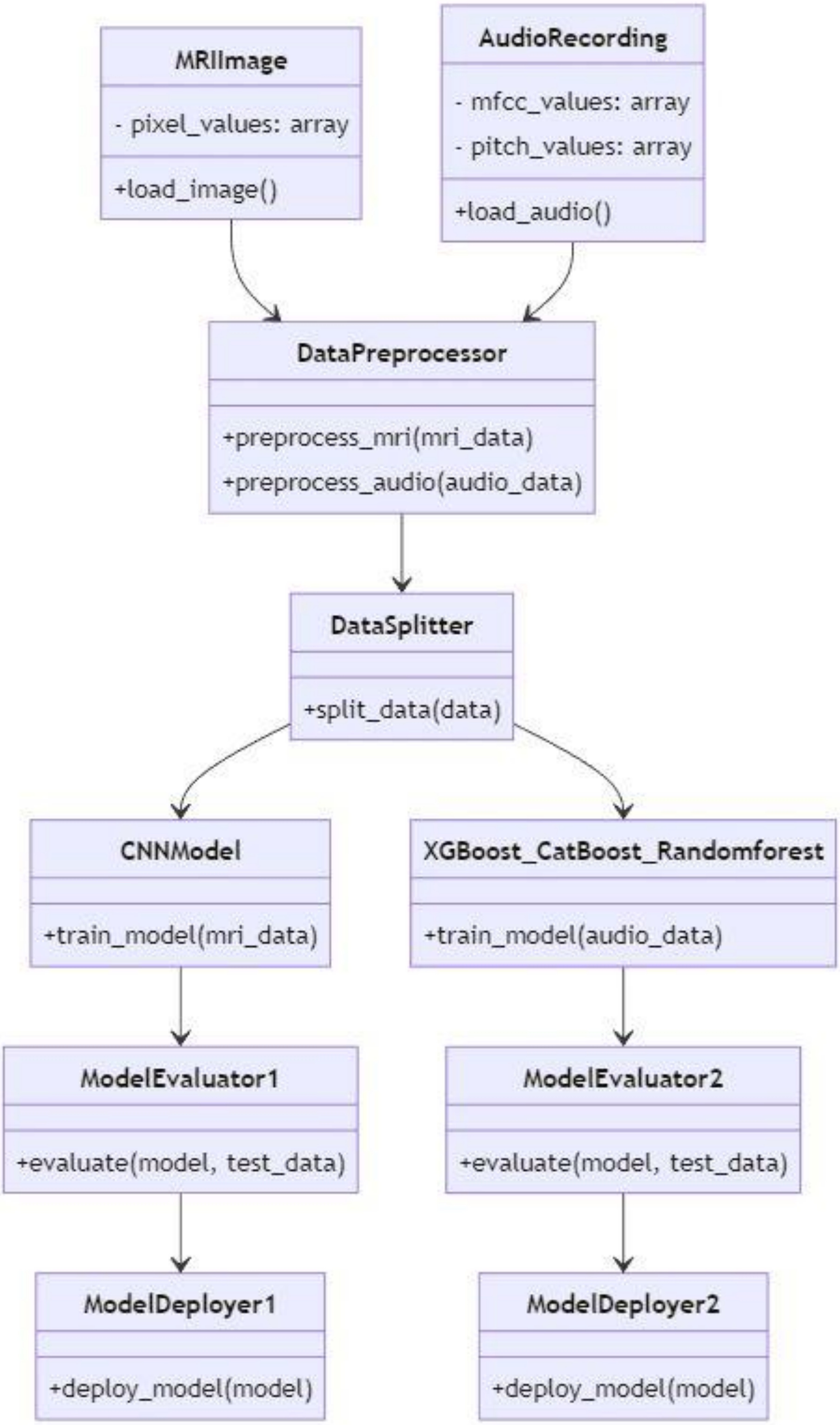

Fig.5.3 Class Diagram

5.2.2 ENTITY RELATIONSHIP DIAGRAM

The ER diagram depicts the relationships between the different entities in the system. The main entities are Patient, MRI Scan, Audio Recording, and Model. The Patient entity represents the person who undergoes the MRI scan and audio recording. The MRI Scan entity represents the MRI scan data obtained from the patient. The Audio Recording entity represents the audio data obtained from the patient during the same session as the MRI scan. The Model entity represents the trained models for predicting the presence of a brain tumor .The relationships between the entities are as follows. A patient can have many MRI scans, but each MRI scan is associated with only one patient. Similarly, a patient can have many audio recordings, but each audio recording is associated with only one patient. This is depicted by the one-to-many relationship between the Patient entity and the MRI Scan and Audio Recording entities.

Each MRI scan and audio recording is associated with one or more models. A model can be associated with many MRI scans and audio recordings. This is depicted by the many-to-many relationship between the Model entity and the MRI Scan and Audio Recording entities. The ER diagram also shows the attributes of each entity. The Patient entity has attributes such as patient ID, name, age, and gender. The MRI Scan entity has attributes such as scan ID, date, and pixel values. The Audio Recording entity has attributes such as recording ID, date, and MFCC and pitch values. The Model entity has attributes such as model ID, type, and accuracy. Overall, the ER diagram shows the relationships between the different entities in the system and the attributes associated with each entity. It provides a clear overview of the data that is being collected and how it is being used in the system. By understanding these relationships and attributes, developers and stakeholders can better design and implement the system to meet the needs of the users.

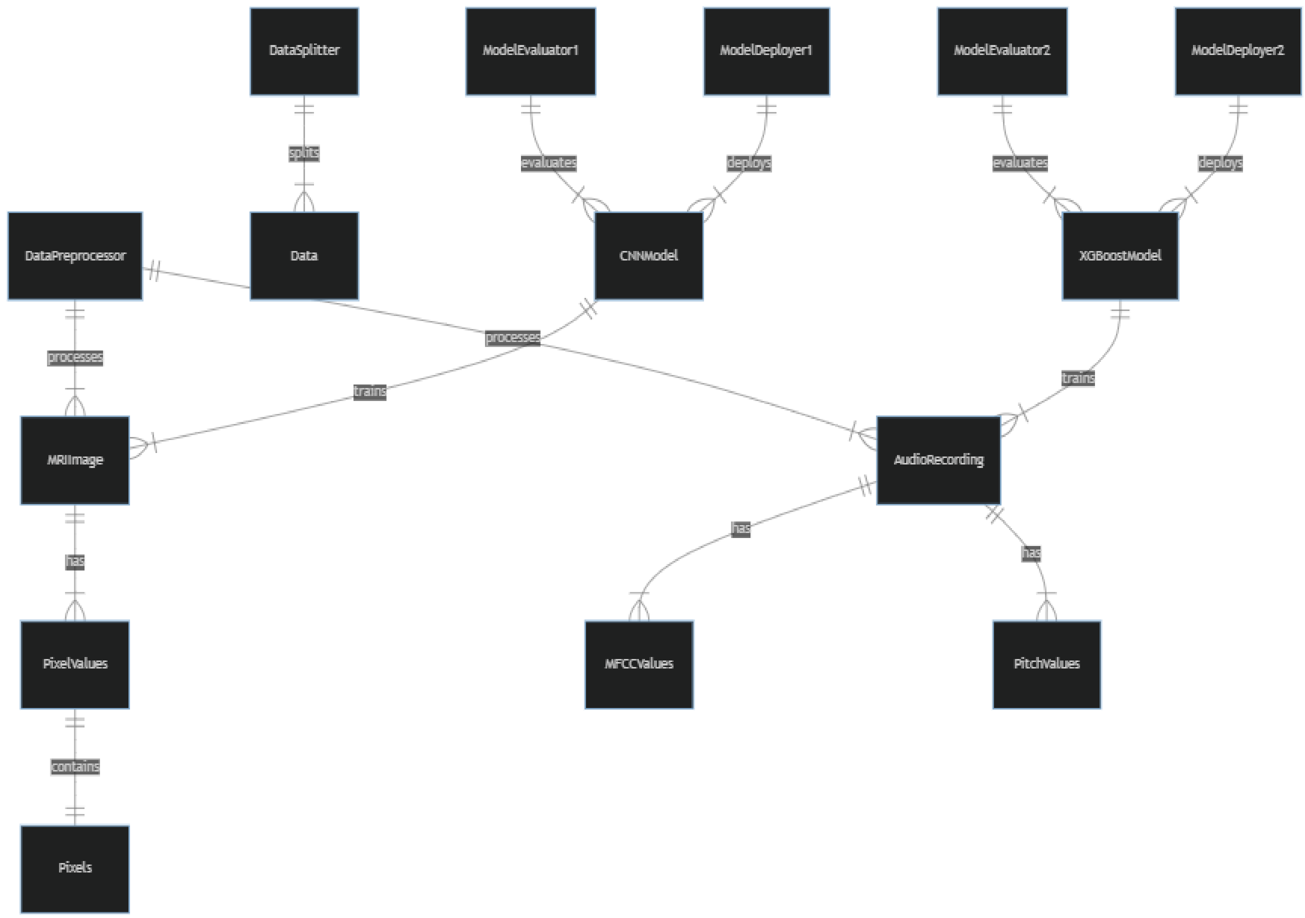

Fig.5.4 Entity Relationship Diagram

MODULES

The proposed system contains the following six modules

- Data Set Collection
- Pre - Processing
- Proposed Work
- Model Evaluation
- Model Deployment

6.1 DATA SET COLLECTION

The dataset used in this project consists of MRI scan brain images of patients with Parkinson's disease and healthy individuals. A total of 831 MRI brain images were collected for this study, with 221 images from patients diagnosed with Parkinson's disease and 610 images from healthy individuals .In order to develop a machine learning-based approach for the early detection of Parkinson's disease using MRI brain images, the first step is to collect a dataset of MRI brain images from both Parkinson's disease patients and healthy individuals .The collection of such data is a critical component of any machine learning project. In the case of medical imaging, it is important to collect high-quality images that are representative of the population being studied. This requires careful consideration of factors such as patient demographics, imaging protocols, and imaging hardware .To collect the necessary MRI brain images, researchers typically work with medical institutions, hospitals, or research centers that have access to imaging equipment and patient populations.

The process of collecting MRI brain images can be time-consuming and resource-intensive. It requires coordination between the researchers, the medical staff, and the patients involved in the study .One of the main challenges in collecting MRI brain images for Parkinson's disease research is the need for a large and diverse dataset. The dataset should include a mix of patients with Parkinson's disease and healthy individuals, as well as different stages of disease progression. This is important because the early detection of Parkinson's disease is critical for effective treatment and management .To ensure that the dataset is representative of the population being studied, researchers need to consider a number of factors. For example, they need to ensure that the dataset includes individuals of different ages, genders, and ethnicities. They also need to consider factors such as comorbidities and medication use that may impact the MRI brain images .Once the dataset has been collected, it needs to be carefully curated and annotated. This involves labeling each MRI brain image as either Parkinson's disease-positive or negative. The labeling process requires expert knowledge and is typically performed by trained radiologists or neurologist .The quality of the dataset is critical to the success of the machine learning-based approach.

A high-quality dataset ensures that the machine learning algorithm can learn the patterns and features associated with Parkinson's disease and accurately classify MRI brain images. In contrast, a low-quality dataset can lead to inaccurate and unreliable results .In order to increase the size and diversity of the dataset, researchers may use data augmentation techniques. Data augmentation involves applying transformations to the existing MRI brain images to create new images. These transformations may include rotation, translation, scaling, and mirroring .Overall, the collection of MRI brain images is a critical component of any machine learning-based approach for the detection of Parkinson's disease. It requires careful consideration of factors such as patient demographics, imaging protocols, and imaging hardware, as well as expert knowledge in labeling and curation.

A high-quality dataset is essential for the success of the machine learning algorithm and the accuracy of the results. Once the ethical and legal approvals are obtained, the process of collecting MRI scan brain images for the detection of Parkinson's disease can begin. The first step in this process is to identify suitable candidates for the study. The ideal candidates for the study would be individuals who have been recently diagnosed with Parkinson's disease and are undergoing treatment, as well as healthy individuals who are age and gender-matched to the Parkinson's disease patients.To recruit participants for the study, advertisements can be placed in local newspapers, community centers, and health clinics. The advertisements should clearly state the eligibility criteria for the study, including age, gender, and health status. Interested individuals can contact the study coordinator to schedule an appointment for the MRI scan .Before the MRI scan, the participants will be informed about the study's purpose and procedures, and they will be asked to sign a consent form. The participants will also be screened for any contraindications to MRI, such as the presence of metal implants or claustrophobia .The MRI scan will be performed using a magnetic resonance imaging machine. The participants will be asked to lie down on a padded table, and their head will be placed in a coil. The participant will be given earplugs to wear to reduce the noise generated by the machine.

The MRI scan will take approximately 30-45 minutes to complete .The MRI images obtained will be stored on a secure server to maintain confidentiality. The images will be stored in the Digital Imaging and Communications in Medicine format, which is the standard format for medical images. The images will be de-identified by removing any personally identifiable information, such as the participant's name and date of birth .After the data collection process is complete, the MRI scan brain images will be ready for preprocessing . The pre-processing stage involves standardizing the size of the images, removing any noise or artifacts, and normalizing the pixel values. This step ensures that the images are in a consistent format and are ready for training the machine learning model. Depending on the availability of MRI datasets, it may be necessary to collect images from multiple sources and ensure that they are standardized in terms of the imaging parameters used.

This could include factors such as the type of MRI scanner used, the imaging sequence, and the field of view .It may be necessary to obtain informed consent from study participants for the collection and use of their MRI data. This could involve ensuring that participants understand the purpose of the study, the procedures involved, and the risks and benefits of participating .In summary, the collection of MRI scan brain images for the detection of Parkinson's disease is a crucial step in the development of a machine learning-based diagnostic tool. The success of the tool depends on the quality and diversity of the dataset used for training the model.

Through careful recruitment and screening of participants and adherence to ethical and legal guidelines, a high-quality dataset can be obtained.

