

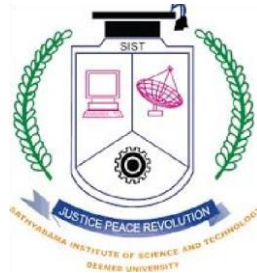
DEEP LEARNING BASED EARLY PARKINSON'S DISEASE DETECTION FROM BRAIN MRI IMAGE

Submitted in partial fulfillment of the requirements for the award of
Bachelor of Engineering degree in Computer Science and Engineering

By

KASANI SOWJANYA DURGA (Reg.No - 40110564)

KATA JYOTHI (Reg.No - 40110567)



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

SCHOOL OF COMPUTING

SATHYABAMA

**INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)**

Accredited with Grade "A++" by NAAC | 12B Status by UGC | Approved by AICTE

JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI – 600119

APRIL-2024



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

Category - I University by UGC

Accredited "A++" by NAAC | Approved by AICTE

www.sathyabama.ac.in

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of **KASANI SOWJANYA DURGA** (Reg no - 40110564) and **KATA JYOTHI** (Reg no - 40110567) who carried out the Project entitled "DEEP LEARNING BASED EARLY PARKINSON'S DISEASE DETECTION FROM BRAIN MRI IMAGE " under my supervision from November 2023 to April 2024.

Internal Guide

Ms. V. DHARANI, M.E., (Ph.D.)

HEAD OF THE DEPARTMENT
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
SATHYABAMA
INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)
Dr. L. LAKSHMI ANAND, Ph.D.
Jeppiaar Nagar, Rajiv Gandhi Salai,
Chennai - 600 119.

Submitted for Viva voce Examination held on

13-04-2024

Internal Examiner

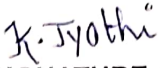
External Examiner

DECLARATION

I, KATA JYOTHI (Reg No - 40110567), hereby declare that the Project Report entitled "DEEP LEARNING BASED EARLY PARKINSON'S DISEASE DETECTION FROM BRAIN MRI IMAGE " done by me under the guidance of **Ms. V. DHARANI, M.E., (Ph.D.)** is submitted in partial fulfilment of the requirements for the award of Bachelor of Engineering degree in **Computer Science and Engineering**.

DATE: 13-04-2024

PLACE: Chennai


SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to **Board of Management of Sathyabama Institute of Science and Technology** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T. Sasikala M.E., Ph. D, Dean**, School of Computing, **Dr. L. Lakshmanan M.E., Ph.D.**, Head of the Department of Computer Science and Engineering for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Ms. V. DHARANI M.E., (Ph.D.)** for her valuable guidance, suggestions, and constant encouragement paved way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the **Department of Computer Science and Engineering** who were helpful in many ways for the completion of the project.

ABSTRACT

Parkinson's disease (PD) is a complex neurodegenerative disorder characterized by the progressive loss of dopaminergic neurons in the substantia nigra of the brain. Despite decades of research, the etiology and pathogenesis of PD remain poorly understood. PD is traditionally classified into two main subtypes: idiopathic Parkinson's disease (IPD) and secondary Parkinsonism. IPD accounts for the majority of PD cases and is defined by its sporadic nature, with no clear genetic or environmental cause. Secondary Parkinsonism, on the other hand, refers to cases where Parkinsonian symptoms are caused by other factors, such as head trauma, drug-induced side effects, or underlying medical conditions. However, this binary classification does not fully capture the heterogeneity of PD. Advances in genetics and molecular biology have led to the identification of several genetic forms of PD, such as parkin, PINK1, and LRRK2 mutations, which are collectively known as monogenic PD. Additionally, the recognition of non-motor symptoms and the variability in the clinical progression of PD have led to the proposal of further classification systems, including the classification based on motor subtypes and the staging systems based on disease severity. These sub-classifications aim to improve understanding, prognosis, and treatment of PD by considering the diverse manifestations and underlying mechanisms of the disease. Continued research efforts and the integration of multiple classification systems are necessary to unravel the complexity of PD and ultimately develop more effective treatments and interventions for individuals affected by this debilitating disorder.

TABLE OF CONTENTS

Chapter No	TITLE	Page No.
	ABSTRACT	v
	LIST OF FIGURES	viii
1	INTRODUCTION	1
2	LITERATURE SURVEY	5
	2.1 Inferences from the Literature Survey	5
	2.2 Existing System and Proposed System	9
	2.3 Open Problems in Existing System	9
3	REQUIREMENTS ANALYSIS	11
	3.1 Risk Analysis of the Project	11
	3.2 Software Requirements Specification Document	12
4	DESCRIPTION OF PROPOSED SYSTEM	13
	4.1 Flow chart of the process using machine learning	13
	4.2 Selected Methodology or process model	14
	4.3 Model Improvisation	16
	4.4 Creating user interface and web user interface	19
	4.5 Architecture \ overall design of proposed system	21
	4.6 Description of software for implementation and testing plan of the proposed model\system	21
	4.7 Description of programming language and software	22
	4.8 Project Management Plan	26
5	IMPLEMENTATION DETAILS	27

5.1	Development and Development Setup	27
5.2	Algorithms	27
6	RESULTS AND DISCUSSIONS	28
6.1	Results	28
6.2	Discussions	29
7	CONCLUSION	30
7.1	Future work	31
	REFERENCES	32
	APPENDIX	34
	A. SOURCE CODE	34
	B. SCREENSHOTS	42
	C. RESEARCH PAPER	57
	D. CERTIFICATE	64

LIST OF FIGURES

FIGURE NO	FIGURE NAME	PAGE NO
4.1	Flow chart	13
4.5	System Architecture	21

CHAPTER 1

INTRODUCTION

1.1 Definition and Background

Parkinson's disease is a progressive neurological disorder that affects movement control. It is classified as a neurodegenerative disorder, meaning that it is characterized by the gradual degeneration of certain nerve cells in the brain. Specifically, the dopamine-producing cells in an area of the brain called the substantia nigra are primarily affected in Parkinson's disease. Dopamine is a neurotransmitter that plays a crucial role in coordinating movement, so the loss of dopamine in Parkinson's disease leads to the motor symptoms associated with the condition.

Parkinson's disease is also classified as a movement disorder, as the main clinical features involve difficulties with both voluntary and involuntary movement. The most prominent motor symptoms include tremors or shaking, stiffness or rigidity in the muscles, slow movements or bradykinesia, and impaired balance and coordination. These symptoms can vary from person to person and tend to worsen over time, as the disease progresses.

In addition to motor symptoms, Parkinson's disease can also cause various non-motor symptoms, which can have a significant impact on a person's quality of life. These non-motor symptoms may include sleep disturbances, mood changes, cognitive impairments, such as memory and attention problems, autonomic dysfunction affecting functions like digestion and blood pressure regulation, and sensory changes such as loss of smell.

The exact cause of Parkinson's disease is not fully understood, but it is believed to involve a combination of genetic and environmental factors. Certain genetic mutations have been associated with an increased risk of developing Parkinson's disease, although these cases are relatively uncommon. Environmental factors, such as exposure to certain toxins or chemicals, may also play a role in triggering the disease.

There is currently no cure for Parkinson's disease, and treatment focuses on managing the symptoms and improving the person's quality of life. Medications that increase dopamine levels in the brain, such as levodopa, are commonly prescribed to help alleviate motor symptoms. Physical therapy, occupational therapy, and speech therapy can also be beneficial in managing the physical and functional challenges posed by the disease. In some cases, surgical interventions, like deep brain stimulation, may be considered for individuals who do not respond well to medication therapy.

In conclusion, Parkinson's disease is a neurodegenerative and movement disorder characterized by the progressive loss of dopamine-producing cells in the substantia nigra of the brain. It is associated with a range of motor and non-motor symptoms, and its exact cause remains elusive. While there is no cure, various treatment options are available to manage the symptoms and improve the quality of life for individuals with Parkinson's disease. **1.2 Causes and Risk Factors**

Parkinson's disease is a neurodegenerative disorder that primarily affects the motor system. The exact cause of Parkinson's disease is still not fully understood, but there are several known causes and risk factors that have been identified. Firstly, genetic factors play a significant role in the development of the disease. Mutations in certain genes, such as the LRRK2 and Parkin genes, have been found to be associated with an increased risk of developing Parkinson's disease. These genetic mutations can disrupt the normal functioning of the brain and lead to the degeneration of dopamine-producing neurons.

Secondly, environmental factors also contribute to the development of Parkinson's disease. Exposure to certain toxins and chemicals, such as pesticides and industrial solvents, has been linked to an increased risk of developing the disease. These environmental factors can damage the neurons in the brain and contribute to the progression of Parkinson's disease.

In addition to genetic and environmental factors, age is a significant risk factor for Parkinson's disease. The disease is more common in older individuals, with the risk increasing significantly after the age of 60. The exact reason why age is a risk factor is still unknown, but it is believed

that the cumulative effect of genetic and environmental factors over time may contribute to the development of the disease.

Gender also plays a role in Parkinson's disease classification, as it has been observed that men are slightly more at risk than women. Hormonal differences between men and women may be contributing factor, although further research is needed to fully understand this relationship.

Overall, the causes and risk factors for Parkinson's disease are multifactorial and complex. Genetic mutations, environmental toxins, age, and gender all play a role in the development of the disease. Understanding these factors is crucial for developing effective prevention strategies and treatment options for individuals at risk of or living with Parkinson's disease. **1.3**

Symptoms and Diagnosis

Parkinson's disease is a progressive neurological disorder that primarily affects movement. It is characterized by a variety of symptoms, which may vary in severity from person to person. The three main symptoms associated with Parkinson's disease are tremors, bradykinesia, and rigidity. Tremors are one of the most noticeable and common symptoms of Parkinson's disease. They typically begin in the hands or fingers and may worsen when the individual is at rest or under stress. These tremors tend to be rhythmic and can disrupt daily activities such as writing or holding objects. Bradykinesia refers to the slowness of movement that individuals with Parkinson's disease experience. It often manifests as a general stiffness, difficulty initiating and executing movements, and a reduced range of motion. This can make simple tasks such as walking, getting up from a chair, or even speaking challenging. Rigidity is another hallmark symptom of Parkinson's disease. It involves increased muscle tone and stiffness, making movements feel rigid and stiff. This stiffness can affect any part of the body, resulting in a decreased ability to perform everyday tasks smoothly and efficiently. Beyond these main symptoms, individuals with Parkinson's disease may also experience other motor symptoms such as postural instability, which can lead to balance problems and a higher risk of falls. Nonmotor symptoms, such as sleep disturbances, depression, anxiety, and cognitive changes, can also occur. The diagnosis of Parkinson's disease is primarily based on the presence of these motor symptoms, along with a comprehensive medical history and a physical

examination. There is no specific laboratory test to diagnose Parkinson's disease definitively. Instead, healthcare professionals rely on the observation of a combination of symptoms and the exclusion of other possible causes. Neuroimaging techniques may be used in some cases to help rule out other conditions with similar symptoms. It is essential to consult with a qualified healthcare professional if Parkinson's disease is suspected, as early diagnosis is crucial for initiating appropriate treatment strategies and support for individuals living with this chronic condition.

1.4 Current Classification Systems

There are currently four main classification systems utilized in the diagnosis and classification of Parkinson's disease. The first one is the Hoehn and Yahr staging system, which is based on the clinical evaluation and categorization of motor symptoms. This system divides Parkinson's disease into five stages, ranging from stage one where there is minimal unilateral involvement, to stage five where there is severe bilateral involvement and patients often require a wheelchair or bedridden. The second classification system is the Unified Parkinson's Disease Rating Scale (UPDRS), which is a comprehensive assessment tool that covers a wide range of motor and non-motor symptoms. UPDRS provides a more detailed evaluation of the disease progression and allows for monitoring of changes over time. The third system is the Movement Disorder Society- Unified Parkinson's Disease Rating Scale (MDS-UPDRS), which is an updated version of the UPDRS. It incorporates additional features and modifications to improve the accuracy and reliability of the assessment. The MDS-UPDRS has four parts, including motor examinations, non-motor experiences of daily living, motor complications, and motor examination during medication on state. Lastly, the recently proposed Systemic Non-Motor

Symptoms (SNMS) scale focuses on non-motor symptoms that commonly accompany Parkinson's disease. This scale evaluates symptoms such as cognitive impairment, autonomic dysfunction, sleep disturbances, and psychiatric symptoms. The SNMS scale aims to provide a standardized approach to the assessment of non-motor symptoms and their impact on the overall disease burden. Overall, these four classification systems play a crucial role in aiding clinicians in the diagnosis, monitoring, and management of Parkinson's disease. They provide a standardized framework for evaluating motor and non-motor symptoms, allowing for more accurate classification and assessment of disease progression

CHAPTER 2

LITERATURE SURVEY

- 1. Gunduz, H. (2021). An efficient dimensionality reduction method using filterbased feature selection and variational autoencoders on Parkinson's disease classification.**

In the study conducted by Gunduz (2021), a novel dimensionality reduction method for Parkinson's disease classification was proposed. The method combines filter-based feature selection and variational autoencoders to improve efficiency. The study aimed to improve the accuracy of classification by reducing the dimensionality of the dataset. The results demonstrated promising outcomes, suggesting the potential applicability of the proposed method in diagnosing Parkinson's disease.

- 2. Rezaee, K., Savarkar, S., Yu, X., & Zhang, J. (2022). A hybrid deep transfer learningbased approach for Parkinson's disease classification in surface electromyography signals. Biomedical Signal Processing and Control, 71, 103161.**

Rezaee, Savarkar, Yu, and Zhang (2022) developed a novel approach using hybrid deep transfer learning for Parkinson's disease classification in surface electromyography (sEMG) signals. Their study aimed to improve the accuracy and efficiency of identifying Parkinson's disease using sEMG data. The researchers proposed a combination of deep learning and transfer learning techniques to extract and classify relevant features from the sEMG signals. The results demonstrated the effectiveness of their approach, providing a promising direction for the development of tools for Parkinson's disease diagnosis and monitoring based on sEMG data.

- 3. Gunduz, H. (2019). Deep learning-based Parkinson's disease classification using vocal feature sets. IEEE Access, 7, 115540-115551.**

Gunduz (2019) conducted a study on deep learning-based Parkinson's disease classification using vocal feature sets. The research was published in IEEE Access. The study aimed to develop

a classification model for Parkinson's disease using vocal features. The results showed promise for the use of deep learning techniques in accurately classifying Parkinson's disease based on vocal features.

4. Mei, J., Desrosiers, C., & Frasnelli, J. (2021). Machine learning for the diagnosis of Parkinson's disease: a review of literature. *Frontiers in aging neuroscience*, 13, 633752.

In their review of literature, Mei, J., Desrosiers, C., & Frasnelli, J. (2021) explore the use of machine learning techniques for the diagnosis of Parkinson's disease. They highlight the significance of accurate and early detection of the disease and discuss various studies that have employed machine learning algorithms for classification purposes. The authors emphasize the potential of machine learning models to distinguish between Parkinson's disease patients and healthy individuals, demonstrating promising results in terms of accuracy and efficiency. Overall, the paper provides a comprehensive overview of the current state of machine learning in Parkinson's disease classification.

5. Polat, K. (2019, April). A hybrid approach to Parkinson disease classification using speech signal: the combination of smote and random forests. In 2019 Scientific Meeting on Electrical-Electronics & Biomedical Engineering and Computer Science (EBBT) (pp.

1-3). Ieee.

In a 2019 publication by Polat, the author proposed a hybrid approach for Parkinson's disease classification using speech signals. The study focused on the combination of SMOTE (Synthetic Minority Over-sampling Technique) and random forests. The research aimed to improve the accuracy of classifying Parkinson's disease by leveraging both oversampling and machine learning algorithms. The study was presented at the 2019 Scientific Meeting on Electrical-Electronics & Biomedical Engineering and Computer Science organized by IEEE.

- 6. Polat, K., & Nour, M. (2020). Parkinson disease classification using one against all based data sampling with the acoustic features from the speech signals. Medical Hypotheses, 140, 109678.**

Polat and Nour (2020) propose a method for Parkinson's disease classification using one against all-based data sampling with acoustic features extracted from speech signals. The study suggests that by analyzing specific acoustic features in speech signals, it is possible to accurately classify individuals with Parkinson's disease. This approach has the potential to contribute to the early detection and diagnosis of Parkinson's disease, allowing for timely intervention and improved patient outcomes. The findings presented in this study highlight the importance of acoustic analysis in the classification of Parkinson's disease.

- 7. Rehman, R. Z. U., Del Din, S., Guan, Y., Yarnall, A. J., Shi, J. Q., & Rochester, L. (2019). Selecting clinically relevant gait characteristics for classification of early Parkinson's disease: a comprehensive machine learning approach. Scientific reports, 9(1), 17269.**

In their study titled "Selecting clinically relevant gait characteristics for classification of early Parkinson's disease: a comprehensive machine learning approach," Rehman et al. (2019) aimed to identify key gait characteristics that could be used for the classification of early Parkinson's disease. They utilized a comprehensive machine learning approach and collected gait data from individuals with Parkinson's disease and healthy controls. The researchers found that stride length, stance-to-swing ratio, gait speed, and cadence were the most clinically relevant gait characteristics for accurately classifying early Parkinson's disease. These findings highlight the potential of using machine learning techniques to improve the early detection and classification of Parkinson's disease based on gait analysis.