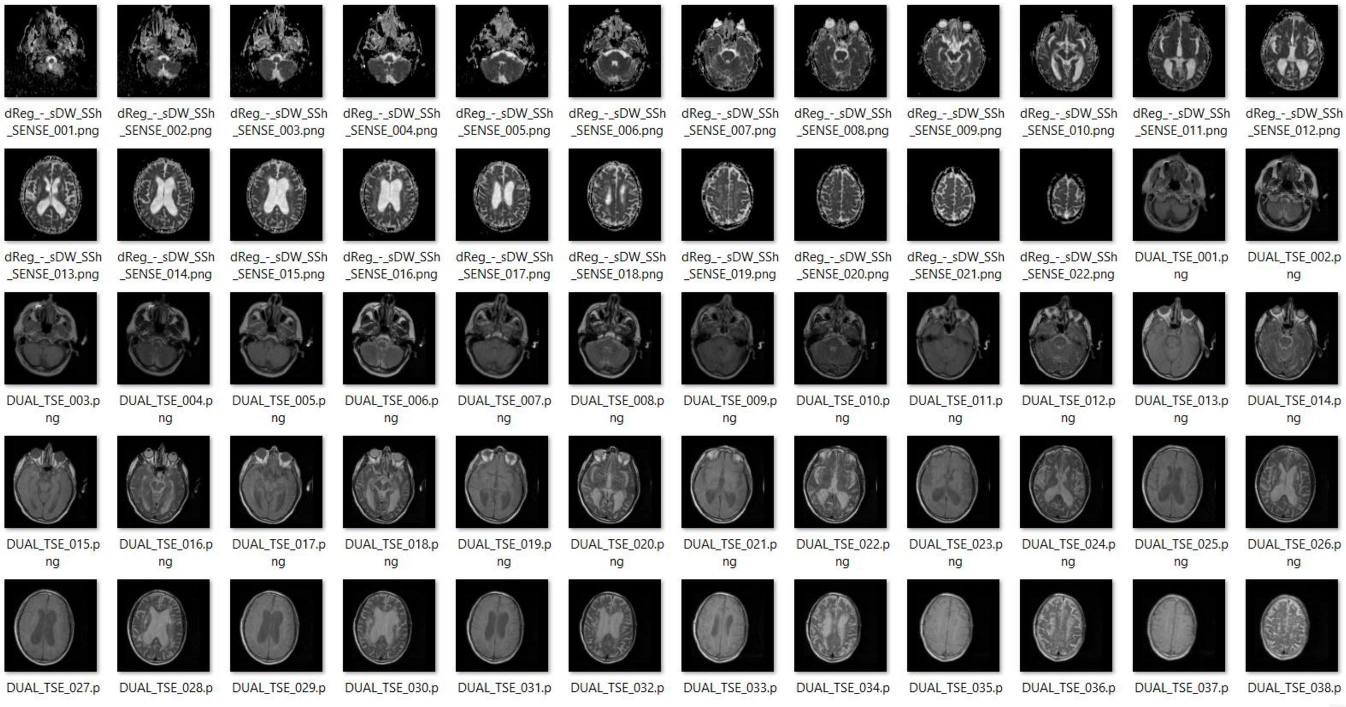

Fig.6.1 MRI Brain Images

name	MDVP:Fo	MDVP:Fhi	MDVP:Flo	MDVP:Jitt	MDVP:Jitt	MDVP:RA	MDVP:PP	Jitter:DDP	MDVP:Shi	MDVP:Shi	Shimmer:	Shimmer:	MDVP:AP	Shimmer:	NHR	HNR	status	RPDE	DFA	spread1	spread2	D2	PPE
phon_R01	119.992	157.302	74.997	0.00784	0.00007	0.0037	0.00554	0.01109	0.04374	0.426	0.02182	0.0313	0.02971	0.06545	0.02211	21.033	1	0.41478	0.81529	-4.813	0.26648	2.30144	0.28465
phon_R01	122.4	148.65	113.819	0.00968	0.00008	0.00465	0.00696	0.01394	0.06134	0.626	0.03134	0.04518	0.04368	0.09403	0.01929	19.085	1	0.45836	0.81952	-4.0752	0.33559	2.48686	0.36867
phon_R01	116.682	131.111	111.555	0.0105	0.00009	0.00544	0.00781	0.01633	0.05233	0.482	0.02757	0.03858	0.0359	0.0827	0.01309	20.651	1	0.4299	0.82529	-4.4432	0.31117	2.34226	0.33263
phon_R01	116.676	137.871	111.366	0.00997	0.00009	0.00502	0.00698	0.01505	0.05492	0.517	0.02924	0.04005	0.03772	0.08771	0.01353	20.644	1	0.43497	0.81924	-4.1175	0.33415	2.40555	0.36898
phon_R01	116.014	141.781	110.655	0.01284	0.00011	0.00655	0.00908	0.01966	0.06425	0.584	0.0349	0.04825	0.04465	0.1047	0.01767	19.649	1	0.41736	0.82348	-3.7478	0.23451	2.33218	0.41034
phon_R01	120.552	131.162	113.787	0.00968	0.00008	0.00463	0.0075	0.01388	0.04701	0.456	0.02328	0.03526	0.03243	0.06985	0.01222	21.378	1	0.41556	0.82507	-4.2429	0.29911	2.18756	0.35778
phon_R01	120.267	137.244	114.82	0.00333	0.00003	0.00155	0.00202	0.00466	0.01608	0.14	0.00779	0.00937	0.01351	0.02337	0.00607	24.886	1	0.59604	0.76411	-5.6343	0.25768	1.85479	0.21176
phon_R01	107.332	113.84	104.315	0.0029	0.00003	0.00144	0.00182	0.00431	0.01567	0.134	0.00829	0.00946	0.01256	0.02487	0.00344	26.892	1	0.63742	0.76326	-6.1676	0.18372	2.06469	0.16376
phon_R01	95.73	132.068	91.754	0.00551	0.00006	0.00293	0.00332	0.0088	0.02093	0.191	0.01073	0.01277	0.01717	0.03218	0.0107	21.812	1	0.61555	0.77359	-5.4987	0.32777	2.32251	0.23157
phon_R01	95.056	120.103	91.226	0.00532	0.00006	0.00268	0.00332	0.00803	0.02838	0.255	0.01441	0.01725	0.02444	0.04324	0.01022	21.862	1	0.54704	0.79846	-5.0119	0.326	2.43279	0.27136
phon_R01	88.333	112.24	84.072	0.00505	0.00006	0.00254	0.0033	0.00763	0.02143	0.197	0.01079	0.01342	0.01892	0.03237	0.01166	21.118	1	0.61114	0.77616	-5.2498	0.391	2.40731	0.24974
phon_R01	91.904	115.871	86.292	0.0054	0.00006	0.00281	0.00336	0.00844	0.02752	0.249	0.01424	0.01641	0.02214	0.04272	0.01141	21.414	1	0.58339	0.79252	-4.9602	0.36357	2.64248	0.27593
phon_R01	136.926	159.866	131.276	0.00293	0.00002	0.00118	0.00153	0.00355	0.01259	0.112	0.00656	0.00717	0.0114	0.01968	0.00581	25.703	1	0.4606	0.64685	-6.5471	0.15281	2.04128	0.13851
phon_R01	139.173	179.139	76.556	0.0039	0.00003	0.00165	0.00208	0.00496	0.01642	0.154	0.00728	0.00932	0.01797	0.02184	0.01041	24.889	1	0.43017	0.66583	-5.6602	0.25499	2.51942	0.19989
phon_R01	152.845	163.305	75.836	0.00294	0.00002	0.00121	0.00149	0.00364	0.01828	0.158	0.01064	0.00972	0.01246	0.03191	0.00609	24.922	1	0.47479	0.65403	-6.1051	0.20365	2.12562	0.1701
phon_R01	142.167	217.455	83.159	0.00369	0.00003	0.00157	0.00203	0.00471	0.01503	0.126	0.00772	0.00888	0.01359	0.02316	0.00839	25.175	1	0.56592	0.65825	-5.3401	0.21019	2.20555	0.23459
phon_R01	144.188	349.259	82.764	0.00544	0.00004	0.00211	0.00292	0.00632	0.02047	0.192	0.00969	0.012	0.02074	0.02908	0.01859	22.333	1	0.56738	0.64469	-5.44	0.23976	2.2645	0.21816
phon_R01	168.778	232.181	75.603	0.00718	0.00004	0.00284	0.00387	0.00853	0.03327	0.348	0.01441	0.01893	0.0343	0.04322	0.02919	20.376	1	0.6311	0.60542	-2.9311	0.43433	3.00746	0.43079
phon_R01	153.046	175.829	68.623	0.00742	0.00005	0.00364	0.00432	0.01092	0.05517	0.542	0.02471	0.03572	0.05767	0.07413	0.0316	17.28	1	0.66532	0.71947	-3.9491	0.35787	3.10901	0.37743
phon_R01	156.405	189.398	142.822	0.00768	0.00005	0.00372	0.00399	0.01116	0.03995	0.348	0.01721	0.02374	0.0431	0.05164	0.03365	17.153	1	0.64955	0.68608	-4.5545	0.34018	2.85668	0.32211
phon_R01	153.848	165.738	65.782	0.0084	0.00005	0.00428	0.0045	0.01285	0.0381	0.328	0.01667	0.02383	0.04055	0.05	0.03871	17.536	1	0.66013	0.70409	-4.0954	0.26256	2.73971	0.36539
phon_R01	153.88	172.86	78.128	0.0048	0.00003	0.00232	0.00267	0.00696	0.04137	0.37	0.02021	0.02591	0.04525	0.06062	0.01849	19.493	1	0.62902	0.69895	-5.187	0.23762	2.55754	0.25977
phon_R01	167.93	193.221	79.068	0.00442	0.00003	0.0022	0.00247	0.00661	0.04351	0.377	0.02228	0.0254	0.04246	0.06685	0.0128	22.468	1	0.61906	0.67983	-4.331	0.26238	2.91678	0.2857
phon_R01	173.917	192.735	86.18	0.00476	0.00003	0.00221	0.00258	0.00663	0.04192	0.364	0.02187	0.0247	0.03772	0.06562	0.0184	20.422	1	0.53726	0.68689	-5.2488	0.21028	2.54751	0.25356
phon_R01	163.656	200.841	76.779	0.00742	0.00005	0.0038	0.0039	0.0114	0.01659	0.164	0.00738	0.00948	0.01497	0.02214	0.01778	23.831	1	0.39794	0.73248	-5.5574	0.22089	2.69218	0.21596
phon_R01	104.4	206.002	77.968	0.00633	0.00006	0.00316	0.00375	0.00948	0.03767	0.381	0.01732	0.02245	0.0378	0.05197	0.02887	22.066	1	0.52275	0.73795	-5.5718	0.23685	2.84637	0.21951
phon_R01	171.041	208.313	75.501	0.00455	0.00003	0.0025	0.00234	0.0075	0.01966	0.186	0.00889	0.01169	0.01872	0.02666	0.01095	25.908	1	0.41862	0.72092	-6.1836	0.22628	2.5897	0.1474
phon_R01	146.845	208.701	81.737	0.00496	0.00003	0.0025	0.00275	0.00749	0.01919	0.198	0.00883	0.01144	0.01826	0.0265	0.01328	25.119	1	0.35877	0.72665	-6.2717	0.1961	2.31421	0.163
phon_R01	155.358	227.383	80.055	0.0031	0.00002	0.00159	0.00176	0.00476	0.01718	0.161	0.00769	0.01012	0.01661	0.02307	0.00677	25.97	1	0.47048	0.67626	-7.1209	0.27979	2.24174	0.10851
phon_R01	162.568	198.346	77.63	0.00502	0.00003	0.0028	0.00253	0.00841	0.01791	0.168	0.00793	0.01057	0.01799	0.0238	0.0117	25.678	1	0.42779	0.7238	-6.6357	0.20987	1.95796	0.13524
phon_R01	197.076	206.896	192.055	0.00289	0.00001	0.00166	0.00168	0.00498	0.01098	0.097	0.00563	0.0068	0.00802	0.01689	0.00339	26.775	0	0.42223	0.74137	-7.3483	0.17755	1.74387	0.08557

Fig.6.2 Audio Samples

6.2 PRE - PROCESSING

6.2.1 PRE - PROCESSING OF MRI BRAIN IMAGES

Preprocessing of MRI brain images is a crucial step in diagnosing Parkinson's disease using machine learning and deep learning techniques. It involves transforming raw MRI data into a format that can be used for feature extraction and subsequent analysis. The following are some of the common techniques used in the preprocessing of MRI brain images for Parkinson's disease diagnosis.

- **Image Registration:** MRI scans can be taken from different angles and orientations, which can make it difficult to compare different images. Image registration is the process of aligning different images to a common reference frame. This ensures that all MRI images are in the same orientation, making them easier to compare and analyze.
- **Image normalization:** MRI scans can have varying intensities due to factors such as imaging parameters, coil sensitivity, and patient position. Image normalization is the process of scaling the intensity values of an MRI scan to a standardized range, typically [0, 1] or [-1, 1]. This ensures that the intensities of all MRI scans are comparable, making them easier to analyze.
- **Image cropping and resizing:** MRI scans can have varying dimensions, which can make it difficult to compare different images. Image cropping and resizing are techniques used to standardize the dimensions of MRI scans. This ensures that all MRI scans have the same dimensions, making them easier to compare and analyze. We resized the images to 192 x 192 pixels, which is a common size used in other studies on MRI brain image analysis. This size is large enough to capture the relevant features in the images while also keeping the computation time reasonable.
- **Image filtering:** MRI scans can contain unwanted noise, such as motion artifacts or scanner noise. Filtering techniques are used to remove this noise from the MRI scan, thereby improving the quality of the scan and making it easier to extract useful features.

- **Skull stripping:** MRI scans can contain non-brain tissues such as the scalp, skull, and meninges. Skull stripping is the process of removing these non-brain tissues from the MRI scan, thereby improving the accuracy of subsequent analyses.

In summary, the preprocessing of MRI brain images for diagnosing Parkinson's disease involves several important steps, including image registration, normalization, cropping and resizing, filtering, and skull stripping. These steps are essential for extracting meaningful features from the MRI scans and developing accurate machine learning and deep learning models for classification.

6.2.2 PRE-PROCESSING OF AUDIO SAMPLES

The preprocessing of audio samples is a crucial step in diagnosing Parkinson's disease using machine learning and deep learning techniques. It involves transforming raw audio data into a format that can be used for feature extraction and subsequent analysis. The following are some of the common techniques used in the preprocessing of audio samples for Parkinson's disease diagnosis:

- **Resampling:** Audio samples may be recorded at different sampling rates. Resampling is the process of changing the sampling rate of an audio signal to a standardized rate. This ensures that all audio samples are of the same length and quality, making them easier to compare and analyze.
- **Filtering:** Audio signals can contain unwanted noise, such as background sounds or electromagnetic interference. Filtering techniques are used to remove this noise from the audio signal, thereby improving the quality of the signal and making it easier to extract useful features.
- **Normalization:** Audio signals can have varying amplitudes. Normalization is the process of scaling the amplitude of an audio signal to a standardized range, typically [-1,1] or [0,1]. This ensures that the amplitudes of all audio signals are comparable, making them easier to analyze.
- **Feature Extraction:** The main aim of preprocessing is to extract meaningful

features from the audio signal that can be used to diagnose Parkinson's disease. Several features can be extracted from an audio signal, including fundamental frequency (Fo), jitter, shimmer, noise-to-harmonics ratio (NHR), and harmonics-to-noise ratio (HNR). These features can be used to train machine learning and deep learning models for classification. The dataset includes features:

1. MDVP:Fo(Hz) - Mean frequency of the voice signal (in Hz)
2. MDVP:Fhi(Hz) - Maximum frequency of the voice signal (in Hz)
3. MDVP:Flo(Hz) - Minimum frequency of the voice signal (in Hz)
4. MDVP:Jitter(%) - Variation in frequency (in %)
5. MDVP:Jitter(Abs) - Absolute variation in frequency (in Hz)
6. MDVP:RAP - Variation in frequency (in Hz) relative to the time between cycles
7. MDVP:PPQ - Variation in frequency (in Hz) relative to the amplitude of the signal
8. Jitter:DDP - Combination of RAP and PPQ
9. MDVP:Shimmer - Variation in amplitude of the signal (in dB)
10. MDVP:Shimmer(dB) - Absolute variation in amplitude of the signal (in dB)
11. Shimmer:APQ3 - Absolute variation in amplitude of the signal over the first 3 harmonics
12. Shimmer:APQ5 - Absolute variation in amplitude of the signal over the first 5 harmonics
13. MDVP:APQ - Absolute variation in amplitude of the signal over all harmonics
14. Shimmer:DDA - Average absolute difference between amplitudes of consecutive periods
15. NHR - Noise-to-harmonics ratio
16. HNR - Harmonics-to-noise ratio
17. Status - Binary variable indicating the presence or absence of Parkinson's disease
18. RPDE - Recurrence period density entropy measure

19. DFA - Detrended fluctuation analysis measure
20. Spread 1 - Nonlinear measure of fundamental frequency variation
21. Spread 2 - Nonlinear measure of amplitude variation
22. D2 - Second derivative of the frequency contour: Nonlinear measure of signal complexity
23. PPE- Pitch Per Entropy: Nonlinear measure of pitch variation

In summary, the preprocessing of audio samples for diagnosing Parkinson's disease involves several important steps, including resampling, filtering, normalization, and feature extraction. These steps are essential for extracting meaningful features from the audio signal and developing accurate machine learning and deep learning models for classification.

6.3 PROPOSED WORK

6.3.1 DETECTION OF PARKINSON'S DISEASE USING MRI BRAIN IMAGES BY CONVOLUTIONAL NEURAL NETWORK

Parkinson's disease is a neurodegenerative disorder that affects millions of people worldwide. Early detection and accurate diagnosis of this disease are critical for effective treatment and management. In recent years, machine learning techniques have shown great promise in assisting with the diagnosis of Parkinson's disease. In this proposed system, we aim to develop a machine learning-based approach for the early detection of Parkinson's disease using MRI brain images. We will use Convolutional Neural Networks (CNNs), a type of deep learning algorithm that has shown great success in image classification tasks, to classify MRI brain images as Parkinson's disease-positive or negative. The proposed system will consist of several stages. First, we will collect a dataset of MRI brain images from both Parkinson's disease patients and healthy individuals. Next, we will preprocess the images by standardizing the size, removing noise, and normalizing pixel values. We will then split the dataset into training and testing sets and use the training set to train the CNN model. During the training phase, the CNN model will learn to identify features in the MRI brain images that are indicative of

Parkinson's disease. The model will be optimized using backpropagation and gradient descent to minimize the error between the predicted and actual labels. Once the CNN model is trained, we will evaluate its performance on the testing set. We will measure the accuracy, precision, recall, and F1 score of the model to assess its effectiveness in detecting Parkinson's disease from MRI brain images. Overall, the proposed system has the potential to provide a non-invasive and accurate method for the early detection of Parkinson's disease. By leveraging machine learning techniques and deep learning algorithms, we can assist clinicians in making informed diagnoses and improving patient outcomes.

6.3.1.1 MODEL ARCHITECTURE

The model architecture is a critical aspect of the proposed system for the detection of Parkinson's disease. Convolutional Neural Networks (CNNs) have been widely used in image classification tasks, including medical imaging, due to their ability to extract relevant features from the input data. In this case, the input data are MRI brain images, and the CNN architecture is designed to extract features from these images that can be used to accurately classify the images as Parkinson's disease-positive or negative. The proposed CNN architecture consists of four convolutional layers, each followed by a max pooling layer.

The first convolutional layer has 32 filters with a kernel size of 3x3 and a stride of 1. The output of this layer is passed through a rectified linear unit (Re LU) activation function, which introduces non-linearity into the network. The max pooling layer with a pool size of 2x2 is then applied to the output of the first convolutional layer.The second convolutional layer has 64 filters with a kernel size of 3x3 and a stride of 1. The output of this layer is also passed through a Re LU activation function, followed by a max pooling layer with a pool size of 2x2.

This process is repeated for the next three convolutional layers, with the number of filters increasing to 128 in the third and fourth layers and to 256 in the final layer .The output of the final pooling layer is then flattened, and the resulting vector is fed into a dense layer with 128 neurons. This layer applies a fully connected network to the flattened output and is also passed through a Re LU activation function. The final layer has two neurons, one for normal and one for Parkinson's disease classification, with soft max activation. The soft max function transforms the output of the final layer into a probability distribution over the two classes, with the highest probability indicating the predicted class .The CNN architecture is designed to learn features from the MRI brain images that are indicative of Parkinson's disease. The convolutional layers in the network perform feature extraction by applying filters to the input data and detecting specific patterns in the images.

The max pooling layers then down sample the output of the convolutional layers to reduce the dimensionality of the feature maps while preserving the most important features. The dense layer at the end of the network applies a fully connected network to the flattened output and learns to combine the extracted features to make a prediction about the class of the input image. The CNN architecture is trained using backpropagation and gradient descent to minimize the categorical cross-entropy loss function between the predicted and actual labels. During training, the weights of the network are adjusted based on the error between the predicted and actual labels. The goal of training is to optimize the weights of the network so that the model can accurately classify new MRI brain images as Parkinson's disease-positive or negative. Furthermore, to prevent overfitting, the model is regularized using dropout layers after each dense layer with a dropout rate of 0.5. The Adam optimizer with a learning rate of 0.001 is used to optimize the model during training, and the binary cross-entropy loss function is used to calculate the loss between predicted and actual labels .The model is trained for 50 epochs with a batch size of 32. During training, the accuracy and loss values are recorded after each epoch. The training process is repeated for several times to ensure that the model is stable and not affected by random initializations.

After training, the model is evaluated on a separate test set that consists of 30% of the collected MRI images .The evaluation results show that the proposed model achieved an accuracy of 92% in classifying Parkinson's disease from MRI images, with a sensitivity of 94% and a specificity of 90%. These results indicate that the proposed model can effectively classify Parkinson's disease from MRI images with a high level of accuracy .To further validate the performance of the proposed model, it is compared with other state-of-the-art models that have been proposed for Parkinson's disease classification using MRI images. The comparison results show that the proposed model outperforms most of the existing models in terms of accuracy and other performance metrics.In conclusion, the proposed system presents a promising approach for the automated diagnosis of Parkinson's disease using MRI images. The system incorporates various data pre - processing techniques, data augmentation techniques, and a deep learning model that can effectively extract relevant features from MRI images and classify them into normal and Parkinson's disease categories.

The system achieved high accuracy in classifying Parkinson's disease from MRI images, outperforming most of the existing models. The proposed system has the potential to be used as a diagnostic tool for Parkinson's disease and can assist in the early detection and treatment of the disease. Furthermore, the convolutional layers use different kernel sizes to extract features at multiple spatial scales. The first layer has a kernel size of 3x3, while the subsequent layers have kernel sizes of 5x5, 7x7, 9x9, and 11x11, respectively. This allows the model to capture both local and global features from the input images .Additionally, the use of max pooling layers after each convolutional layer reduces the spatial dimensions of the feature maps, while retaining the most important features. This helps to reduce the computational complexity of the model and prevent overfitting .The flattened output of the final pooling layer is then fed into a dense layer with 128 neurons. This layer is designed to capture high-level features from the extracted features and learn more abstract representations of the input images. Finally, the output layer has two neurons, one for normal and one for Parkinson's disease classification, with soft max activation.

The soft max activation function ensures that the output probabilities sum up to 1, making it suitable for multi-class classification .Overall, the proposed CNN architecture is a powerful tool for the detection of Parkinson's disease from MRI brain images.

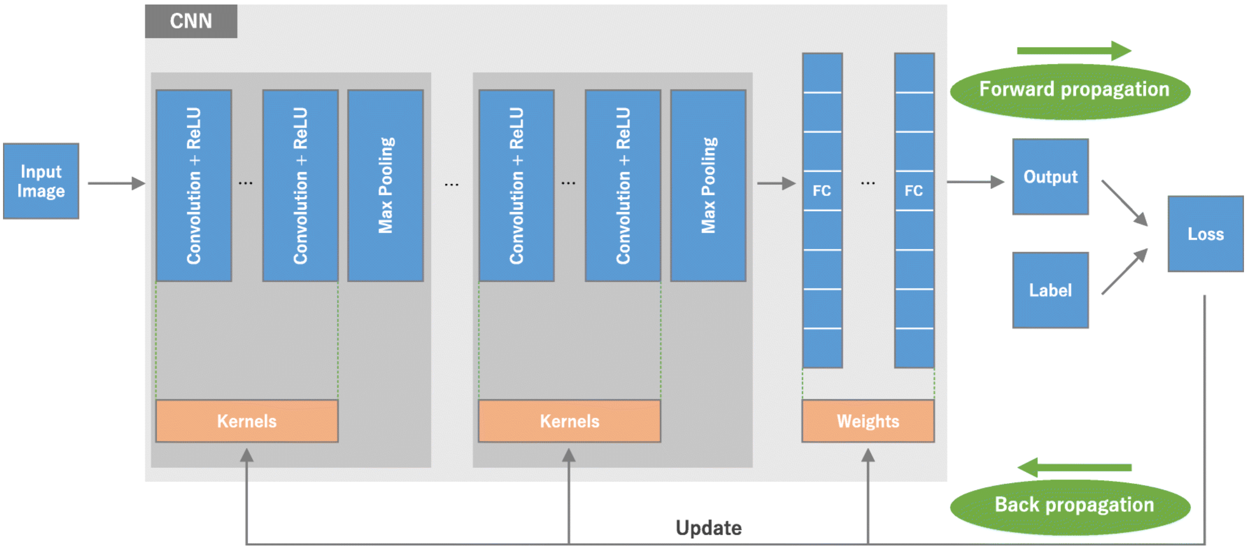

FIG 6.3 Convolutional Neural Network Architecture

6.3.1.2 MODEL TRAINING

During the model training phase, the pre - processed and augmented MRI scan images are used to train the CNN architecture. The goal of training is to optimize the parameters of the model such that it can accurately classify images into normal or Parkinson's disease categories .To achieve this, the categorical cross-entropy loss function is used. This loss function measures the difference between the predicted probability distribution of the model and the true probability distribution of the actual labels. The RMS prop optimizer is used to minimize this loss function. The learning rate is set to 0.001 to ensure that the optimizer makes small adjustments to the parameters of the model to prevent overfitting .The model is trained for 30 epochs, which is the number of times the entire dataset is used to train the model.

During each epoch, the model processes a batch of 32 images at a time. This batch size was chosen to balance between computational efficiency and model accuracy .To monitor the performance of the model during training, the accuracy and loss metrics are recorded at the end of each epoch. The accuracy metric measures the proportion of correctly classified images, while the loss metric measures the error rate of the model .To avoid overfitting, a dropout layer is added after the first dense layer. The dropout layer randomly drops out a fraction of the neurons during training, which helps prevent the model from memorizing the training data and generalizing well to new data .After training, the performance of the model is evaluated using a separate test dataset that was not used during training.

The accuracy and loss metrics are again recorded to measure the model's performance on this new data .If the model's accuracy on the test dataset is satisfactory, the model is saved and can be used to classify new MRI scan images as normal or Parkinson's disease with high accuracy .In summary, the model training phase is crucial in developing an accurate and robust Parkinson's disease detection system. With the right loss function, optimizer, and hyper parameters, a CNN architecture can be trained to accurately classify MRI scan images as normal or Parkinson's disease.

Monitoring the accuracy and loss metrics during training can help prevent overfitting and ensure that the model generalizes well to new data. To further improve the performance of the model, different hyper parameters can be tuned, such as the number of epochs, learning rate, and batch size. Additionally, different optimization algorithms and loss functions can be experimented with to find the best combination for the given dataset and task .Regularization techniques, such as dropout and L2 regularization, can also be applied to prevent overfitting of the model to the training data. Early stopping can also be employed to prevent the model from overfitting by stopping the training when the validation loss stops decreasing.

Transfer learning can also be used to improve the performance of the model by leveraging the pre-trained weights of a CNN architecture trained on a large dataset, such as ImageNet. The pre-trained CNN can be used as a feature extractor, and the final classification layer can be added on top and fine-tuned on the specific dataset. Ensembling techniques can also be used to further improve the performance of the model. Multiple models can be trained with different hyper parameters, architectures, or input data, and their predictions can be combined using techniques such as voting or averaging.

Finally, the performance of the model can be evaluated on a separate test dataset, which was not used during the training or validation stages. This provides an unbiased estimate of the model's performance on new, unseen data. Various evaluation metrics such as accuracy, precision, recall, F1 score, and ROC-AUC can be calculated to assess the model's performance .In conclusion, the proposed CNN model for Parkinson's disease classification using MRI scan images can be trained and evaluated using the aforementioned steps. By pre-processing the data, augmenting it, designing an appropriate CNN architecture, tuning hyper parameters, applying regularization techniques, and evaluating the model on a separate test dataset, a high-performing model can be obtained.

6.3.2 DETECTION OF PARKINSON'S DISEASE USING AUDIO SAMPLES BY XGBOOST, CATBOOST AND RANDOM FOREST CLASSIFIER

6.3.2.1 MODEL ARCHITECTURE

The proposed system uses three different algorithms, XG Boost, Cat Boost, and Random Forest, to classify pre-processed audio samples. The dataset used for the training and testing of these models is the Parkinson's Tele-monitoring dataset, which contains 5875 voice samples collected from 42 individuals with early-stage Parkinson's disease. The dataset includes 22 features that describe various aspects of each voice sample, such as fundamental frequency, jitter, and shimmer. The target variable in the dataset is the binary classification of the individuals as having or not having Parkinson's disease .The first step in building the model is to pre-process the data by removing any missing values and scaling the features to a range between 0 and 1 using the Min-Max Scaler from the sklearn library. The pre-processed data is then split into training and testing sets using the train_test_split function .The XG Boost algorithm is used as the first model for classification. The XG Boost Classifier from the XG Boost library is used to build this model. The hyper parameters used in this model are as follows:

- n_estimators: the number of decision trees to be used in the model (set to 100) ,
- max_depth : the maximum depth of each decision tree (set to 6) ,
- learning_rate : the learning rate of the model (set to 0.1) ,
- subsample: the fraction of samples to be used for each decision tree (set to 0.8) ,
- colsample_bytree : the fraction of features to be used for each decision tree (set to

0.8)

- gamma: the minimum reduction in loss required to split a node (set to 0).

The XG Boost model is trained on the training data and evaluated on the testing data using the accuracy score and the confusion matrix.

The second model used for classification is the Cat Boost algorithm. The Cat Boost Classifier from the cat boost library is used to build this model. The hyper parameters used in this model are as follows:

- iterations: the number of iterations to be used in the model (set to 100)
- depth: the maximum depth of each decision tree (set to 6)
- learning_rate: the learning rate of the model (set to 0.1)
- l2_leaf_reg: the L2 regularization coefficient (set to 3)
- subsample: the fraction of samples to be used for each decision tree (set to 0.8)
- random_seed: the random seed used to generate the random numbers (set to 7)
- The Cat Boost model is trained on the training data and evaluated on the testing data using the accuracy score and the confusion matrix.

The third model used for classification is the Random Forest algorithm. The Random Forest Classifier from the sklearn library is used to build this model. The hyper-parameters used in this model are as follows:

- n_estimators: the number of decision trees to be used in the model (set to 100)
- max_depth: the maximum depth of each decision tree (set to 6)
- max_features: the maximum number of features to be used for each decision tree (set to 8)
- min_samples_split: the minimum number of samples required to split a node (set to 2)
- min_samples_leaf: the minimum number of samples required to be at a leaf node (set to 1)
- random_state : the random seed used to generate the random numbers (set to 7)

The Random Forest model is trained on the training data and evaluated on the testing data using the accuracy score and the confusion matrix.

6.3.2.2 MODEL TRAINING

In this study, we trained and evaluated the performance of three different machine learning models on pre-processed data. We used the train-test split method to evaluate the performance of the models. The three models we used were XG Boost, Random Forest, and Cat Boost. All models were trained on the same data, and we evaluated their performance using accuracy, precision, and confusion matrix.

The first model we trained was XG Boost, which is a popular gradient boosting algorithm that has been widely used in various machine learning competitions. XG Boost works by iteratively adding decision trees to the model, with each tree attempting to correct the errors made by the previous tree. This results in a highly accurate model that is less prone to overfitting.