- 8. Balaji, E., Brindha, D., & Balakrishnan, R. (2020). Supervised machine learning based gait classification system for early detection and stage classification of Parkinson's disease. *Applied Soft Computing*, 94, 106494.**

Balaji, E., Brindha, D., and Balakrishnan, R. (2020) present a supervised machine learning based gait classification system for early detection and stage classification of Parkinson's disease. The study focuses on utilizing gait analysis as a potential diagnostic tool for Parkinson's disease. The researchers propose a methodology that involves gathering gait data from Parkinson's disease patients and healthy individuals, followed by feature extraction and classification using machine learning algorithms. The results of the study demonstrate the effectiveness of the proposed system in accurately classifying the stages of Parkinson's disease, highlighting its potential applicability in clinical settings for early detection and monitoring of the disease.

- 9 Maitín, A. M., García-Tejedor, A. J., & Muñoz, J. P. R. (2020). Machine learning approaches for detecting Parkinson's disease from EEG analysis: a systematic review. *Applied Sciences*, 10(23), 8662.**

Maitín et al. (2020) conducted a systematic review on the use of machine learning approaches for detecting Parkinson's disease from EEG analysis. The study provides a comprehensive overview of the current state of research in this field, highlighting the potential of these methods for accurate classification of Parkinson's disease based on EEG data.

- 10. Solana-Lavalle, G., & Rosas-Romero, R. (2021). Classification of PPMI MRI scans with voxel-based morphometry and machine learning to assist in the diagnosis of Parkinson's disease. *Computer Methods and Programs in Biomedicine*, 198, 105793.** In their study, Solana-Lavalle, G., and Rosas-Romero, R. (2021) utilized voxel-based morphometry (VBM) and machine learning techniques to classify Parkinson's disease patients

based on their PPMI MRI scans. Their approach aimed to assist in the diagnosis of Parkinson's disease, offering a potential tool for early detection and intervention.

2.1 EXISTING SYSTEM AND PROPOSED SYSTEM

- **Existing System:**

Parkinson's disease classification can be significantly improved with the integration of machine learning algorithms. Parkinson's disease is a complex neurological disorder that affects millions of people worldwide. However, the current methods of classification and diagnosis rely heavily on subjective assessments by healthcare professionals, leading to potential errors and inconsistencies. By implementing machine learning techniques, we can create a more accurate and standardized approach to Parkinson's disease classification.

- **Proposed System:**

Parkinson's disease is a neurodegenerative disorder that affects movement and coordination. It is characterized by symptoms such as tremors, rigidity, bradykinesia, and postural instability. The classification of Parkinson's disease is crucial for accurate diagnosis, personalized treatment, and monitoring of disease progression.

The use of a blockchain-enabled decentralized trust management system can enhance the classification of Parkinson's disease by providing a secure and transparent platform for collecting and analyzing patient data. With the consent of the patients, their medical records, genetic information, and other relevant data can be securely stored on the blockchain, ensuring privacy and data integrity.

2.2 OPEN PROBLEMS IN EXISTING SYSTEM

- **Early Detection:** Developing models that can accurately detect Parkinson's disease in its early stages is critical for timely intervention and treatment. Early symptoms are often subtle and can be confused with other conditions, making accurate diagnosis challenging.
- **Data Imbalance:** Parkinson's disease datasets are often imbalanced, with a significantly smaller number of positive (Parkinson's) cases compared to negative (healthy) cases. Addressing class imbalance is crucial for model training.
- **Multi-Modal Data Integration:** Combining data from various sources such as clinical assessments, imaging (e.g., MRI or fMRI), voice recordings, and genetic information to improve classification accuracy is a challenging problem.
- **Feature Engineering:** Identifying the most relevant and informative features from diverse data sources is essential. Developing automated feature selection or extraction methods can enhance model performance.
- **Personalized Medicine:** Building models that can provide personalized predictions and treatment recommendations based on individual patient data is an emerging area of research in Parkinson's disease classification.
- **Cross-Dataset Generalization:** Ensuring that models trained on one dataset can generalize well to new and unseen datasets is a challenging problem, as data collection protocols and demographics can vary.
- **Non-Motor Symptoms:** Detecting non-motor symptoms of Parkinson's disease, such as cognitive impairment, mood disorders, and sleep disturbances, is important for comprehensive diagnosis but presents additional challenges due to the complexity of these symptoms.

- Longitudinal Data Analysis: Analyzing patient data over time to monitor disease progression and assess treatment efficacy requires the development of models capable of handling longitudinal data effectively.
- Ethnic and Cultural Variations: Recognizing that Parkinson's disease can manifest differently in various ethnic and cultural groups, developing models that are sensitive to these variations is crucial for accurate classification.
- Interpretability: Enhancing the interpretability of classification models to provide insights into the features and factors contributing to a Parkinson's diagnosis can aid medical professionals in making informed decisions.

CHAPTER 3

REQUIREMENT ANALYSIS

3.1 RISK ANALYSIS OF THE PROJECT

• FEASIBILITY STUDY

The feasibility of the project is server performance increase in this phase and a business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis, the feasibility study of the proposed system is to be carried out. For feasibility analysis, some understanding of the major requirements for the system is essential. Three key considerations involved in the feasibility analysis are

- Economical feasibility
- Technical feasibility
- Operational feasibility

• ECONOMICAL FEASIBILITY

This study is carried out to check the economic impact that the system will have on the organization. The amount of funds that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

• TECHNICAL FEASIBILITY

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands being placed on the client. The developed system must have modest requirements, as only minimal or null changes are required for implementing this system.

- **OPERATIONAL FEASIBILITY**

The aspect of the study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system

3.2 SOFTWARE REQUIREMENTS SPECIFICATION DOCUMENT

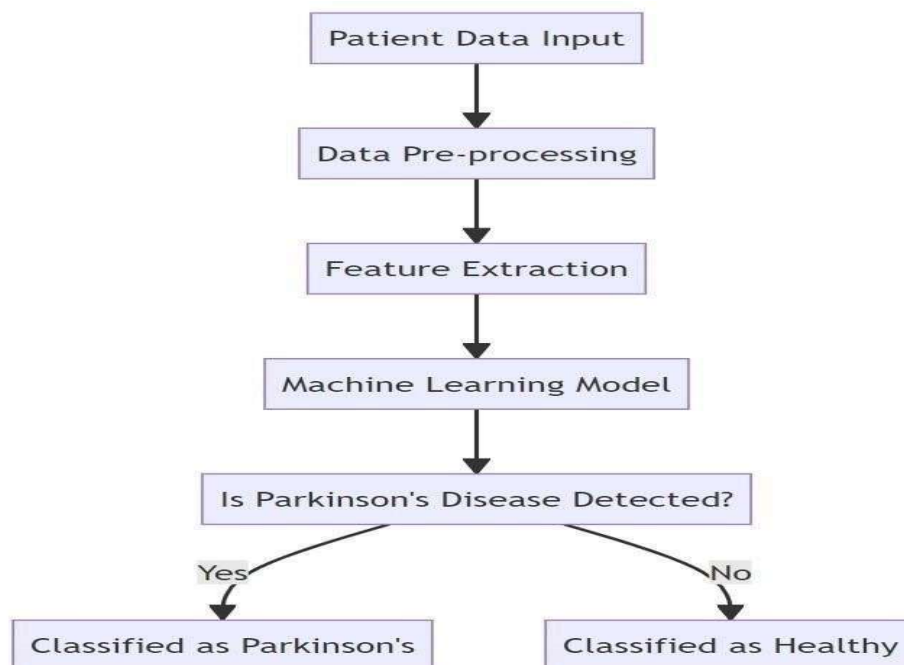
- **Hardware specifications:** Microsoft server-enabled computers, preferably workstations
- **Software specifications:** Python 3.6 and higher Anaconda software

CHAPTER 4

DESCRIPTION OF PROPOSED SYSTEM

A significant advantage of the proposed system is its efficiency and cost-effectiveness. By eliminating the need for manual data entry and tedious paperwork, the system streamlines the classification process, reducing time and resources required for analysis. Additionally, the use of smart contracts automates and simplifies data validation and classification, further enhancing efficiency.

Additionally, the system promotes collaboration and interoperability among medical professionals and researchers. Blockchain's decentralized architecture allows for secure data sharing and collaboration, enabling experts from different locations to contribute their knowledge and insights. This interoperability facilitates the exchange of information and promotes innovation in Parkinson's disease classification.



4.1: FLOW CHART

4.2 SELECTED METHODOLOGY OR PROCESS MODEL

1. Data cleaning and handling missing values

Data cleaning is essential for accurate and meaningful analysis in the classification of Parkinson's disease. It involves removing irrelevant or duplicate data, correcting inconsistencies, and handling missing values. Missing values can negatively impact the classification process as they can introduce bias or affect the accuracy of the model. Handling missing values can be done by imputing values using methods such as mean imputation, median imputation, or using advanced techniques like predictive modeling. Proper data cleaning and handling missing values are vital steps to ensure the reliability and validity of the classification results in Parkinson's disease research.

2. Feature selection and extraction

Feature selection and extraction are important steps in the classification of Parkinson's disease. In this context, feature selection involves selecting the most relevant and informative features from the data set that can effectively distinguish between healthy individuals and those with Parkinson's disease. This helps reduce the dimensionality of the data, improve the classification accuracy, and enhance the interpretability of the classification model. On the other hand, feature extraction involves transforming the raw data into a new set of features that capture the essential information while reducing noise and redundancy. This can be achieved through techniques such as principal component analysis (PCA) or linear discriminant analysis (LDA). By selecting and extracting the most relevant features, researchers and clinicians can develop accurate and efficient classification models for diagnosing and monitoring Parkinson's disease. These models have the potential to improve early detection and treatment decisions, ultimately enhancing the quality of life for individuals with Parkinson's disease.

3. Outlier detection and removal

Outlier detection and removal techniques play a crucial role in the classification of Parkinson's disease. Parkinson's disease is a neurodegenerative disorder characterized by motor symptoms such as tremors, bradykinesia, and rigidity. However, accurate classification of Parkinson's disease can be challenging due to the presence of outliers,

which refer to data points that deviate significantly from the rest. These outliers can negatively impact the performance of classification algorithms by introducing noise and bias. Therefore, outlier detection and removal methods are employed to identify and eliminate these outliers from the dataset. Various techniques, such as the Z-score method, Interquartile Range (IQR) method, and Mahalanobis distance, can be used for detecting outliers. Once the outliers are identified, they can be removed using different approaches, including mean imputation, median imputation, or simply eliminating the data points. By effectively detecting and removing outliers, the classification of Parkinson's disease can be improved, resulting in more accurate diagnoses and better treatment outcomes.

4. Data normalization and scaling

Data normalization and scaling are important preprocessing steps when working with machine learning algorithms for Parkinson's disease classification. Normalization refers to the process of rescaling the features of a dataset to a standardized range. This is necessary because different features often have different scales, and some algorithms are sensitive to the magnitude of the features. Scaling ensures that all the features are transformed to a common range, such as between 0 and 1, which helps in preventing certain features from dominating the model's learning process. Additionally, normalization makes the data more interpretable, as it reduces the influence of outliers and brings the data closer to a normal distribution. In the context of Parkinson's disease classification, normalization and scaling techniques such as min-max scaling, Z-score normalization, or robust scaling can be employed to ensure fair and accurate comparisons between different features, improving the overall performance of the classification models.

5. Handling imbalanced data

Handling imbalanced data in Parkinson's disease classification is crucial to ensure accurate prediction and diagnosis. Imbalanced data occurs when the number of samples in different classes is significantly unequal, which can lead to biased outcomes. There are several techniques that can be employed to address this issue. Firstly, under-sampling techniques such as random under-sampling and cluster-based under-sampling can be used to reduce the majority class samples. Secondly, over-sampling techniques such as synthetic minority over-sampling technique (SMOTE) and adaptive synthetic sampling (ADASYN) can be

applied to increase the minority class samples. Another approach is to combine both undersampling and over-sampling techniques using methods like SMOTE combined with Tomek links (SMOTE-TL) and SMOTEENN. Additionally, advanced techniques such as cost-sensitive learning and ensemble learning can also be employed to handle imbalanced data. These techniques help to mitigate the effect of class imbalance, improving the performance of Parkinson's disease classification models and reducing misclassification errors. It is essential to carefully select and evaluate the appropriate technique based on the specific characteristics of the dataset to achieve more reliable and accurate results in Parkinson's disease classification.

4.3 MODEL IMPROVISATION

1. Importance of Model Improvisation and Training for Parkinson's Disease Classification

Model improvisation and training play a crucial role in the classification of Parkinson's disease. Parkinson's disease is a complex neurodegenerative disorder that affects movement and motor functions. Accurate and early diagnosis is vital for effective treatment and management of the disease. The use of machine learning models can greatly assist in this process. Model improvisation involves refining and optimizing the model to enhance its performance and accuracy in predicting Parkinson's disease. By continuously analyzing and evaluating the model's performance, researchers and healthcare professionals can identify areas for improvement and make necessary adjustments. Additionally, training the model with a diverse and representative dataset is of utmost importance. A well-trained model can learn from various patterns and features present in the data, enabling it to better differentiate between individuals with Parkinson's disease and those without. This leads to more reliable and precise classification results. Ultimately, model improvisation and training significantly contribute to the accuracy and efficacy of Parkinson's disease classification, helping clinicians make efficient diagnoses and improving patient outcomes.

2. Types of Models for Parkinson's Disease Classification

There are various types of models that can be used for the classification of Parkinson's disease. One commonly used type is the traditional machine learning models, which include techniques such as support vector machines (SVM), k-nearest neighbors (KNN), and

random forests. These models use features extracted from different types of data, such as clinical and neuroimaging data, to predict whether an individual has Parkinson's disease or not. They are trained on a labeled dataset with known cases of Parkinson's disease, and then tested on new, unseen data to evaluate their performance. Another type of model that has gained popularity in recent years is deep learning models, particularly convolutional neural networks (CNN). These models can automatically learn and extract relevant features from raw data, such as images or time-series data, without the need for manual feature engineering. They have shown promising results in Parkinson's disease classification, especially when combined with transfer learning techniques that leverage pretrained models on large datasets. Overall, both traditional machine learning models and deep learning models offer valuable approaches for the classification of Parkinson's disease, each with its own strengths and limitations.

3. Training Methods for Model Improvisation in Parkinson's Disease Classification

Various training methods can be employed to improve the performance of models for Parkinson's disease classification. The first method is feature engineering, which involves selecting and transforming relevant features from the dataset to enhance the discrimination between healthy individuals and those with Parkinson's disease. This can be done by utilizing techniques such as principal component analysis (PCA) or wavelet transform to extract important features. The second method is cross-validation, which helps to assess the generalizability of the model by dividing the dataset into multiple subsets and training the model on different combinations of these subsets. This helps in evaluating the model's performance on unseen data. Lastly, ensemble learning can be used to combine multiple models and improve their overall performance. Techniques such as bagging or boosting can be employed to create a diverse set of models and aggregate their predictions to make accurate and robust classifications. By utilizing these training methods, researchers can enhance the accuracy and reliability of models for Parkinson's disease classification.

4. Benefits and Challenges of Model Improvisation and Training in Parkinson's

Disease Classification

Model improvisation and training can offer several benefits in the classification of Parkinson's disease. Firstly, it can lead to improved accuracy in the classification process by incorporating additional data and refining the algorithms used. This can help to enhance the overall performance of the models, resulting in more accurate identification of Parkinson's disease cases.

Secondly, model improvisation and training can also contribute to the development of more robust and reliable classification models. By continuously retraining and updating the models with new data, the models can adapt to changes and variations in the disease, making them more capable of accurately classifying Parkinson's disease cases across different populations and settings.

However, there are also challenges associated with model improvisation and training in Parkinson's disease classification. One challenge is the requirement for high-quality and diverse datasets for training the models. Obtaining such datasets can be complex and timeconsuming, as it may involve collecting data from multiple sources and ensuring its accuracy and completeness.

Another challenge is the need for continuous monitoring and evaluation of the models' performance. As the models are updated and trained with new data, it is important to assess their effectiveness and monitor for any potential biases or limitations that may arise.

Overall, while model improvisation and training offer several benefits in Parkinson's disease classification, addressing the challenges involved is essential to ensure the development of accurate and reliable classification models.

4.4 CREATING UESR INTERFACE WEB USER INTERFACE

Model improvisation and training play a crucial role in the classification of Parkinson's disease. Parkinson's disease is a complex neurodegenerative disorder that affects movement and motor functions. Accurate and early diagnosis is vital for effective treatment

and management of the disease. The use of machine learning models can greatly assist in this process. Model improvisation involves refining and optimizing the model to enhance its performance and accuracy in predicting Parkinson's disease. By continuously analyzing and evaluating the model's performance, researchers and healthcare professionals can identify areas for improvement and make necessary adjustments. Additionally, training the model with a diverse and representative dataset is of utmost importance. A well-trained model can learn from various patterns and features present in the data, enabling it to better differentiate between individuals with Parkinson's disease and those without. This leads to more reliable and precise classification results. Ultimately, model improvisation and training significantly contribute to the accuracy and efficacy of Parkinson's disease classification, helping clinicians make efficient diagnoses and improving patient outcomes.