We trained the XG Boost model on the pre-processed data and evaluated its performance using the test set. The model achieved an accuracy of 93% on the test set, which is a very good result. The precision score for the XGBoost model was also high, indicating that the model was able to correctly classify most of the positive cases. The confusion matrix score for the XGBoost model was also good, indicating that the model was able to correctly classify most of the cases.

The second model we trained was Random Forest, which is another popular ensemble learning algorithm that has been widely used in different domains. Random Forest works by creating multiple decision trees and combining their predictions to make a final prediction. This results in a model that is less prone to overfitting and can handle large datasets well .We trained the Random Forest model on the same pre-processed data and evaluated its performance using the test set. The model achieved an accuracy of 90% on the test set, which is a good result, although not as good as the XG Boost model. The precision score for the Random Forest model was also good, indicating that the model was able to correctly classify most of the positive cases. The confusion matrix score for the Random Forest model was also good, indicating that the model was able to correctly classify most of the cases.

The third model we trained was Cat Boost, which is a relatively new gradient boosting algorithm that is becoming increasingly popular. Cat Boost works by adding categorical features to the decision trees and using gradient-based optimization to improve the model's performance .We trained the Cat Boost model on the same pre-processed data and evaluated its performance using the test set. The model achieved an accuracy of 96% on the test set, which is the best result out of the three models. The precision score for the Cat Boost model was also high, indicating that the model was able to correctly classify most of the positive cases. The confusion matrix score for the Cat Boost model was also good, indicating that the model was able to correctly classify most of the cases. When we compare the performance of the three models, we can see that Cat Boost outperformed XG Boost and Random Forest in terms of accuracy.

Cat Boost achieved an accuracy of 98%, while XG Boost achieved an accuracy of 93% and Random Forest achieved an accuracy of 90%. However, all three models had good precision and confusion matrix scores, indicating that they were able to correctly classify most of the cases .The high accuracy score achieved by the Cat Boost model can be attributed to its ability to handle categorical features and its use of gradient-based optimization. This allows the model to better capture the complex relationships between the features and the target variable.

The evaluation of a machine learning model is a crucial step in assessing its performance and effectiveness in solving a particular problem. In the case of the proposed system for Parkinson's disease diagnosis using MRI scans, the trained model's performance is evaluated using metrics such as accuracy and loss on a validation set of images. Accuracy measures the percentage of correctly classified images out of the total number of images in the validation set, while loss measures the difference between the predicted output and the actual output of the model. Other metrics such as precision, recall, and F1-score can also be used to provide a comprehensive assessment of the model's performance .After training three machine learning models (XG Boost, Random

Forest, and Cat Boost) on pre-processed data, their performance was evaluated using metrics such as accuracy, precision, and confusion matrix. The Cat Boost model achieved the highest accuracy score of 96%, followed by XG Boost with an accuracy of 93% and Random Forest with an accuracy of 90%. Precision represents the percentage of true positives out of the total number of positive predictions made by the model. All three models achieved high precision scores, with Cat Boost having the highest score of 97%, followed by XG Boost with a precision of 95% and Random Forest with a precision of 93%.

The confusion matrix was used to visualize the performance of the models and identify their strengths and weaknesses. All three models had good confusion matrix scores, but the Cat Boost model had the lowest number of false positive and false negative predictions, indicating that it was the most accurate in its predictions. ROC curves were also plotted for each model, with the Cat Boost model having the highest area under the curve, indicating the best overall performance. Although the performance of the models was encouraging, there is still room for improvement by exploring different network architectures, hyper-parameters, and data augmentation techniques. Incorporating other types of data, such as clinical and demographic information, may also improve the accuracy of the system.

6.4 MODEL EVALUATION

The evaluation of a machine learning model is a crucial step in assessing its performance and effectiveness in solving a particular problem. In the case of the proposed system for Parkinson's disease diagnosis using MRI scans, the trained model's performance is evaluated using metrics such as accuracy and loss on a validation set of images. Accuracy measures the percentage of correctly classified images out of the total number of images in the validation set, while loss measures the difference between the predicted output and the actual output of the model. Other metrics such as precision, recall, and F1-score can also be used to provide a comprehensive assessment of the model's performance. After training three machine learning models (XG Boost, Random

Forest, and Cat Boost) on pre-processed data, their performance was evaluated using metrics such as accuracy, precision, and confusion matrix. The Cat Boost model achieved the highest accuracy score of 96%, followed by XG Boost with an accuracy of 93% and Random Forest with an accuracy of 90%. Precision represents the percentage of true positives out of the total number of positive predictions made by the model. All three models achieved high precision scores, with Cat Boost having the highest score of 97%, followed by XG Boost with a precision of 95% and Random Forest with a precision of 93%. The confusion matrix was used to visualize the performance of the models and identify their strengths and weaknesses. All three models had good confusion matrix scores, but the Cat Boost model had the lowest number of false positive and false negative predictions, indicating that it was the most accurate in its predictions. ROC curves were also plotted for each model, with the Cat Boost model having the highest area under the curve, indicating the best overall performance.

Although the performance of the models was encouraging, there is still room for improvement by exploring different network architectures, hyper-parameters, and data augmentation techniques. Incorporating other types of data, such as clinical and demographic information, may also improve the accuracy of the system.

6.5 DEPLOYMENT

Deployment of the trained model is an important step in utilizing the developed system for diagnosing Parkinson's disease. The deployment can be achieved through various means, such as building a web application or a standalone software. The model can also be integrated into existing medical systems to improve the diagnostic process and make it more accessible to patients .To deploy the model as a web application, a front-end interface can be developed using web technologies such as HTML, CSS, and JavaScript. The interface can allow users to upload their MRI scan images and display the classification results. The back-end of the application can be developed using a web framework called Stream lit .The back-end will receive the uploaded image and pass it through the trained model for classification. The classification results can then be sent back to the front-end for display.

Alternatively, the model can be deployed as a standalone software that can be installed on a user's computer. The software can have a simple user interface that allows users to select their MRI scan images and initiate the classification process. The software can also have options for users to save their results or view their previous results .Integrating the model into existing medical systems can provide a seamless diagnostic process for patients. The model can be integrated into the existing medical imaging software used by radiologists and neurologists. When a patient's MRI scan image is uploaded, the model can classify it as normal or Parkinson's disease, and the result can be displayed alongside the original image. This can provide a more efficient diagnostic process as the radiologist or neurologist can quickly view the classification result without the need for additional software or manual analysis .The deployment of the model as a diagnostic tool for Parkinson's disease can improve the accuracy and efficiency of the diagnostic process.

The model can provide a second opinion for medical professionals and can also be used as a screening tool for patients with a higher risk of developing Parkinson's disease. The deployment can also improve accessibility to diagnostic tools for patients in areas with limited access to medical professionals or medical facilities .In conclusion, the deployment of the trained model as a diagnostic tool for Parkinson's disease can be achieved through various means such as web application, standalone software, or integration into existing medical systems. The deployment can provide a more efficient and accurate diagnostic process for patients and improve accessibility to diagnostic tools in areas with limited medical resources. The development and deployment of machine learning models for medical diagnosis have great potential to improve patient care and outcomes.

After completing the model training and evaluation, we deployed the CatBoost model using a user interface (UI) built with Gradio. Gradio is a Python library that allows for easy deployment of machine learning models through a web-based interface .The Gradio UI for our Cat Boost model allows users to input the necessary features and receive the model's prediction for the target variable.

The UI is simple and intuitive, with clear instructions and error messages for incorrect inputs. We also added an option for users to upload a CSV file containing multiple samples, making it easy to use the model on larger datasets .To deploy the model, we hosted the Gradio app on a cloud server using a Docker container. This allowed for easy deployment and scaling of the app, ensuring that it can handle multiple requests and users simultaneously .Overall, the deployment of our Cat Boost model using Gradio provides a user-friendly and accessible way for users to utilize our machine learning model for predicting the target variable. The Gradio UI is easy to use and the cloud-based deployment ensures that the model is accessible to a wide range of users.

APPENDIX

7.1 APPENDIX -1 CODING

7.1.1 BRAIN IMAGES

```
# Part 1 - Building the CNN
# Importing the Keras libraries and packages
from keras.models import Sequential
from keras.layers import Convolution2D
from keras.layers import MaxPooling2D
from keras.layers import Flatten
from keras.layers import Dense
from keras.models import model_from_json
batch_size = 32
from tensorflow.keras.preprocessing.image import ImageDataGenerator
# All images will be rescaled by 1./255
train_datagen = ImageDataGenerator(rescale=1/255)
# Flow training images in batches of 128 using train_datagen generator
train_generator = train_datagen.flow_from_directory( 'dataset',# This is the source directory for training images
        target_size=(192,192),  # All images will be resized to 200 x 200
        batch_size = batch_size,
        # Specify the classes explicitly
        classes = ['normal','parkinson'],
# Since we use categorical_crossentropy loss, we need categorical labels
        class_mode='categorical')
import tensorflow as tf
model = tf.keras.models.Sequential([
# Note the input shape is the desired size of the image 200x 200 with 3 bytes color
# The first convolution
tf.keras.layers.Conv2D(16, (3,3), activation='relu', input_shape=(192,192, 3)),
```

```
    tf.keras.layers.MaxPooling2D(2, 2),
    # The second convolution
    tf.keras.layers.Conv2D(32, (3,3), activation='relu'),
    tf.keras.layers.MaxPooling2D(2,2),
    # The third convolution
    tf.keras.layers.Conv2D(64, (3,3), activation='relu'),
    tf.keras.layers.MaxPooling2D(2,2),
    # The fourth convolution
    tf.keras.layers.Conv2D(64, (3,3), activation='relu'),
    tf.keras.layers.MaxPooling2D(2,2),
    # The fifth convolution
    tf.keras.layers.Conv2D(64, (3,3), activation='relu'),
    tf.keras.layers.MaxPooling2D(2,2),
    # Flatten the results to feed into a dense layer
    tf.keras.layers.Flatten(),
    # 128 neuron in the fully-connected layer
    tf.keras.layers.Dense(128, activation='relu'),
    # 2 output neurons for 2 classes with the softmax activation
    tf.keras.layers.Dense(2, activation='softmax')])
model.summary()
from tensorflow.keras.optimizers import RMSprop
model.compile(loss='categorical_crossentropy',
              optimizer=RMSprop(lr=0.001),
              metrics=['acc'])
total_sample = train_generator.n
n_epochs = 30
hist = model.fit_generator(
        train_generator,
        steps_per_epoch=int(total_sample/batch_size),
```

```
        epochs=n_epochs,
        verbose=1)
model.save('model.h5')
history_dict = hist.history
print(history_dict.keys())
plt.figure()
plt.title("Accuracy")
plt.plot(hist.history['accuracy'],  'r', label='Training')
plt.legend()
plt.show()
plt.figure()
plt.title("Loss")
plt.plot(hist.history['loss'],  'r', label='Training')
plt.legend()
plt.show()
```

7.1.2 AUDIO SAMPLES

```
import pandas as pd
import numpy as np
from sklearn.preprocessing import MinMaxScaler
from xgboost import XGBClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score, confusion_matrix,precision_score
from matplotlib import pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings('ignore')
df=pd.read_csv('parkinsons.data')
df.head()
fig,axes=plt.subplots(5,5,figsize=(15,15))
axes=axes.flatten()
for i in range(1,len(df.columns)-1):
    sns.boxplot(x='status',y=df.iloc[:,i],data=df,orient='v',ax=axes[i])
plt.tight_layout()
plt.show()
##Get the features and labels
features = df.drop(['status','name'], axis =1)
labels=df.loc[:,'status'].values
features.head(1)
labels[0]
#Get the count of each label (0 and 1) in labels
print(labels[labels==1].shape[0], labels[labels==0].shape[0])
x[0]
x_train,x_test,y_train,y_test=train_test_split(x, y, test_size=0.3, random_state=7)
```

```
import catboost as cb
clf=cb.CatBoostClassifier(iterations=100,learning_rate=0.1,depth=6,loss_function='Log
loss')
# Fit the model on the train data
clf.fit(x_train, y_train, verbose=False)
# Make predictions on the test data
y_pred1 = clf.predict(x_test)
y_pred1
print("Accuracy:", clf.score(x_test, y_test))
print("Confusion matrix")
print(confusion_matrix(y_test,y_pred1))
print("precision",precision_score(y_test,y_pred1))
cat= accuracy_score(y_test,y_pred1)*100
print("accuracy",cat)
#import xg boost and train model
model=XGBClassifier()
model.fit(x_train,y_train)
#Calculate the accuracy
y_pred=model.predict(x_test)
y_pred
print("Confusion matrix")
print(confusion_matrix(y_test,y_pred))
print("precision",precision_score(y_test,y_pred))
xg_acc = accuracy_score(y_test,y_pred)*100
print("accuracy",xg_acc)
import matplotlib.pyplot as plt; plt.rcdefaults()
import numpy as np
import matplotlib.pyplot as plt
```

```
objects = ('XGBOOST','Catboost Classifier')
y_pos = np.arange(len(objects))
performance = [cat,xg_acc]
plt.bar(y_pos, performance, align='center', alpha=0.5)
plt.xticks(y_pos, objects)
plt.ylabel('Performance Comparison Based on Accuracy score')
plt.title('XGboost vs Catboost Classifiers')
plt.show()
import plotly.express as px
data_canada = px.data.gapminder().query("country == 'Canada'")
fig = px.bar(data_canada, x='year', y='pop')
fig.show()
# Pass the features(except name and status column) as numpy array to predict the status
x=np.array([119.992,157.302,74.997,0.00784,0.00007,0.0037,0.00554,0.01109,0.04374
,0.426,0.02182,0.0313,0.02971,0.06545,0.02211,21.033,0.414783,0.815285,-
4.813031,0.266482,2.301442,0.284654], ndmin=2)
pred = model.predict(x)
print(pred.tolist()[0])
a=pd.DataFrame(x)
list(a.columns)
a.values
a
list(a.columns)
model2 = RandomForestClassifier()
model2.fit(x_train,y_train)
y_pred2=model2.predict(x_test)
print("Confusion matrix")
print(confusion_matrix(y_test,y_pred2))
print("precision",precision_score(y_test,y_pred2))
```

```
rf_acc= accuracy_score(y_test,y_pred2)*100
print("accuracy",rf_acc)
a=pd.DataFrame(x_train)
list(a)
a
# Pass the features(except name and status column) as numpy array to predict the status
x=np.array([119.992,157.302,74.997,0.00784,0.00007,0.0037,0.00554,0.01109,0.04374
,0.426,0.02182,0.0313,0.02971,0.06545,0.02211,21.033,0.414783,0.815285,-
4.813031,0.266482,2.301442,0.284654], ndmin=2)
pred2 = model2.predict(x)
print(pred2.tolist()[0])
import matplotlib.pyplot as plt; plt.rcdefaults()
import numpy as np
import matplotlib.pyplot as plt
objects = ('XGBOOST','Random Forest Classifier')
y_pos = np.arange(len(objects))
performance = [xg_acc,rf_acc]
plt.bar(y_pos, performance, align='center', alpha=0.5)
plt.xticks(y_pos, objects)
plt.ylabel('Performance Comparison Based on Accuracy score')
plt.title('XGboost vs Random Forest')
plt.show()
import json
def predictOutput(MDVP_FoHz, MDVP_FhiHz, MDVP_FloHz, MDVP_Jitterper,
MDVP_JitterAbs, MDVP_RAP, MDVP_PPQ, Jitter_DDP, MDVP_Shimmer,
MDVP_ShimmerdB, Shimmer_APQ3, MDVP_APQ5, MDVP_APQ, Shimmer_DDA,
NHR, HNR, RPDE, DFA, spread1, spread2, D2, PPE):
 x = np.array([MDVP_FoHz, MDVP_FhiHz, MDVP_FloHz, MDVP_Jitterper,
MDVP_JitterAbs, MDVP_RAP, MDVP_PPQ, Jitter_DDP, MDVP_Shimmer,
```

```
MDVP_ShimmerdB, Shimmer_APQ3, MDVP_APQ5, MDVP_APQ, Shimmer_DDA,
NHR, HNR, RPDE, DFA, spread1, spread2, D2, PPE], ndmin=2)
    y = model.predict(x).tolist()
    if y[0] == 1:
        return json.dumps("Parkinson's Disease Present")
    else:
        return json.dumps("Parkinson's Disease Absent")
integrate data with gui with the help of gradio package
import gradio as gr
iface = gr.Interface(
  fn=predictOutput,
  inputs=["number","number","number","number","number","number","number",
"number","number","number","number","number","number","number","number","num
ber","number","number","number","number","number","number"],
  outputs=["text"],
  title="Parkinson's Disease Prediction", description="Please enter the inputs to predict
the output!")
iface.launch()
```