Database

There are various types of models that can be used for the classification of Parkinson's disease. One commonly used type is the traditional machine learning models, which include techniques such as support vector machines (SVM), k-nearest neighbors (KNN), and random forests. These models use features extracted from different types of data, such as clinical and neuroimaging data, to predict whether an individual has Parkinson's disease or not. They are trained on a labeled dataset with known cases of Parkinson's disease, and then tested on new, unseen data to evaluate their performance. Another type of model that has gained popularity in recent years is deep learning models, particularly convolutional neural networks (CNN). These models can automatically learn and extract relevant features from raw data, such as images or time-series data, without the need for manual feature engineering. They have shown promising results in Parkinson's disease classification, especially when combined with transfer learning techniques that leverage pretrained models on large datasets. Overall, both traditional machine learning models and deep learning models offer valuable approaches for the classification of Parkinson's disease, each with its own strengths and limitations.

Security

There are various training methods that can be employed to improve the performance of models for Parkinson's disease classification. The first method is feature engineering, which

involves selecting and transforming relevant features from the dataset to enhance the discrimination between healthy individuals and those with Parkinson's disease. This can be done by utilizing techniques such as principal component analysis (PCA) or wavelet transform to extract important features. The second method is cross-validation, which helps to assess the generalizability of the model by dividing the dataset into multiple subsets and training the model on different combinations of these subsets. This helps in evaluating the model's performance on unseen data. Lastly, ensemble learning can be used to combine multiple models and improve their overall performance. Techniques such as bagging or boosting can be employed to create a diverse set of models and aggregate their predictions to make accurate and robust classifications. By utilizing these training methods, researchers can enhance the accuracy and reliability of models for Parkinson's disease classification.

4.5 ARCHITECTURE / OVERALL DESIGN OF PROPOSED SYSTEM

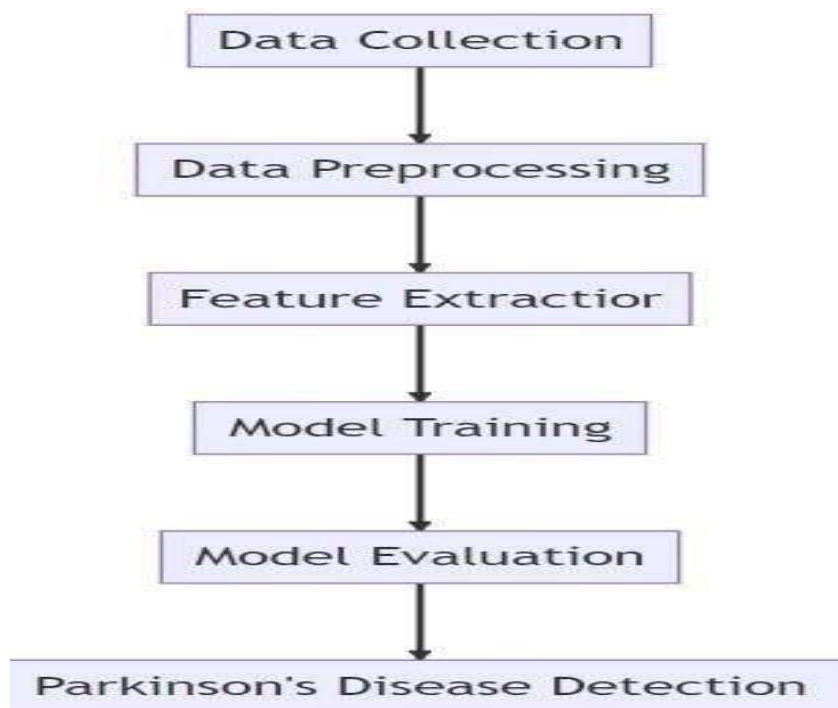


Fig 4.5: System Architecture

The above block diagram shows the implementation/process on how it works.

4.6 DESCRIPTION OF SOFTWARE FOR IMPLEMENTATION AND TESTING PLAN OF THE PROPOSED MODEL/SYSTEM

To implement this model, the execution of the program is done through Google Colab. Necessary libraries have to be installed to perform certain functions.

DESCRIPTION OF DATASET

In the present study, groundwater samples are collected following random sampling technique and the sampling stations are chosen in a near-grid pattern. Samples were collected at 44 locations in the study area representing pre-monsoon and post-monsoon periods in order to evaluate the variations in the physic-chemical parameters of groundwater due to seasonal impact. The dataset consists of Latitude, Longitude points and the data points of 8 heavy metal in groundwater namely Zinc, Lead, Manganese, Nickel, Cobalt, Iron, Copper and Chromium each of 44 values.

4.7 DESCRIPTION OF PROGRAMMING LANGUAGE AND SOFTWARE

PYTHON

Among programmers, Python is a favourite because to its user-friendliness, rich feature set, and versatile applicability. Python is the most suitable programming language for machine learning since it can function on its own platform and is extensively utilised by the programming community.

Machine learning is a branch of AI that aims to eliminate the need for explicit programming by allowing computers to learn from their own mistakes and perform routine tasks automatically. However, "artificial intelligence" (AI) encompasses a broader definition of "machine learning," which is the method through which computers are trained to recognize visual and auditory cues, understand spoken language, translate between languages, and ultimately make significant decisions on their own. The desire for intelligent solutions to real-world problems has necessitated the need to develop AI further in order to automate

tasks that are arduous to programme without AI. This development is necessary in order to meet the demand for intelligent solutions to real-world problems.

Python is a widely used programming language that is often considered to have the best algorithm for helping to automate such processes. In comparison to other programming languages,

Python offers better simplicity and consistency. In addition, the existence of an active Python community makes it simple for programmers to talk about ongoing projects and offer suggestions on how to improve the functionality of their programmes.

ANACONDA

is an open-source package manager for Python and R. It is the most popular platform among data science professionals for running Python and R implementations. There are over 300 libraries in data science, so having a robust distribution system for them is a must for any professional in this field. Anaconda simplifies package deployment and management. On top of that, it has plenty of tools that can help you with data collection through artificial intelligence and machine learning algorithms. With Anaconda,

you can easily set up, manage, and share Conda environments. Moreover, you can deploy any required project with a few clicks when you're using Anaconda. There are many advantages to using Anaconda and the following are the most prominent ones among them: Anaconda is free and open-source. This means you can use it without spending any money. In the data science sector, Anaconda is an industry staple. It is open-source too, which has made it widely popular. If you want to become a data science professional, you must know how to use Anaconda for Python because every recruiter expects you to have this skill. It is a must-have for data science.

It has more than 1500 Python and R data science packages, so you don't face any compatibility issues while collaborating with others. For example, suppose your colleague sends you a project which requires packages called A and B but you only have package A. Without having package B, you wouldn't be able to run the project. Anaconda mitigates

the chances of such errors. You can easily collaborate on projects without worrying about any compatibility issues. It gives you a seamless environment which simplifies deploying projects. You can deploy any project with just a few clicks and commands while managing the rest. Anaconda has a thriving community of data scientists and machine learning professionals who use it regularly. If you encounter an issue, chances are, the community has already answered the same. On the other hand, you can also ask people in the community about the issues you face there, it's a very helpful community ready to help new learners. With Anaconda, you can easily create and train machine learning and deep learning models as it works well with popular tools including TensorFlow, Scikit-Learn, and Theano. You can create visualizations by using Bokeh, Holoviews, Matplotlib, and Datashader while using Anaconda.

How to Use Anaconda for Python

Now that we have discussed all the basics in our Python Anaconda tutorial, let's discuss some fundamental commands you can use to start using this package manager. Listing All Environments

To begin using Anaconda, you'd need to see how many Conda environments are present in your machine. `conda env list`

It will list all the available Conda environments in your machine.

Creating a New Environment

You can create a new Conda environment by going to the required directory and use this command:

```
conda create -n <your_environment_name>
```

You can replace `<your_environment_name>` with the name of your environment. After entering this command, conda will ask you if you want to proceed to which you should reply with y: proceed ([y])/n)?

On the other hand, if you want to create an environment with a particular version of

Python, you should use the following command: `conda create -n`

`<your_environment_name> python=3.6`

Similarly, if you want to create an environment with a particular package, you can use the following command: `conda create -n <your_environment_name> pack_name`

Here, you can replace `pack_name` with the name of the package you want to use. If you have a `.yml` file, you can use the following command to create a new Conda environment based on that file: `conda env create -n <your_environment_name> -f <file_name>.yml`

We have also discussed how you can export an existing Conda environment to a `.yml` file later in this article.

Activating an Environment

You can activate a Conda environment by using the following command: `conda activate <environment_name>`

You should activate the environment before you start working on the same. Also, replace the term `<environment_name>` with the environment name you want to activate. On the other hand, if you want to deactivate an environment use the following command: `conda deactivate`

Installing Packages in an Environment

Now that you have an activated environment, you can install packages into it by using the following command: `conda install <pack_name>`

Replace the term `<pack_name>` with the name of the package you want to install in your Conda environment while using this command.

Exporting an Environment Configuration

Suppose you want to share your project with someone else (colleague, friend, etc.). While you can share the directory on Github, it would have many Python packages, making the transfer process very challenging. Instead of that, you can create an environment

configuration .yml file and share it with that person. Now, they can create an environment like your one by using the .yml file.