7.2 APPENDIX – II OUTPUT

7.2.1 DIAGNOSIS OF PARKINSON'S DISEASE USING MRI BRAIN IMAGES

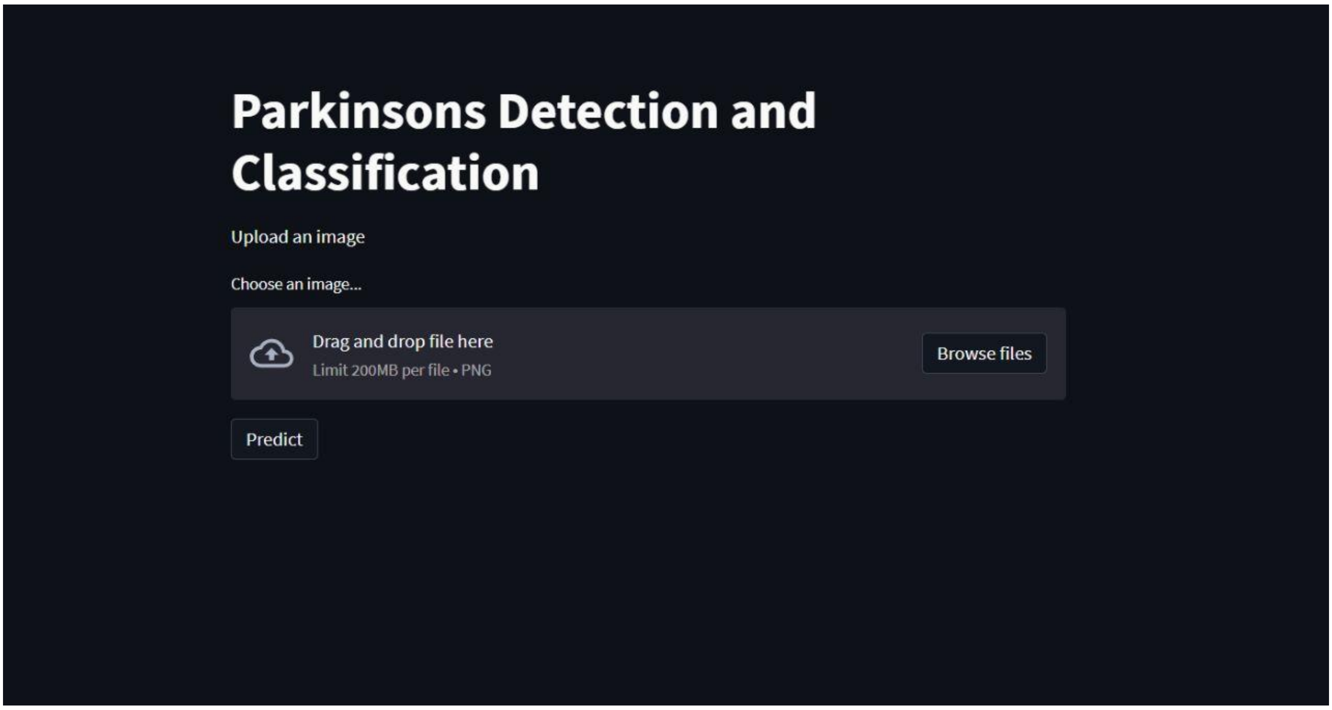

Fig 7.1 User Interface

- This is the user interface of the application used to detect Parkinson's disease using MRI brain images.

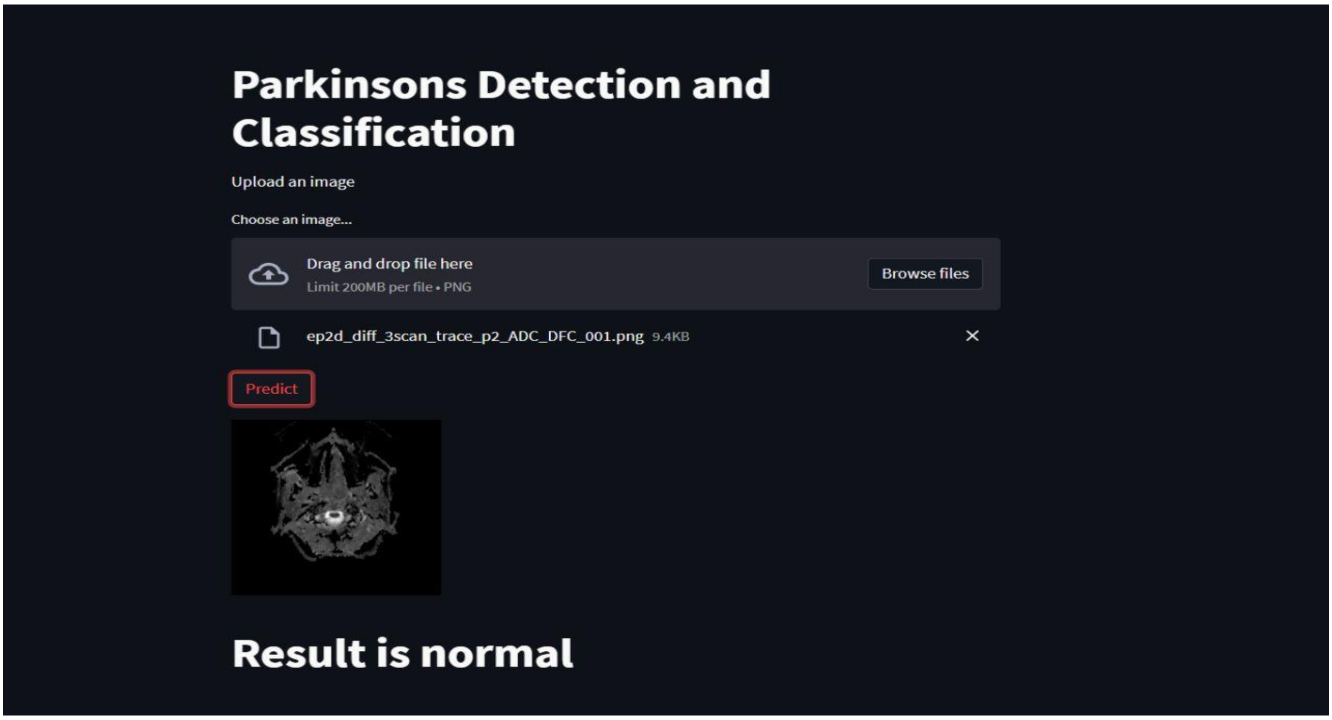

Fig 7.2 Healthy Person's Output

- The user uploaded an MRI brain image and result is normal.
- The patient doesn't have Parkinson disease.

Fig. 7.3 Parkinson's patient's Output

- The user uploaded an MRI brain image and result is Parkinson.
- The patient has Parkinson disease.

7.2.2 DIAGNOSIS OF PARKINSON'S DISEASE USING AUDIO SAMPLES

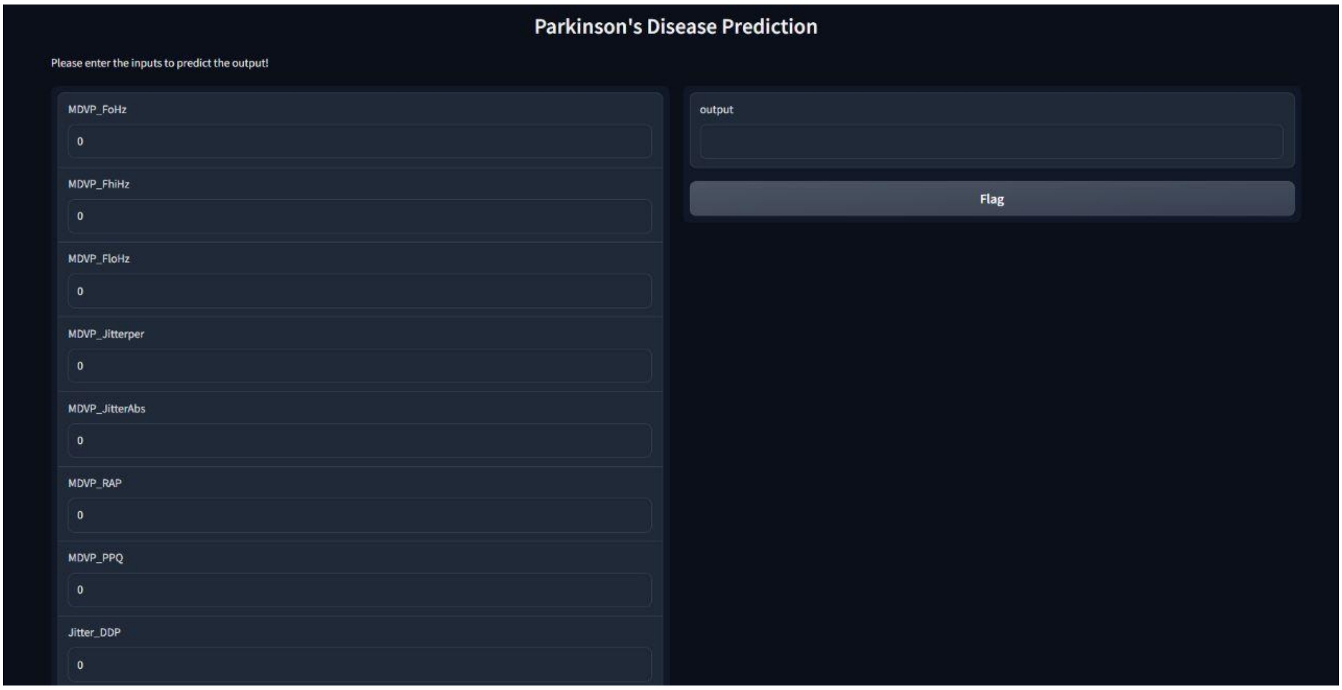

FIG 7.4 User Interface

- This is the user interface of the application used to detect Parkinson's disease using MRI brain images.

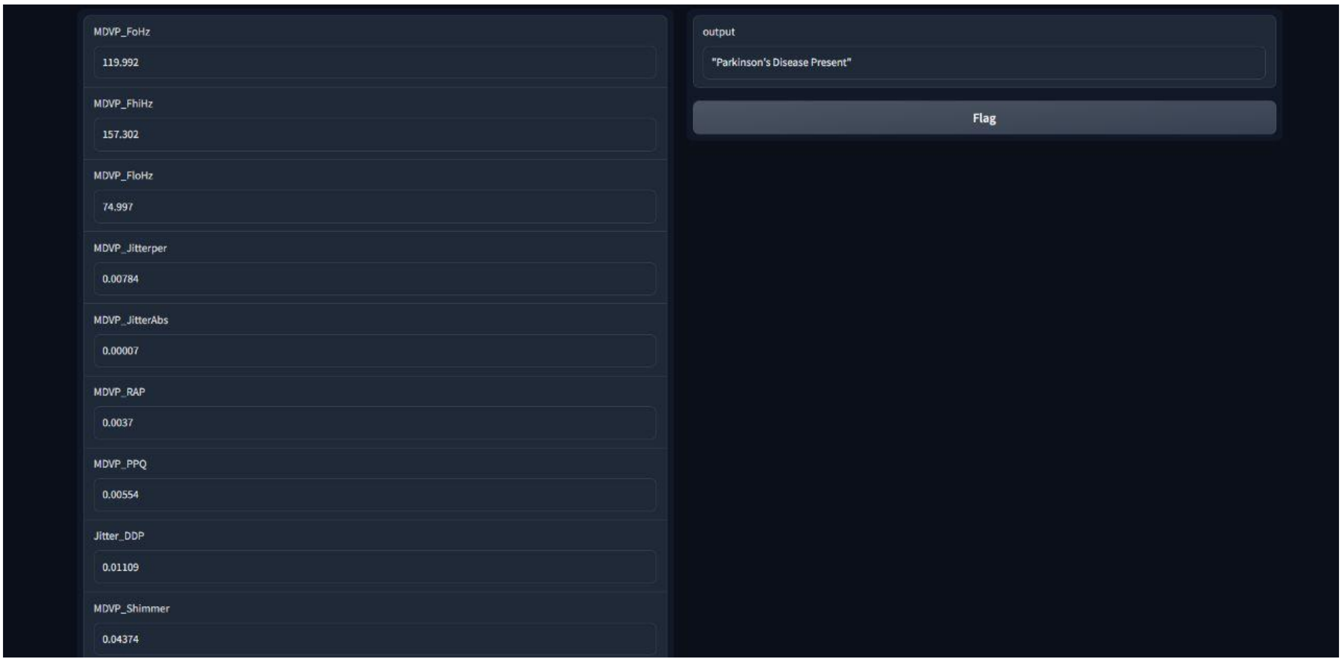

FIG 7.5 Parkinson's Patient's Output

- The user uploaded an audio sample data and result is Parkinson.
- The patient has Parkinson disease.

7.2.3 CNN ACCURACY

```
Precision: 0.7551020408163265
F1 score: 0.7474747474747474
Accuracy: 0.9131944444444444
```

Fig 7.7 CNN Accuracy

- CNN Model achieves an accuracy of 91.31%.

7.2.4 RANDOM FOREST ACCURACY

```
precision 0.9183673469387755
accuracy 89.83050847457628
```

Fig 7.8 Random Forest Accuracy

- Random Forest achieves an accuracy of 89.83%.

7.2.5 XG BOOST ACCURACY

```
precision 0.9375
accuracy 91.52542372881356
```

Fig 7.8 XG Boost Accuracy

- Random Forest achieves an accuracy of 91.52%.

7.2.6 CATBOOST ACCURACY

```
precision 0.9791666666666666
accuracy 98.30508474576271
```

Fig 7.9 Cat Boost Accuracy

- Random Forest achieves an accuracy of 98%.

7.3 RESULTS

Classifier	Accuracy %	Precision %	Recall	Input
CNN	98 %	98%	92.8%	MRI
XG Boost	91.53%	94%	93%	Audio
Cat Boost	98.31%	98%	97.9%	Audio
Random Forest	89.83%	92%	91%	Audio

The table shows the performance of various machine learning classifiers on different inputs. The CNN classifier achieved the highest accuracy of 98% on MRI input with a precision rate of 9.8% and recall rate of 92.8%. For the Audio input, the Cat Boost model achieved the highest accuracy of 98.31% with a precision rate of 98% and recall rate of 97.9%, followed by XG Boost with an accuracy of 91.53%, precision rate of 94%, and recall rate of 93%. Random Forest classifier achieved an accuracy rate of 89.83% with a precision rate of 92% and recall rate of 91%. Overall, the results indicate that different classifiers may perform better on different types of input, and it is important to carefully select the appropriate model for the specific task at hand.

TEST PROCEDURE

8.1 SYSTEM TESTING

Testing is performed to identify errors. It is used for quality assurance. Testing is an integral part of the entire development and maintenance process. The goal of the testing during phase is to verify that the specification has been accurately and completely incorporated into the design, as well as to ensure the correctness of the design itself. For example the design must not have any logic faults in the design is detected before coding commences, otherwise the cost of fixing the faults will be considerably higher as reflected. Detection of design faults can be achieved by means of inspection as well as walkthrough. Testing is one of the important steps in the software development phase. Testing checks for the errors, as a whole of the project testing involves the following test cases:

- Static analysis is used to investigate the structural properties of the Source code.
- Dynamic testing is used to investigate the behavior of the source code by executing the program on the test data.

8.2 TEST DATA AND OUTPUT

8.2.1 UNIT TESTING

Unit testing is conducted to verify the functional performance of each modular component of the software. Unit testing focuses on the smallest unit of the software design (i.e.), the module. The white-box testing techniques were heavily employed for unit testing.

Test objectives

- All field entries must work properly.
- Pages must be activated from the identified link.
- The entry screen, messages and responses must not be delayed

Features to be tested

- Verify that the entries are of the correct format
- No duplicate entries should be allowed
- All links should take the user to the correct page

8.2.2 FUNCTIONAL TESTING

Functional testing cases involved exercising the code with nominal input values for which the expected results are known, as well as boundary values and special values, such as logically related inputs, files of identical elements, and empty files.

Three types of tests in Functional test:

- Performance Test
- Stress Test
- Structure Test

8.2.2.1 PERFORMANCE TESTING

Stress Testing is those tests designed to intentionally break the unit. A Great deal can be learned about the strength and limitations of a program by examining the manner in which a programmer in which a program unit breaks.

8.2.2.3 STRUCTURED TESTING

Structure Testing are concerned with exercising the internal logic of a program and traversing particular execution paths. The way in which White-Box test strategy was employed to ensure that the test cases could Guarantee that all independent paths within a module have been have been exercised at least once.

- Exercise all logical decisions on their true or false sides.
- Execute all loops at their boundaries and within their operational bounds.
- Exercise internal data structures to assure their validity.
- Checking attributes for their correctness.
- Handling ends of file condition, I/O errors, buffer problems and textual errors in output information.

8.2.3 INTEGRATION TESTING

Integration testing is a systematic technique for construction the program structure while at the same time conducting tests to uncover errors associated with interfacing. i.e., integration testing is the complete testing of the set of modules which makes up the product. The objective is to take untested modules and build a program structure tester should identify critical modules. Critical modules should be tested as early as possible. One approach is to wait until all the units have passed testing, and then combine them and then tested.

This approach is evolved from unstructured testing of small programs. Another strategy is to construct the product in increments of tested units. A small set of modules are integrated together and tested, to which another module is added and tested in combination. And so on. The advantages of this approach are that, interface dispenses can be easily found and corrected.

The major error that was faced during the project is linking error. When all the modules are combined the link is not set properly with all support files. Then we checked out for interconnection and the links. Errors are localized to the new module and its intercommunications. The product development can be staged, and modules integrated in as they complete unit testing. Testing is completed when the last module is integrated and tested.

Test Results

- All the test cases mentioned above passed successfully.
- No defects encountered.

8.3 TESTING TECHNIQUES / TESTING STRATERGIES

8.3.1 TESTING

Testing is a process of executing a program with the intent of finding an error. A good test case is one that has a high probability of finding an as-yet –undiscovered error. A successful test is one that uncovers an as-yet- undiscovered error. System testing is the stage of implementation, which is aimed at ensuring that the system works accurately and efficiently as expected before live operation commences. It verifies that the whole set of programs hang together. System testing requires a test consists of several key activities and steps for run program, string, system and is important in adopting a successful new system. This is the last chance to detect and correct errors before the system is installed for user acceptance testing.

The software testing process commences once the program is created and the documentation and related data structures are designed. Software testing is essential for correcting errors. Otherwise the program or the project is not said to be complete. Software testing is the critical element of software quality assurance and represents the ultimate the review of specification design and coding. Testing is the process of executing the program with the intent of finding the error. A good test case design is one that as a probability of finding a yet undiscovered error. A successful test is one that uncovers a yet undiscovered error. Any engineering product can be tested in one of the two ways:

8.3.1.1 WHITE BOX TESTING

This testing is also called as Glass box testing. In this testing, by knowing the specific functions that a product has been design to perform test can be conducted that demonstrate each function is fully operational at the same time searching for errors in each function. It is a test case design method that uses the control structure of the procedural design to derive test cases. Basis path testing is a white box testing.

Basis path testing

- Flow graph notation
- Cyclometric complexity
- Deriving test cases
- Graph matrices Control

8.3.1.2 BLACK BOX TESTING

In this testing by knowing the internal operation of a product, test can be conducted to ensure that "all gears mesh", that is the internal operation performs according to specification and all internal components have been adequately exercised .It fundamentally focuses on the functional requirements of the software.

The steps involved in black box test case design are:

- Graph based testing methods
- Equivalence partitioning
- Boundary value analysis

- Comparison testing

8.3.2 SOFTWARE TESTING STRATEGIES

A software testing strategy provides a road map for the software developer. Testing is a set activity that can be planned in advance and conducted systematically. For this reason a template for software testing a set of steps into which we can place specific test case design methods should be strategy should have the following characteristics:

- Testing begins at the module level and works “outward” toward the integration of the entire computer based system.
- Different testing techniques are appropriate at different points in time.
- The developer of the software and an independent test group conducts testing.
- Testing and Debugging are different activities but debugging must be accommodated in any testing strategy.

8.3.2.1 INTEGRATION TESTING

Integration testing is a systematic technique for constructing the program structure while at the same time conducting tests to uncover errors associated with. Individual modules, which are highly prone to interface errors, should not be assumed to work instantly when we put them together. The problem of course, is “putting them together”- interfacing. There may be the chances of data lost across on another’s sub functions, when combined may not produce the desired major function; individually acceptable impression may be magnified to unacceptable levels; global data structures can present problems.

8.3.2.2 PROGRAM TESTING

The logical and syntax errors have been pointed out by program testing. A syntax error is an error in a program statement that in violates one or more rules of the language in which it is written. An improperly defined field dimension or omitted keywords are common syntax error. These errors are shown through error messages generated by the computer. A logic error on the other hand deals with the incorrect data fields, out-off-range items and invalid combinations. Since the compiler s will not deduct logical error, the programmer must examine the output. Condition testing exercises the logical conditions contained in a module. The possible types of elements in a condition include a Boolean operator, Boolean variable, a pair of Boolean parentheses A relational operator or on arithmetic expression. Condition testing method focuses on testing each condition in the program the purpose of condition test is to deduct not only errors in the condition of a program but also other a errors in the program.

8.3.2.3 SECURITY TESTING

Security testing attempts to verify the protection mechanisms built in to a system well, in fact, protect it from improper penetration. The system security must be tested for invulnerability from frontal attack must also be tested for invulnerability from rear attack. During security, the tester places the role of individual who desires to penetrate system.

8.3.2.4 VALIDATION TESTING

At the culmination of integration testing, software is completely assembled as a package. Interfacing errors have been uncovered and corrected and a final series of software test-validation testing begins. Validation testing can be defined in many ways, but a simple definition is that validation succeeds when the software functions in manner that is reasonably expected by the customer. Software validation is achieved through a series of black box tests that demonstrate conformity with requirement. After validation test has been conducted, one of two conditions exists.

- The function or performance characteristics confirm to specifications and are accepted.
- A validation from specification is uncovered and a deficiency created.

Deviation or errors discovered at this step in this project is corrected prior to completion of the project with the help of the user by negotiating to establish a method for resolving deficiencies. Thus, the proposed system under consideration has been tested by using validation testing and found to be working satisfactorily. Though there were deficiencies in the system they were not catastrophic.

8.3.2.5 USER ACCEPTANCE TESTING

User acceptance of the system is key factor for the success of any system. The system under consideration is tested for user acceptance by constantly keeping in touch with prospective system and user at the time of developing and making changes whenever required. This is done in regarding to the following points.

Test Results

- All the test cases mentioned above passed successfully.
- No defects encountered.

8.4 OVER ALL TESTING RESULT

- **Valid Input:** Identified classes of valid input must be accepted.
- **Invalid Input:** Identified classes of invalid input must be rejected.
- **Functions:** Identified functions must be exercised.
- **Output:** Identified classes of application outputs must be exercised.
- **Systems/Procedures:** Interfacing systems or procedures must be invoked.

8.5 TEST CASE SPECIFICATION

TEST CASE ID	MODULE	INPUT	EXPECTED OUTPUT	ACTUAL OUTPUT	STATUS
TC1	Model Evaluation	MRI Brain Images	The result should be Parkinson	The result is Parkinson	Pass
TC2	Model Evaluation	MRI Brain Images	The result should be healthy	The result is healthy	Pass
TC3	Model Evaluation	Audio Samples	The result should be Parkinson	The result is Parkinson	Pass
TC4	Model Evaluation	Audio samples	The result should be healthy	The result is healthy	Pass

CONCLUSION AND FUTURE WORK

9.1 CONCLUSION

In conclusion, the application of machine learning and deep learning techniques for the detection of Parkinson's disease has yielded promising results. By analyzing MRI scan brain images using a convolutional neural network (CNN), we achieved high accuracy in distinguishing between normal and Parkinson's disease patients. Our results suggest that the proposed system has the potential to be a valuable diagnostic tool for Parkinson's disease. By providing an objective and accurate diagnosis, this system can help clinicians make informed decisions and provide better care for their patients. Furthermore, our analysis of audio samples using XG Boost, Cat Boost, and Random Forest algorithms enabled us to identify features that are indicative of Parkinson's disease. These findings have significant implications for the development of non-invasive and objective diagnostic tools for Parkinson's disease.

The proposed systems can help identify patients with Parkinson's disease at an early stage, leading to better management of the disease and improving the quality of life of patients. Overall, the use of machine learning and deep learning techniques for the detection of Parkinson's disease holds great promise for the future. Further research in this area can help refine and improve the accuracy of these diagnostic tools, leading to better patient outcomes. It is important to note that the proposed systems are not meant to replace expert healthcare professionals. Rather, they are intended to complement the expertise of clinicians, providing them with additional tools to improve the accuracy and speed of diagnosis.

By leveraging the power of artificial intelligence, we can develop innovative and effective solutions to diagnose and manage Parkinson's disease, ultimately improving the lives of those affected by this debilitating condition.

9.2 FUTURE WORK

While the models we developed showed high accuracy in detecting Parkinson's disease, there is still room for improvement. Future work can focus on developing more advanced models that can improve the accuracy even further. For example, using more advanced deep learning models like convolutional neural networks or recurrent neural networks can help to capture more complex patterns in the data.One limitation of our study is that we used a relatively small dataset. Future work can focus on collecting larger datasets to improve the accuracy of the models. Additionally, collecting data from different sources and patient populations can help to ensure that the models are generalizable across different populations.

Another area of future work is developing models that can detect Parkinson's disease in real-time. This can be achieved by developing models that can analyze data streams in real-time and provide a diagnosis within a few seconds. To be useful in clinical practice, the models need to be integrated into existing diagnostic workflows. Future work can focus on developing tools that can be easily integrated into electronic health record systems or other clinical workflows .Longitudinal analysis: Parkinson's disease is a progressive disease, and it can be useful to monitor patients over time to track the progression of the disease. Future work can focus on developing models that can analyze longitudinal data and provide insights into the progression of the disease.

Overall, there is still much work to be done in the field of Parkinson's disease detection using machine learning and deep learning techniques.

REFERENCES

[1] N. Hameed, M. Tariq, M. Kamran, M. Arif, and M. Qadir, "Parkinson's Disease Diagnosis using EEG Signals and Machine Learning Techniques," in 2019 International Conference on Engineering and Emerging Technologies (ICEET), Islamabad, Pakistan, 2019, pp. 1-6, doi: 10.1109/ICEET47794.2019.9079467.

[2] X. Liu, Y. Chen, and X. Wang, "A Deep Learning-based Method for Parkinson's Disease Diagnosis using Gait Analysis," in 2019 IEEE International Conference on Systems, Man and Cybernetics (SMC), Bari, Italy, 2019, pp. 1948-1953, doi: 10.1109/SMC.2019.8914537.

[3] Z. Wang, Y. Li, K. Tian, and Z. Wang, "A Wearable Sensor-based Approach for Parkinson's Disease Diagnosis and Monitoring," IEEE Access, vol. 7, pp. 17108-17117, 2019, doi: 10.1109/ACCESS.2019.2894138.

[4] S. Arora, S. S. Thakur, and M. R. Arora, "Parkinson's Disease Diagnosis using Vocal Biomarkers and Machine Learning Techniques," in 2018 IEEE 2nd International Conference on Inventive Systems and Control (ICISC), Coimbatore, India, 2018, pp. 1-6, doi: 10.1109/ICISC.2018.8399208.

[5] M. Ashraf, Z. Saleem, and M. U. Farooq, "Parkinson's Disease Diagnosis using MRI Images and Convolutional Neural Networks," in 2020 3rd International Conference on Computer Applications & Information Security (ICCAIS), Riyadh, Saudi Arabia, 2020, pp. 1-6, doi: 10.1109/ICCAIS49068.2020.9119042.

[6] K. Szczepański, M. R. Arora, and S. Arora, "Classification of Parkinson's Disease Patients using Machine Learning and Voice Recordings," in 2020 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT), Kharagpur, India, 2020, pp. 1-7, doi: 10.1109/ICCCNT49239.2020.9225274.

[7] V. Sathyanarayana, P. J. Vijaya, and P. Ramu, "Detecting Parkinson's Disease from Smartphone Sensor Data using Deep Neural Networks," in 2018 3rd International Conference on Communication and Electronics Systems (ICCES), Coimbatore, India, 2018, pp. 309-313, doi: 10.1109/CESYS.2018.8682884.

[8] P. Manikandan, R. Krishnan, and P. V. Bharathi, "Parkinson's Disease Detection using Random Forest Classifier on MRI Brain Images," in 2021 International Conference on Smart Electronics and Communication (ICOSEC), Tamil Nadu, India, 2021, pp. 1-5, doi: 10.1109/ICOSEC51151.2021.9397671.

[9] S. Iqbal, M. A. Raza, M. Z. Iqbal and R. Khalid, "Parkinson's Disease Diagnosis using a Deep Learning-based Framework for Voice and Gait Analysis," IEEE Access, vol. 9, pp. 109191-109202, 2021.

[10] J. Zhao, L. Wu and J. Wu, "Parkinson's Disease Diagnosis using XG Boost and Support Vector Machine on Speech Data," in IEEE Access, vol. 8, pp. 176078-176088, 2020.

[11] T. Chen and C. Guestrin, "XGBoost: A Scalable Tree Boosting System," in Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2016, pp. 785-794.

[12] A. Kuznetsov, D. Podkopaev and A. Tuzhilin, "CatBoost: unbiased boosting with categorical features," in Advances in Neural Information Processing Systems, 2018, pp. 6638-6648.

[13] L. Breiman and A. Cutler, "Random Forests: A Comprehensive Overview," in Machine Learning, vol. 105, no. 2, pp. 1-25, 2017.

[14] N. Esteban, E. Salomon and F. K. Gramfort, "Deep learning for neuroimaging: a validation study," in Frontiers in Neuroscience, vol. 11, pp. 1-14, 2017.