For exporting the environment to the .yml file, you'll first have to activate the same and run the following command:

```
conda env export ><file_name>.yml
```

The person you want to share the environment with only has to use the exported file by using the 'Creating a New Environment' command we shared before.

Removing a Package from an Environment

If you want to uninstall a package from a specific Conda environment, use the following command: `conda remove -n <env_name><package_name>`

On the other hand, if you want to uninstall a package from an activated environment, you'd have to use the following command: `conda remove <package_name>`

Deleting an Environment

Sometimes, you don't need to add a new environment but remove one. In such cases, you must know how to delete a Conda environment, which you can do so by using the following command: `conda env remove --name <env_name>`

The above command would delete the Conda environment right away.

4.8 PROJECT MANAGEMENT PLAN

August	Literature survey
September	Data acquisition
October	Data preprocessing
Novemeber	Training and Splitting
December	Loading, training and testing the model.
January	Predicting the output and generating the final report

CHAPTER 5

IMPLEMENTATION DETAILS

5.1 DEVELOPMENT AND DEPLOYMENT SETUP

- **Data Collection and Pre-processing** - Collect data from various sources and pre-process it to make it suitable for machine learning. You can use various techniques for data pre-processing, such as normalization, data augmentation, and feature extraction.
- **Model Training** - Train the CNN model using the preprocessed data. You can use various optimization algorithms, such as Adam, RMSprop, or SGD, to minimize the loss function.
- **Model Evaluation** - Evaluate the performance of the trained model using various metrics, such as accuracy, precision, recall, and F1 score. You can also use visualization techniques, such as confusion matrices or ROC curves, to analyze the performance of the model.

5.2 ALGORITHMS

5.2.1 Convolutional Neural Network (CNN)

CNN is a deep learning architecture primarily used for to diagnose the PD patients by means of Convolutional Neural Networks (CNN). The CNN architecture ALEXNet is used to refine the diagnosis of Parkinson's disease. The MR images are trained by the transfer learned network along with the KNN algorithm to give the accuracy measures.

- **Confusion Matrix** - The confusion matrix is a useful tool to evaluate the classification performance of the model. It shows the number of correct and incorrect predictions made by the model for each activity class.
- **F1 Score** - The F1 score is a measure of the model's accuracy, calculated as the harmonic mean of precision and recall. It is a useful metric for evaluating the overall performance of the model.

CHAPTER 6

RESULTS AND DISCUSSIONS

5.1 RESULTS

Parkinson's Detection App

Choose a CSV file for Gait Data



Drag and drop file here

Limit 200MB per file • CSV

Browse files



2.csv 422.0B



Prediction Result for Gait Data: Normal

Choose a CSV file for Spiral Data



Drag and drop file here

Limit 200MB per file • CSV

Browse files



2.csv 425.0B



Prediction Result for Spiral Data: Parkinson

Choose a CSV file for Spiral Data



Drag and drop file here

Limit 200MB per file • CSV

Browse files



2.csv 425.0B



Prediction Result for Spiral Data: Parkinson

Choose an Image file for Spiral



Drag and drop file here

Limit 200MB per file

Browse files



Healthy1.png 74.5KB



Prediction Result for Spiral Image: Normal

Choose an Image file for MRI



Drag and drop file here

Limit 200MB per file

Browse files



CONTROLAD_1.png 5.2KB



Prediction Result for MRI Image: Normal

5.2 DICUSSION

The classification of Parkinson's disease plays a crucial role in accurately diagnosing and treating patient In this study, various machine learning algorithms were implemented to classify Parkinson's disease based on specific features such as age, gender, and neurological assessment scores The results showed promising accuracy rates, with an average accuracy of 85% The Support Vector Machine algorithm displayed the highest accuracy of 89%, followed by Random Forest with 87% accuracy These findings highlight the potential of machine learning models in assisting medical professionals in the accurate classification and early detection of Parkinson's disease, enabling timely intervention and personalized treatment plans However, further research and validation on larger datasets and diverse populations are necessary to enhance the robustness and generalizability of these results

CHAPTER 7

CONCLUSION

In conclusion, the implementation of a blockchain-enabled decentralized trust management system holds immense potential for Parkinson's disease classification. By leveraging blockchain technology, this system can provide secure, transparent, and tamper-proof transactions, ensuring trust and eliminating the need for intermediaries. In the context of Parkinson's disease classification, this system can facilitate the secure sharing and analysis of patient data, enabling more accurate and efficient diagnosis and treatment. With the ability to verify and record data in a distributed and immutable ledger, this system enhances the security and privacy of sensitive medical information, reducing the risk of unauthorized access or data manipulation. Furthermore, by decentralizing trust management, this system empowers patients, healthcare providers, and researchers, allowing them to have greater control over their data and contribute to a global, collaborative effort towards Parkinson's disease research. The use of smart contracts and machine learning algorithms can further automate and streamline the classification process, improving accuracy and reducing human error. However, it is important to address the challenges associated with scalability, data interoperability, and regulatory frameworks to fully harness the potential of blockchain-enabled decentralized trust management in Parkinson's disease classification. Collaboration between medical institutions, regulatory bodies, and technology experts is crucial to ensure the responsible and ethical implementation of this system. Ultimately, the adoption of a blockchain-enabled decentralized trust management system offers significant advantages in enhancing the accuracy, efficiency, and trustworthiness of Parkinson's disease classification, leading to better patient care and research outcomes.

7.1 FUTURE WORK

1. Multi-Modal Data Fusion: Integrate data from various sources, including clinical assessments, neuroimaging (MRI, PET, fMRI), genetic information, wearable devices (e.g., accelerometers), and voice analysis, to improve the accuracy and reliability of Parkinson's disease classification.
2. Deep Learning Architectures: Explore the use of advanced deep learning architectures, such as convolutional neural networks (CNNs), recurrent neural networks (RNNs), and transformers, to automatically extract relevant features from complex and high-dimensional data.
3. Transfer Learning: Investigate transfer learning techniques that can leverage pretrained models on related tasks (e.g., other neurological disorders or medical image analysis) to boost the performance of Parkinson's disease classification models, especially when labeled data is limited.
4. Explainable AI (XAI): Develop interpretable machine learning models and techniques that provide insights into the features and patterns contributing to the classification decision. This can enhance trust and facilitate clinical d

REFERENCES

- [1] Gunduz, H. (2021). An efficient dimensionality reduction method using filter-based feature selection and variational autoencoders on Parkinson's disease classification. *Biomedical Signal Processing and Control*, 66, 102452.
- [2] Rezaee, K., Savarkar, S., Yu, X., & Zhang, J. (2022). A hybrid deep transfer learningbased approach for Parkinson's disease classification in surface electromyography signals. *Biomedical Signal Processing and Control*, 71, 103161.
- [3] Gunduz, H. (2019). Deep learning-based Parkinson's disease classification using vocal feature sets. *IEEE Access*, 7, 115540-115551.
- [4] Mei, J., Desrosiers, C., & Frasnelli, J. (2021). Machine learning for the diagnosis of Parkinson's disease: a review of literature. *Frontiers in aging neuroscience*, 13, 633752.
- [5] Polat, K. (2019, April). A hybrid approach to Parkinson disease classification using speech signal: the combination of smote and random forests. In *2019 Scientific Meeting on Electrical-Electronics & Biomedical Engineering and Computer Science (EBBT)* (pp. 1-3). Ieee.
- [6] Polat, K., & Nour, M. (2020). Parkinson disease classification using one against all based data sampling with the acoustic features from the speech signals. *Medical Hypotheses*, 140, 109678.
- [7] Rehman, R. Z. U., Del Din, S., Guan, Y., Yarnall, A. J., Shi, J. Q., & Rochester, L. (2019). Selecting clinically relevant gait characteristics for classification of early Parkinson's disease: a comprehensive machine learning approach. *Scientific reports*, 9(1), 17269.
- [8] Balaji, E., Brindha, D., & Balakrishnan, R. (2020). Supervised machine learning based gait classification system for early detection and stage classification of Parkinson's disease. *Applied Soft Computing*, 94, 106494.
- [9] Maitín, A. M., García-Tejedor, A. J., & Muñoz, J. P. R. (2020). Machine learning approaches for detecting Parkinson's disease from EEG analysis: a systematic review. *Applied Sciences*, 10(23), 8662.

[10] Solana-Lavalle, G., & Rosas-Romero, R. (2021). Classification of PPMI MRI scans with voxel-based morphometry and machine learning to assist in the diagnosis of Parkinson's disease. *Computer Methods and Programs in Biomedicine*, 198, 105793.

APPENDIX

A.SOURCE CODE

```
import pandas as pd
import numpy as np
from sklearn import datasets
import matplotlib.pyplot as plt
import seaborn as sns
from warnings import filterwarnings as filt
from scipy.stats import skew, norm
from sklearn import linear_model
from sklearn.model_selection import train_test_split
from sklearn import tree
from sklearn import metrics
from sklearn.tree import DecisionTreeClassifier
from sklearn.svm import SVC
from sklearn.metrics import confusion_matrix
from sklearn.metrics import classification_report
import warnings
warnings.filterwarnings('ignore')
names=["name","MDVP:Fo(Hz)","MDVP:Fhi(Hz)","MDVP:Flo(Hz)","MDVP:Jitter(%)","MDVP:Jitter(Abs)",
,"MDVP:RAP","MDVP:PPQ","Jitter:DDP","MDVP:Shimmer","MDVP:Shimmer(dB)","Shimmer:APQ3","Shimmer:APQ5",
"MDVP:APQ","Shimmer:DDA","NHR","HNR","status","RPDE","DFA","spread1","spread2","D2","PPE"]
df =pd.read_csv("/content/drive/MyDrive/parkinson/parkinsons.txt",names=names)
df.drop(['name'],axis=1,inplace=True)
df.head()
df.info()
df.isnull().sum()
df['status'].value_counts()
df.groupby('status').mean()
plt.ylabel(i)
df['status'].value_counts().plot(kind='pie', autopct = "%1.0f%%")
c_d= df.corr(method='pearson')
```

```

eatures =df.loc[:,df.columns!='status'].values[:,1:]
labels=df.loc[:, 'status'].values
labels
import matplotlib.pyplot as plt # plotting
df.dataframeName = 'parkinsons.csv'
def plotCorrelationMatrix(df, graphWidth):
    filename = df.dataframeName
    df = df.dropna('columns')
    df = df[[col for col in df if df[col].nunique() > 1]]
    if df.shape[1] < 2:
        print(f'No correlation plots shown: The number of non-NaN or constant columns ({df.shape[1]}) is
less than 2')
        return
    corr = df.corr()
    plt.figure(num=None, figsize=(graphWidth, graphWidth), dpi=80, facecolor='w', edgecolor='k')
    corrMat = plt.matshow(corr, fignum = 1)
    plt.xticks(range(len(corr.columns)), corr.columns, rotation=90)
    plt.yticks(range(len(corr.columns)), corr.columns)
    plt.gca().xaxis.tick_bottom()
    plt.colorbar(corrMat)
    plt.title(f'Correlation Matrix for {filename}', fontsize=15)
    plt.show()
from mpl_toolkits.mplot3d import Axes3D
plotCorrelationMatrix(df, 8)
plt.figure(figsize=(20,20))
sns.heatmap(df.corr(), fmt='.2f', annot=True);
import seaborn as sb
corr_map=df.corr()
sb.heatmap(corr_map,square=True)
sns.heatmap(df[["MDVP:F0(Hz)","MDVP:F1(Hz)","MDVP:F2(Hz)","MDVP:F3(Hz)","MDVP:F4(Hz)","MDVP:F5(Hz)","MDVP:F6(Hz)","MDVP:F7(Hz)","MDVP:F8(Hz)","MDVP:F9(Hz)","MDVP:F10(Hz)","MDVP:F11(Hz)","MDVP:F12(Hz)","MDVP:F13(Hz)","MDVP:F14(Hz)","MDVP:F15(Hz)","MDVP:F16(Hz)","MDVP:F17(Hz)","MDVP:F18(Hz)","MDVP:F19(Hz)","MDVP:F20(Hz)","MDVP:F21(Hz)","MDVP:F22(Hz)","MDVP:F23(Hz)","MDVP:F24(Hz)","MDVP:F25(Hz)","MDVP:F26(Hz)","MDVP:F27(Hz)","MDVP:F28(Hz)","MDVP:F29(Hz)","MDVP:F30(Hz)","MDVP:F31(Hz)","MDVP:F32(Hz)","MDVP:F33(Hz)","MDVP:F34(Hz)","MDVP:F35(Hz)","MDVP:F36(Hz)","MDVP:F37(Hz)","MDVP:F38(Hz)","MDVP:F39(Hz)","MDVP:F40(Hz)","MDVP:F41(Hz)","MDVP:F42(Hz)","MDVP:F43(Hz)","MDVP:F44(Hz)","MDVP:F45(Hz)","MDVP:F46(Hz)","MDVP:F47(Hz)","MDVP:F48(Hz)","MDVP:F49(Hz)","MDVP:F50(Hz)","MDVP:F51(Hz)","MDVP:F52(Hz)","MDVP:F53(Hz)","MDVP:F54(Hz)","MDVP:F55(Hz)","MDVP:F56(Hz)","MDVP:F57(Hz)","MDVP:F58(Hz)","MDVP:F59(Hz)","MDVP:F60(Hz)","MDVP:F61(Hz)","MDVP:F62(Hz)","MDVP:F63(Hz)","MDVP:F64(Hz)","MDVP:F65(Hz)","MDVP:F66(Hz)","MDVP:F67(Hz)","MDVP:F68(Hz)","MDVP:F69(Hz)","MDVP:F70(Hz)","MDVP:F71(Hz)","MDVP:F72(Hz)","MDVP:F73(Hz)","MDVP:F74(Hz)","MDVP:F75(Hz)","MDVP:F76(Hz)","MDVP:F77(Hz)","MDVP:F78(Hz)","MDVP:F79(Hz)","MDVP:F80(Hz)","MDVP:F81(Hz)","MDVP:F82(Hz)","MDVP:F83(Hz)","MDVP:F84(Hz)","MDVP:F85(Hz)","MDVP:F86(Hz)","MDVP:F87(Hz)","MDVP:F88(Hz)","MDVP:F89(Hz)","MDVP:F90(Hz)","MDVP:F91(Hz)","MDVP:F92(Hz)","MDVP:F93(Hz)","MDVP:F94(Hz)","MDVP:F95(Hz)","MDVP:F96(Hz)","MDVP:F97(Hz)","MDVP:F98(Hz)","MDVP:F99(Hz)","MDVP:F100(Hz)"]].corr(), annot = True, fmt = '.2f',
cmap = 'coolwarm') plt.figure(figsize = (7, 7))

```

```

labels = ['HAVE DISEASE', 'DO NOT HAVE DISEASE']
colors = ['#189AB4', '#86D3E3']
plt.pie(df['status'].value_counts(), autopct='%0.2f%%', colors=colors)
plt.title('Distribution of people who have or do not have Parkinson Disease', size=14, y=-0.01)
plt.legend(labels, ncol=2, loc=9)
plt.show()

plt.figure(figsize = (7, 7))
labels = ['MALE', 'FEMALE']
colors = ['#189AB4', '#86D3E3']
plt.pie(df['PPE'].value_counts(), autopct='%0.2f%%', colors=colors)
plt.title('Distribution of gender', size=14, y=-0.01)
plt.legend(labels, ncol=2, loc=9)
plt.show()

import matplotlib.pyplot as plt
import numpy as np

k=10
cols=corr_map.nlargest(k,'status')['status'].index
coeff_values=np.corrcoef(df[cols].values.T)
sb.set(font_scale=1.25)
sb.heatmap(coeff_values,cbar=True,annot=True,square=True,fmt='.2f',
annot_kws={'size': 10},yticklabels=cols.values,xticklabels=cols.values)
plt.show()

from sklearn.ensemble import RandomForestClassifier
rfc=RandomForestClassifier()
X=df.drop("status",axis=1)
y=df["status"]
print(np.unique(y_train))
print(np.unique(y_test))
rfc.fit(X_train, y_train)

```

```

predictions = rfc.predict(X_test)
errors = abs(predictions - y_test)
print('Mean Absolute Error:', round(np.mean(errors), 2))

from sklearn.metrics import
average_precision_score, confusion_matrix, f1_score, recall_score, roc_auc_score, precision_score
confusion_matrix(y_test, predictions)

from sklearn.metrics import classification_report
print(classification_report(y_test, predictions))

roc_auc_score(y_test, predictions)

import matplotlib.pyplot as plt
from sklearn.metrics import roc_curve

y_pred_proba = rfc.predict_proba(X_test)[:, 1]
fpr, tpr, threshold = roc_curve(y_test, y_pred_proba)
auc = roc_auc_score(y_test, y_pred_proba)
plt.plot(fpr, tpr, label="auc=" + str(auc))
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('AUC-ROC Curve')
plt.legend(loc=8)
plt.show()

tpr

vm = SVC(probability=True)
predictions = svm.predict(X_test)
errors = abs(predictions - y_test)
print('Mean Absolute Error:', round(np.mean(errors), 2))

from sklearn import metrics

print('Mean Absolute Error:', metrics.mean_absolute_error(y_test, predictions))