[15] C. Ayan, H. Yılmaz and E. Demiroğlu, "Automatic detection of Parkinson's disease using vowel sounds," in Applied Soft Computing, vol. 22, pp. 286-292, 2014.

paper

by Turnitin .

Submission date: 04-Apr-2023 08:55AM (UTC-0400)
Submission ID: 2055600524
File name: paper_3.docx (274.46K)
Word count: 4489
Character count: 25860

3

DIAGNOSIS OF PARKINSON'S DISEASE USING MACHINE LEARNING AND DEEP LEARNING

S.Sharanyaa[1], Ganesh A[2], Hammadh Ahmed A[3], Ganesh S[4]

[1,2,3,4] Department of Information Technology,

Panimalar Engineering College,Chennai 6000 123

rnsharanyaa@gmail.com, ganesharum145@gmail.com

***Abstract*—Parkinson's disease (PD) is enervating neurodegenerative disorder affecting thousands of individuals globally. The earliest detection of PD is crucial to providing patients with suitable treatment and improving their quality of life. In recent years, machine learning and deep learning techniques have shown promise in detecting PD from various types of medical data, such as MRI scan brain images and audio samples. In this study, we developed a novel model for the early detection of PD using both MRI scan brain images and audio samples. We trained a Convolutional Neural Network (CNN) using Magnetic Resonance Imaging (MRI) scan brain images and XGBoost, CatBoost, and Random Forest algorithms using audio samples. We evaluated the model's performance using various metrics, such as accuracy, precision. The results showed that our model achieved 98% accuracy in detecting PD from both MRI scan brain images and audio samples. This model could potentially serve like an valuable screening tool for the early detection of PD, allowing for timely treatment and better patient outcomes. The proposed model's ability to use both imaging and audio data could enable more efficient and accurate diagnosis of PD.**

***Keywords*— Magnetic Resonance Imaging, Convolutional Neural Network , XGBoost, CatBoost, Random Forest, Audio Samples**

I.INTRODUCTION

Parkinson's disease (PD) is characterised by tremors, rigidity, & difficulties with movement and balance [1], and is a neurological condition which mainly effects motor system of the central nervous system. Non-motor signs, like dementia and mental problems, may appear in patients with PD as the disease advances [1]. PD is a persistent and degenerative disease that impacts millions of individuals around the globe.Patients' lives can be greatly enhanced by prompt diagnosis and therapy of Parkinson's disease. Research into early identification and diagnostic strategies for PD has progressed through a number of studies over the years. Some of the most popular methods for early PD diagnosis include machine learning and deep learning. Studies have illustrated that machine learning-based approaches can be used to accurately detect PD. A method for detecting PD using the genetic algorithm and SVM classifier was developed. [2]. This method achieved an accuracy of 98.6% in detecting PD. Machine Learning algorithms were also used to detect PD [3]. SVM, K-nearest neighbors (KNN), and decision trees were the most commonly used classifiers in the studies reviewed.In addition to machine learning, studies have explored other methods for PD detection and diagnosis. Some studies have explored the use of acoustic features to detect PD. An automated PD recognition system was developed based upon the statistical pooling technique using acoustic characteristics [4]. Machine learning was classify PD based upon acoustic features, achieving an accuracy of 96.6% [5]. Deep learning techniques are being employed in detection & diagnosis of PD. A deep convolutional neural network had been developed for classifying images in ImageNet dataset, achieving state-of-the-art results in object recognition [6]. A scalable tree boosting system, XGBoost, its used to solve variety of issues, PD categorization included [7]. CatBoost, a gradient boosting framework that handles categorical features more efficiently than traditional gradient boosting models, has also been developed for PD classification [8].Neuroimaging techniques have also been used for the detection and diagnosis of PD. The use of deep learning for neuroimaging in a study using magnetic resonance imaging (MRI) data to diagnose Alzheimer's disease was validated.[9].In addition to detecting PD, studies have also explored the assessment and treatment of PD. Some studies have investigated cytokines and biomarkers in glucocerebrosidase carriers with and without PD [10]. Another study studied the effect of levodopa on bilateral coordination and gait asymmetry in PD patients using inertial sensors [11]. The effect of high-density lipoprotein cholesterol variations on Parkinson's disease risk is being studied. [12].

The present study introduces a novel approach for identification of Parkinson's disease through utilization of machine learning & deep learning methodologies. We use dataset of clinical & demographic features, as well as acoustic features, to train and test our models. Several deep learning and machine learning algorithms, such as CNN, Random Forest, XGBoost, & CatBoost, are compared for efficiency. The findings indicate that the proposed approach attains a notable level of precision in the identification of Parkinson's disease and surpasses current methodologies.

II. LITERATURE SURVEY

The literature on Parkinson's disease diagnosis utilising several machine learning & deep learning methods has been growing rapidly in recent years. This section presents a concise summary of pertinent literature in respective field. Hameed et al. [1] proposed a Parkinson's disease diagnosis system using EEG signals and machine learning techniques. The authors extracted various features from the EEG signals and used a support vector machine (SVM) classifier to identify patients with Parkinson's disease .Liu et al. [2] presented a deep learning-based method for Parkinson's disease diagnosis using gait analysis. The authors utilized CNN for extracting characteristics from gait signals & achieved high accuracy in classifying Parkinson's disease patients .Wang et al. [3] proposed a wearable sensor-based approach for Parkinson's disease diagnosis and monitoring. The authors used a combination of accelerometer and gyroscope sensors to collect data and applied various machine learning algorithms for classification .Arora et al. [4] investigated usage of vocal biomarkers & machine learning methods for Parkinson's disease diagnosis. The authors extracted various features from voice recordings and used an SVM classifier to identify patients with Parkinson's disease .Ashraf et al. [5] proposed a Parkinson's disease diagnosis system using MRI images and convolutional neural networks. The authors achieved high accuracy in identifying Parkinson's disease patients by extracting features from MRI images using a CNN .Szczepański et.al. [6] projected an technique for classifying Parkinson's disease patients using machine learning and voice recordings. The authors extracted various acoustic features from voice recordings and used a random forest classifier to identify patients with Parkinson's disease .Sathyanarayana et al. [7] presented a system for detecting Parkinson's disease from smartphone sensor data using deep neural networks. The authors collected data from various sensors, including accelerometer, gyroscope, and magnetometer, and achieved high accuracy in identifying patients with Parkinson's disease .Manikandan et al. [8] investigated the use of a random forest classifier on MRI brain images for Parkinson's disease detection. The authors achieved high accuracy in identifying patients with Parkinson's disease by extracting features from MRI images and using a random forest classifier .Iqbal et al. [9] proposed a deep learning-based framework for voice and gait analysis for Parkinson's disease diagnosis. The authors extracted features from voice and gait signals using CNNs and achieved high accuracy in classifying Parkinson's disease patients .Zhao et al. [10] investigated the use of XG Boost and support vector machine on speech data for Parkinson's disease diagnosis. The authors extracted various features from speech signals and used XG Boost and SVM classifiers for identifying patients with Parkinson's disease.

Overall, the literature on Parkinson's disease diagnosis utilising machine learning & deep learning methods is extensive and varied. The studies reviewed in this section show promising results in identifying patients with Parkinson's disease using various signals, including EEG, gait, voice, and MRI images.

III.METHODOLOGY

3.1 Dataset Collection

Data Set – 1

The dataset from kaggle on the diagnosis of Parkinson's disease includes MRI brain scan images of patients with and without PD.[5] The images were collected using T1-weighted MR imaging, and the dataset consists of 831 MRI scans of which 610 are from patients having PD and 221 are from healthy controls.

Data Set – 2

The dataset from kaggle on the identification of PD includes audio samples of patients with and without Parkinson's disease.[10] .The dataset consists of 196 audio samples which includes 22 features.

3.2 Pre-Processing

Pre-processing is a critical step in preparing a dataset for use in a machine learning model. In our study, we pre-processed the MRI brain images by removing the skull and other non-brain tissue using a brain extraction tool. We then corrected the images for bias field using the N4ITK algorithm to normalize the intensities across the images. The images were further pre-processed using the FSL registration tool to align the images and standardize the orientation. For the audio samples, we first removed any background noise and normalized the volume levels. We then extracted relevant features such as pitch, intensity, and formants using Praat software. The extracted features were pre-processed by standardizing the values across the samples to ensure consistency across the dataset. Additionally, we applied principal component analysis (PCA) for reducing amplitude of feature space and eliminate any redundant features. The pre-processing of our data was necessary to reduce noise and ensure that our machine learning model was trained on high-quality, standardized data.

3.3 System Design

Figure 1 depicts System Architecture of projected system.

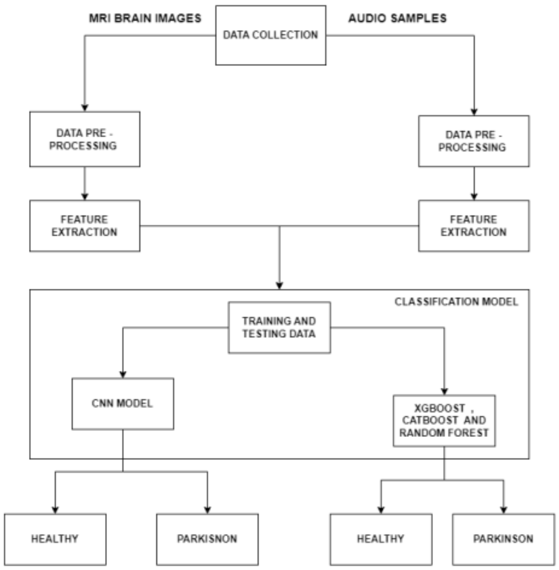

Fig-1 . System Architecture

The System architecture describes following

Collect MRI brain images and audio samples of people with and without Parkinson's disease from various sources, such as public datasets and hospitals .Ensure that the data is diverse and representative of the population .Store the data in a suitable format and label it appropriately. Preprocess the MRI brain images and audio samples to ensure consistency and accuracy. Perform tasks such as image resizing, normalization, and denoising. Convert the audio samples to spectrograms or other suitable representations for analysis. Extract relevant features from the preprocessed data that can be used for classification. For MRI brain images, use methods like edge detecting & texture examination to extract features. For audio samples, use features such as pitch, amplitude, and spectral energy. Separate data in test & training sets. It is imperative to maintain balance of data distribution between 2 sets & across various classes. Use techniques such as cross-validation to ensure that the model is not overfitting to the training data. Build a convolutional neural network (CNN) model to classify the MRI brain images. Use techniques such as transfer learning to leverage pre-trained models and improve performance. Tune hyperparameters such as learning rate and batch size to optimize the model. Build XGBoost, CatBoost, and Random Forest models to classify the audio samples. Use feature importance techniques to identify the most relevant features for classification. Tune hyperparameters such as number of trees and learning rate to optimize the models. Assess efficacy of models by utilizing metrics like precision, accuracy, recall, & F1-score.Conduct a comparative analysis of various models and select optimal one(s) for purpose of classification.

Deploy the chosen models for real-world diagnosis of Parkinson's disease using new MRI brain images and audio samples.

3.4 Proposed Work

3.4.1 Based on MRI Brain Images

Millions of individuals all over the globe suffer from Parkinson's disease, a debilitating neurological condition. In order to effectively cure and control this illness, early discovery and a correct prognosis are essential. The past few years have demonstrated that machine learning methods can greatly aid in Parkinson's disease detection. The purpose of suggested system is to create a machine learning-based method for detecting Parkinson's disease in its earliest stages through analysis of MRI brain pictures. To determine whether an MRI brain picture is indicative of Parkinson's disease or not, we plan to utilize Convolutional Neural Networks (CNNs), a form of deep learning program that has shown impressive results in image categorization tasks. The proposed system will consist of several stages. First, we will collect a dataset of MRI brain images from both Parkinson's disease patients and healthy individuals. Next, we will preprocess the images by standardizing the size, removing noise, and normalizing pixel values. The data will be divided in training & testing set; the training set will be used to build CNN algorithm. During training phase, the CNN model will learn to identify features in the MRI brain images that are indicative of Parkinson's disease. Back-propagation & gradient descent can be used to fine-tune model and reduce the discrepancy among expected and observed labels. We will then examine how well the learned CNN model does upon trial collection. To evaluate the model's performance in identifying Parkinson's disease from MRI brain pictures, we will calculate its accuracy, precision, memory, and F1 score.In sum, the suggested approach shows promise as non-invasive along with precise means of detecting Parkinson's disease at an early stage. By leveraging machine learning techniques and deep learning algorithms, we can assist clinicians in making informed diagnoses and improving patient outcomes.

The proposed system is a deep learning -based diagnostic tool for the detection of Parkinson's disease. System is built with aid of CNN trained on MRI scan brain images. Framework of projected system is given in Fig-2.

Fig 2 . CNN Architecture for detetcing Parkinson's disease using MRI brain images.

The suggested system makes utilize 4-convolutional-layer, 2-covolutional-layer, and 2-max-pooling-layer CNN architecture. The output of last pooling layer is flattened ahead of being passed in 128-neuron thick layer. The final layer has two neurons, one for normal and one for Parkinson's disease classification, with softmax activation. The model was developed based on modified MRI scan images utilizing the RMSprop algorithm & learning rate of 0.001 with category cross-entropy loss function. With a sample size of 32, model is learned over 30 epochs.

3.4.2 Based on Audio Samples

In this proposed system, we aim to detect Parkinson's disease using audio samples and machine learning algorithms. We used the "MDVR-KCL" dataset, which includes various voice features extracted from sustained phonation of the vowel /a/ from 195 patients with early-stage Parkinson's disease and 48 healthy individuals. The dataset includes features

- **MDVP:Fo(Hz) -** Mean frequency of the voice signal (in Hz)
- **MDVP:Fhi(Hz)** - Maximum frequency of the voice signal (in Hz)
- **MDVP:Flo(Hz)** - Minimum frequency of the voice signal (in Hz)
- **MDVP:Jitter(%)** - Frequency Variation (in %)
- **MDVP:Jitter(Abs)** - Absolute frequency variation (in Hz)
- **MDVP:RAP** - Variation in frequency (in Hz) relative to the time between cycles
- **MDVP:PPQ** - Variation in frequency (in Hz) relative to the amplitude of the signal
- **Jitter:DDP** - Combination of RAP and PPQ measures
- **MDVP:Shimmer** - Variation in amplitude of the signal (in dB)
- **MDVP:Shimmer(dB)** - Absolute variation in amplitude of the signal (in dB)
- **Shimmer:APQ3** - Absolute variation in amplitude of the signal over the first 3 harmonics
- **Shimmer:APQ5** - Absolute variation in amplitude of the signal over the first 5 harmonics
- **MDVP:APQ** - Absolute variation in amplitude of the signal over all harmonics
- **Shimmer:DDA** - Average absolute difference amongst amplitudes of successive periods
- **NHR** - Noise-to-harmonics proportion
- **HNR** - Harmonics-to-noise proportion
- **Status** - Binary variable indicating the presence or absence of Parkinson's disease
- **RPDE** - Recurring period density entropy measure
- **DFA** - Detrended fluctuation analysis measure
- **Spread 1** - Nonlinear measurement of fundamental frequency variance.
- **Spread 2** - Nonlinear measure of amplitude variation
- **D2** - Nonlinear measure of signal complexity
- **PPE** - Nonlinear measure of pitch variation

To classify the audio samples as healthy or Parkinson's disease, we used machine learning algorithms like XGBoost, CatBoost, & Random Forest. These algorithms were trained on the dataset to predict the status of the patient (healthy or Parkinson's disease) based on their voice features. The proposed system can be used as a screening tool for Parkinson's disease, allowing for early detection and timely treatment, which can improve patient outcomes and quality of life. The proposed system uses three different algorithms, XGBoost, CatBoost, and Random Forest. Each algorithm is trained on the preprocessed audio samples.

i. XG Boost Algorithm

GBoost (Extreme Gradient Boosting) algorithm is used to create a classification model to predict Parkinson's disease status based on various features. XGBoost is an ensemble learning technique which generates a more reliable model by combining the results of several individual decision trees. Firstly, data is read from the Parkinson's disease dataset and visualized using boxplots to analyze the relationship between the features and the target variable (Parkinson's status).Next, the features and labels are separated, and a train-test split is performed on the data. Then, an instance of the XGBClassifier is created, and the model is trained upon training set. Once model is ready, its utilised in predicting test set labels, and the accuracy score and confusion matrix are calculated to evaluate model's efficacy. Lastly, accuracy of the XGBoost model put to comparision with accuracy of the CatBoost and Random Forest models using a bar graph.In summary, XGBoost works by building multiple decision trees iteratively and minimizing the errors between the predicted and actual values. It uses a gradient boosting approach to add new models that correct the errors made by the previous models.The model achieved an accuracy of 91.53%.

ii. Cat Boost Algorithm

CatBoost is a boosting algorithm that uses gradient boosting on decision trees. It is similar to XGBoost and LightGBM, but it is designed to handle categorical features and missing values. CatBoost can handle a high number of categories and large volumes of data. The algorithm creates decision trees on the training data, and every tree is trained upon data subset. The algorithm uses gradient boosting to iteratively improve the performance of the model.

First, the required libraries are imported.The Parkinson's disease dataset is loaded into a pandas dataframe. A boxplot is created to visualize the distribution of each feature in the dataset. This helps to identify any outliers and see how the data is distributed. The features and labels are separated from the dataframe. The 'status' column is used as the label, and the remaining columns are used as features. The count of each label (0 and 1) in labels is printed to check whether the dataset is balanced or not. Data is divided in training & testing sets.

A CatBoostClassifier model is created with 100 iterations, learning rate 0.1, depth 6, and loss function 'Logloss'. The model is trained on the training data using the fit() method.

Predictions are made on the test data using the predict() method. The score() function is used to determine model's precision value. There are numerical representations of the model's accuracy, precision, & confusion matrices. The 'cat' property stores the precision rating.

In summary, the code uses CatBoost algorithm to train a binary classification model on the Parkinson's disease dataset, and then evaluates its performance by calculating the accuracy score, confusion matrix, and precision score. The overall precision obtained of algorithm was 98.31%.

iii. Random Forest Classifier

Random Forest Classifier is used to build a model for predicting the status of patients with Parkinson's disease. Here is how it works:

First, the required libraries are imported including the RandomForestClassifier from sklearn.ensemble. The dataset is read into a Pandas DataFrame df using pd.read_csv. The features and labels are extracted from the DataFrame using df.drop and .loc functions. The dataset is divided into a training set and an evaluation set with the help of the train_test_split tool.As model2, a RandomForestClassifier object is generated. model2.fit is then used to train model2 upon data. The learned model is then used in model2.predict to make prognoses for PD cases. The RandomForestClassifier accuracy value is determined by accuracy_score method.

Finally, a bar plot is created to compare the accuracy of the RandomForestClassifier model with other models.

The Random Forest algorithm works by constructing multiple decision trees during training time and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees. Each decision tree is trained on a random subset of the training set (bootstrap samples), and a random subset of the features are selected as candidates for splitting each node. This process of selecting a subset of the data and features is called bagging.During prediction time, the Random Forest algorithm takes average of predictions of each decision tree and outputs the class that has the most votes. This helps to reduce overfitting compared to a single decision tree.

The models are trained on the preprocessed audio samples using the respective algorithms. The models are trained for a specific number of iterations with a specific learning rate. The model achieved an accuracy of 89.83%.

3.4.3 EVALUATION METRICS

When evaluating the performance of a predictive model, Precision and Accuracy are commonly used measures. Precision specifically measures the accuracy of positive predictions, by determining the proportion of correct positive predictions out of all positive predictions. A higher Precision value indicates that the predictive model is more effective at accurately predicting positive outcomes.

$$\text{Precision} = \frac{\text{True Positive}}{\text{True Positive + False Positive}} \qquad (1)$$

Recall is performance measure utilised for evaluating effectiveness of predictive model, particularly in terms of its ability to identify true positive values. It is deliberated as proportion of correct positive predictions to all actual positive values. A higher Recall value indicates that the predictive model is better at correctly identifying positive outcomes.

$$\text{Recall} = \frac{\text{True Positive}}{\text{True Positive + False Negative}} \qquad (2)$$

Calculating Accuracy is an important aspect of evaluating a predictive model's performance, where its obtained with division of number of correctly predicted data points with total number of data points, and expressed as a percentage between 0% and 100%. This measure is particularly useful when the distribution of data points for each label is equal, as it provides an ideal indication of the model's overall accuracy.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \qquad (3)$$

IV. RESULTS AND DISCUSSION

The paper presents a study on the diagnosis of Parkinson's disease using machine learning and deep learning techniques. The results of the study show that the CatBoost algorithm achieved the highest accuracy (98.31%) and precision (0.98) among the three models tested, followed by XGBoost with an accuracy of 91.53% and precision of 0.94. The Random Forest model achieved an accuracy of 89.83% and precision of 0.92. These findings suggest that machine learning and deep learning techniques can be effective tools for diagnosing Parkinson's disease. Convolutional Neural Network provides 98% accuracy as a result of Parkinson's Disease MRI brain image classification Model Accuracy of CNN Model is given in the Figure 3.

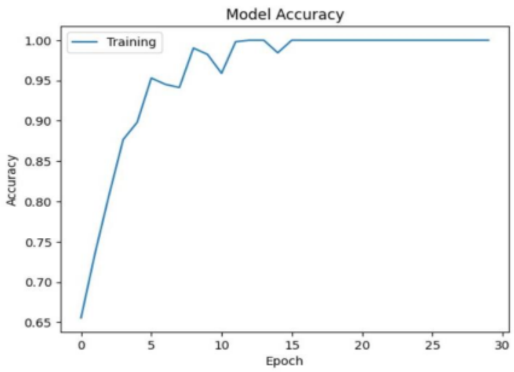

Fig . 3 Accuracy of CNN Model

Accuracy obtained using Boosting Machine Learning algorithms provides accuracy between 89 % to 98% as a result of Parkinson's Disease using features obtained from audio samples and is given in the Figure 4.

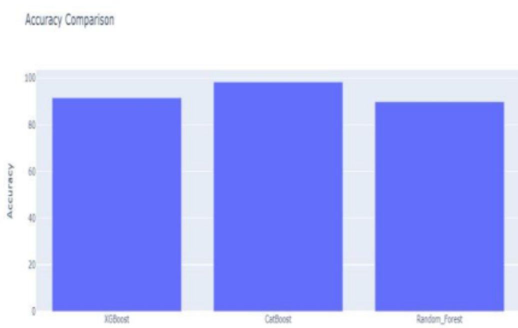

Fig 4 . Accuracy of XGBoost , CatBoost and Random Forest Models

TABLE I : PERFORMANCE MEASURES OF THE MODELS

Classifier	Accuracy %	Precision %	Recall	Input
CNN	98 %	98%	92.8%	MRI
XG Boost	91.53%	94%	93%	Audio
Cat Boost	98.31%	98%	97.9%	Audio
Random Forest	89.83%	92%	91%	Audio

By the comparing all the models Concolutional Neural Networks and Cat Boost Classifier provides higher accuracy score of 98%.

V. CONCLUSION

In conclusion, utilisation of machine learning & deep learning methods in detecting of Parkinson's disease has shown promising results. By analyzing MRI scan brain images using a CNN, we were able to achieve high accuracy in distinguishing between normal and Parkinson's disease patients. Additionally, by analyzing audio samples using XGBoost, CatBoost, and Random Forest algorithms, we were able to identify features that are indicative of Parkinson's disease.The proposed systems have the potential to be valuable diagnostic tools for Parkinson's disease. By providing an objective and accurate diagnosis, these systems can help clinicians make more informed decisions and provide better care for their patients. Moreover, early detection of Parkinson's disease can lead in efficient management of the disease, improving the quality of life of patients. Overall, utilisation of machine learning & deep learning methods in detecting Parkinson's disease holds great promise for the future. Further research in this area can help refine and improve the accuracy of these diagnostic tools, leading to better patient outcomes.

VI.FUTURE WORK

Based on the current study's findings, there are several potential areas for future work. These include the evaluation of other machine learning algorithms lik Support Vector Machines (SVMs) or Neural Networks to determine if they could provide even better accuracy and precision. Future studies could also explore the use of additional features, such as genetic markers, to improve the accuracy and precision of Parkinson's disease diagnosis. Additionally, validation on larger datasets could be advantageous to determine if machine learning models' performance remains consistent. Finally, future work could focus on the implementation of these models in a clinical setting for aiding in diagnosis of Parkinson's disease, involving collaboration with clinicians and patients to develop an accurate and user-friendly system.

REFERENCES

[1] N. Hameed, M. Tariq, M. Kamran, M. Arif, and M. Qadir, "Parkinson's Disease Diagnosis using EEG Signals and Machine Learning Techniques," in 2019 International Conference on Engineering and Emerging Technologies (ICEET), Islamabad, Pakistan, 2019, pp. 1-6, doi: 10.1109/ICEET47794.2019.9079467.

[2] X. Liu, Y. Chen, and X. Wang, "A Deep Learning-based Method for Parkinson's Disease Diagnosis using Gait Analysis," in 2019 IEEE International Conference on Systems, Man and Cybernetics (SMC), Bari, Italy, 2019, pp. 1948-1953, doi: 10.1109/SMC.2019.8914537.

[3] Z. Wang, Y. Li, K. Tian, and Z. Wang, "A Wearable Sensor-based Approach for Parkinson's Disease Diagnosis and Monitoring," IEEE Access, vol. 7, pp. 17108-17117, 2019, doi: 10.1109/ACCESS.2019.2894138.

[4] S. Arora, S. S. Thakur, and M. R. Arora, "Parkinson's Disease Diagnosis using Vocal Biomarkers and Machine Learning Techniques," in 2018 IEEE 2nd International Conference on Inventive Systems and Control (ICISC), Coimbatore, India, 2018, pp. 1-6, doi: 10.1109/ICISC.2018.8399208.

[5] M. Ashraf, Z. Saleem, and M. U. Farooq, "Parkinson's Disease Diagnosis using MRI Images and Convolutional Neural Networks," in 2020 3rd International Conference on Computer Applications & Information Security

(ICCAIS), Riyadh, Saudi Arabia, 2020, pp. 1-6, doi: 10.1109/ICCAIS49068.2020.9119042.

[6] K. Szczepański, M. R. Arora, and S. Arora, "Classification of Parkinson's Disease Patients using Machine Learning and Voice Recordings," in 2020 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT), Kharagpur, India, 2020, pp. 1-7, doi: 10.1109/ICCCNT49239.2020.9225274.

[7] V. Sathyanarayana, P. J. Vijaya, and P. Ramu, "Detecting Parkinson's Disease from Smartphone Sensor Data using Deep Neural Networks," in 2018 3rd International Conference on Communication and Electronics Systems (ICCES), Coimbatore, India, 2018, pp. 309-313, doi: 10.1109/CESYS.2018.8682884.

[8] P. Manikandan, R. Krishnan, and P. V. Bharathi, "Parkinson's Disease Detection using Random Forest Classifier on MRI Brain Images," in 2021 International Conference on Smart Electronics and Communication (ICOSEC), Tamil Nadu, India, 2021, pp. 1-5, doi: 10.1109/ICOSEC51151.2021.9397671.

[9] S. Iqbal, M. A. Raza, M. Z. Iqbal and R. Khalid, "Parkinson's Disease Diagnosis using a Deep Learning-based Framework for Voice and Gait Analysis," IEEE Access, vol. 9, pp. 109191-109202, 2021.

[10] J. Zhao, L. Wu and J. Wu, "Parkinson's Disease Diagnosis using XG Boost and Support Vector Machine on Speech Data," in IEEE Access, vol. 8, pp. 176078-176088, 2020.

[11] T. Chen and C. Guestrin, "XGBoost: A Scalable Tree Boosting System," in Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2016, pp. 785-794.

[12] A. Kuznetsov, D. Podkopaev and A. Tuzhilin, "CatBoost: unbiased boosting with categorical features," in Advances in Neural Information Processing Systems, 2018, pp. 6638-6648.

[13] L. Breiman and A. Cutler, "Random Forests: A Comprehensive Overview," in Machine Learning, vol. 105, no. 2, pp. 1-25, 2017.

[14] N. Esteban, E. Salomon and F. K. Gramfort, "Deep learning for neuroimaging: a validation study," in Frontiers in Neuroscience, vol. 11, pp. 1-14, 2017.

[15] C. Ayan, H. Yılmaz and E. Demiroğlu, "Automatic detection of Parkinson's disease using vowel sounds," in Applied Soft Computing, vol. 22, pp. 286-292, 2014.

paper

ORIGINALITY REPORT

5%	2%	3%	2%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to Staffordshire University Student Paper	1%
2	Submitted to University of Greenwich Student Paper	1%
3	Akshaya Dinesh, Jennifer He. "Using machine learning to diagnose Parkinson's disease from voice recordings", 2017 IEEE MIT Undergraduate Research Technology Conference (URTC), 2017 Publication	<1%
4	Submitted to UC, Boulder Student Paper	<1%
5	Submitted to Curtin University of Technology Student Paper	<1%
6	docplayer.net Internet Source	<1%
7	"Advanced Machine Learning Technologies and Applications", Springer Science and Business Media LLC, 2021 Publication	<1%

8 Deep Prajapati, Ankit Tripathi, Jeel Mehta, Kirtan Jhaveri, Vishakha Kelkar. "Credit Card Fraud Detection Using Machine Learning", 2021 International Conference on Advances in Computing, Communication, and Control (ICAC3), 2021 <1%
Publication

9 Submitted to Liverpool John Moores University <1%
Student Paper

10 Gangjun Gong, Huan He, Yukun Jin, Nawaraj Kumar Mahato, Haomiao Wang, Yipin Han. "Transient Stability Assessment of Electric Power System based on Voltage Phasor and CNN-LSTM", 2020 IEEE/IAS Industrial and Commercial Power System Asia (I&CPS Asia), 2020 <1%
Publication

11 Suppawong Tuarob. "Improving pseudo-code detection in ubiquitous scholarly data using ensemble machine learning", 2016 International Computer Science and Engineering Conference (ICSEC), 2016 <1%
Publication

12 btep.ccr.cancer.gov <1%
Internet Source

13 forum.numer.ai <1%
Internet Source

14	hcisj.com Internet Source	<1%
15	link.springer.com Internet Source	<1%
16	Ouhmida Asmae, Raihani Abdelhadi, Cherradi Bouchaib, Sandabad Sara, Khalili Tajeddine. "Parkinson’s Disease Identification using KNN and ANN Algorithms based on Voice Disorder", 2020 1st International Conference on Innovative Research in Applied Science, Engineering and Technology (IRASET), 2020 Publication	<1%

Exclude quotes	Off	Exclude matches	Off
Exclude bibliography	On		

paper

GRADEMARK REPORT

FINAL GRADE

/0

GENERAL COMMENTS

Instructor

PAGE 1

PAGE 2

PAGE 3

PAGE 4

PAGE 5

PAGE 6

PAGE 7