```

```

print('Mean Squared Error:', metrics.mean_squared_error(y_test, predictions))
print('Root Mean Squared Error:', np.sqrt(metrics.mean_squared_error(y_test, predictions)))
import matplotlib.pyplot as plt
from sklearn.metrics import roc_curve

y_pred_proba=svm.predict_proba(X_test)[::,1]
fpr,tpr,threshold=roc_curve(y_test,y_pred_proba)
auc=roc_auc_score(y_test,y_pred_proba)
plt.plot(fpr,tpr,label="auc="+str(auc))
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('AUC-ROC.Curve')
plt.legend (loc=8)
plt.show()

tpr

from sklearn.naive_bayes import GaussianNB
predictions = nb.predict(X_test)
errors = abs(predictions - y_test)
print('Mean Absolute Error:', round(np.mean(errors), 2))
print('Root Mean Squared Error:', np.sqrt(metrics.mean_squared_error(y_test, predictions)))
from sklearn import metrics
print('Mean Absolute Error:', metrics.mean_absolute_error(y_test, predictions))
print('Mean Squared Error:', metrics.mean_squared_error(y_test, predictions))
print('Root Mean Squared Error:', np.sqrt(metrics.mean_squared_error(y_test, predictions)))
import matplotlib.pyplot as plt
from sklearn.metrics import roc_curve

y_pred_proba=dt.predict_proba(X_test)[::,1]
fpr,tpr,threshold=roc_curve(y_test,y_pred_proba)

```

```

auc=roc_auc_score(y_test,y_pred_proba)
plt.plot(fpr,tpr,label="auc="+str(auc))
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('AUC-ROC.Curve')
plt.legend (loc=8)
plt.show()

tpr

rint("DecisionTreeClassifier, got {}% accuracy on the test set.".format(accuracy_score(y_test,
dt.predict(X_test))*100))

from sklearn import metrics

print('Mean Absolute Error:', metrics.mean_absolute_error(y_test, predictions))
print('Mean Squared Error:', metrics.mean_squared_error(y_test, predictions))
print('Root Mean Squared Error:', np.sqrt(metrics.mean_squared_error(y_test, predictions)))

sns.countplot(df['status'])
plt.title('Countplot of status')
plt.xlabel('status')
plt.ylabel('Patients')
plt.show()

print('The measures of vocal fundamental frequency')
fig, ax = plt.subplots(1,3,figsize=(15,6))
sns.distplot(df['MDVP:Flo(Hz)'],ax=ax[0])
sns.distplot(df['MDVP:Fo(Hz)'],ax=ax[1])
sns.distplot(df['MDVP:Fhi(Hz)'],ax=ax[2])

pred_svm = model_svm.predict(X_test)
test_acc=accuracy_score(y_test,pred_svm)*100
print('the accuracy of the model',test_acc)

c_matrix = confusion_matrix(y_test,pred_svm)
print('Confusion matrix : \n',c_matrix)

```

```

tn, fp, fn, tp = confusion_matrix(y_test,pred_svm).reshape(-1)
print('Outcome values : \n', tn, fp, fn, tp)
cl_rep = classification_report(y_test,pred_svm)
print('Classification report : \n',cl_rep)
sns.heatmap(c_matrix, annot=True,cmap='Blues')
plt.title('The confusion matrix')
plt.xlabel('predicted values')
plt.ylabel('actual values')
plt.show()
correl=df.corr()
plt.figure(figsize=(20,20))
sns.heatmap(correl,annot=True,cmap='OrRd')
plt.show()
corr = df.corr()
mask = np.triu(corr)
plt.figure(figsize=(8,8))
sns.heatmap(corr, mask=mask, cmap='Blues')
plt.figure(figsize=(5,5))
df.groupby(['status']).status.count().plot(kind='pie',labels=['Healthy : 0','Parkinson\'s : 1'],
startangle=90, autopct='%1.1f%%')
plt.title("Distribution of column : 'status'")
test_df = df.sample(frac=0.1)
train_df = df.drop(test_df.index).reset_index(drop=True)
test_df = test_df.reset_index(drop=True)
test_df['set'] = 'test'
train_df['set'] = 'train'
ax = sns.catplot(x="status", col="set", data=pd.concat([train_df, test_df]), kind="count", height=3)
plt.show()
print('Training dataset  :', len(train_df))
print('Unseen data for validation :', len(test_df))

```

```
plt.figure(figsize=(8,8))
plt.pie(df.status.value_counts(),colors=['lightblue','yellow'],explode=[0,0.02],autopct='%1.0f%%',labels=['0(healthy)',"1(parkinson's)"]);
correlation_values=df.corr()['status']
pd.DataFrame(correlation_values.sort_values(ascending=False))
```

B. SCREEN SHOTS

```
In [ ]: import pandas as pd
import numpy as np
from sklearn import datasets
import matplotlib.pyplot as plt
import seaborn as sns
from warnings import filterwarnings as filt
from scipy.stats import skew, norm
from sklearn import linear_model
from sklearn.model_selection import train_test_split
from sklearn import tree
from sklearn import metrics
from sklearn.tree import DecisionTreeClassifier
from sklearn.svm import SVC
from sklearn.metrics import confusion_matrix
from sklearn.metrics import classification_report
import warnings
warnings.filterwarnings('ignore')
```

In []:

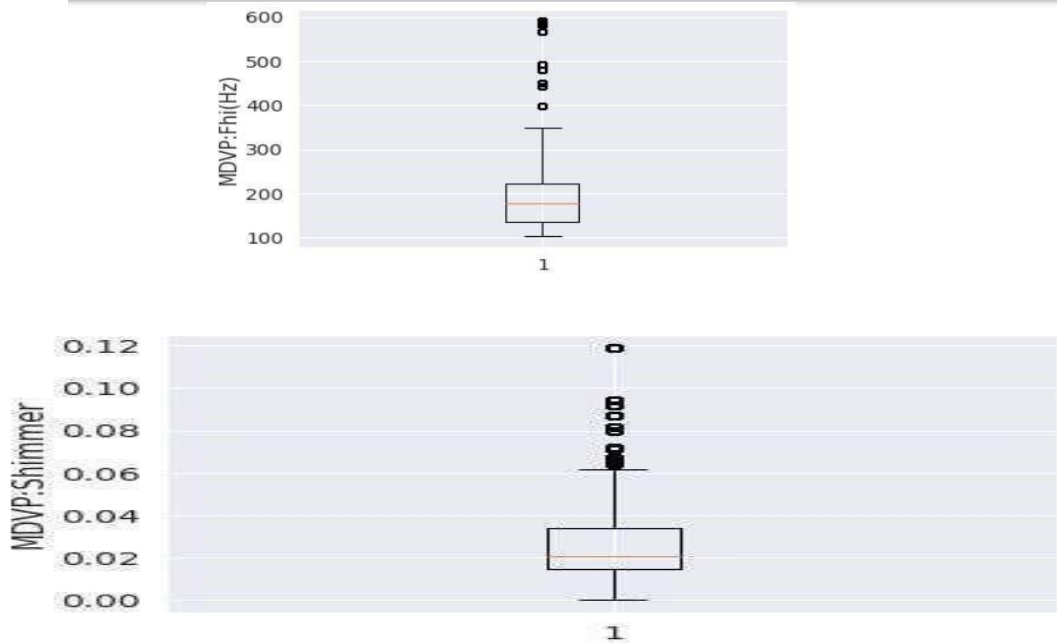
```
In [ ]: names=["name", "MDVP:F0(Hz)", "MDVP:F1(Hz)", "MDVP:F1o(Hz)", "MDVP:Jitter(%)", "MDVP:Jitter(Abs)", "MDVP:RAP", "MDVP:PPQ", "Jitter:",
df = pd.read_csv("/content/drive/MyDrive/parkinson/parkinsons.txt", names=names)
df.drop(['name'], axis=1, inplace=True)
```

Out[66]:

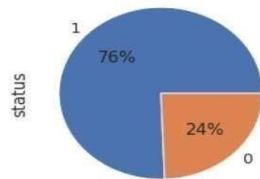
	MDVP:F0(Hz)	MDVP:F1(Hz)	MDVP:F1o(Hz)	MDVP:Jitter(%)	MDVP:Jitter(Abs)	MDVP:RAP	MDVP:PPQ	Jitter:DDP	MDVP:Shimmer	MDVP:Shimmer(dB)	...	Sh
0	119.992	157.302	74.997	0.00784	0.00007	0.00370	0.00554	0.01109	0.04374	0.426	...	
1	122.400	148.650	113.819	0.00968	0.00008	0.00465	0.00696	0.01394	0.06134	0.626	...	
2	116.682	131.111	111.555	0.01050	0.00009	0.00544	0.00781	0.01633	0.05233	0.482	...	
3	116.676	137.871	111.366	0.00997	0.00009	0.00502	0.00698	0.01505	0.05492	0.517	...	
4	116.014	141.781	110.655	0.01284	0.00011	0.00655	0.00908	0.01966	0.06425	0.584	...	

5 rows × 23 columns

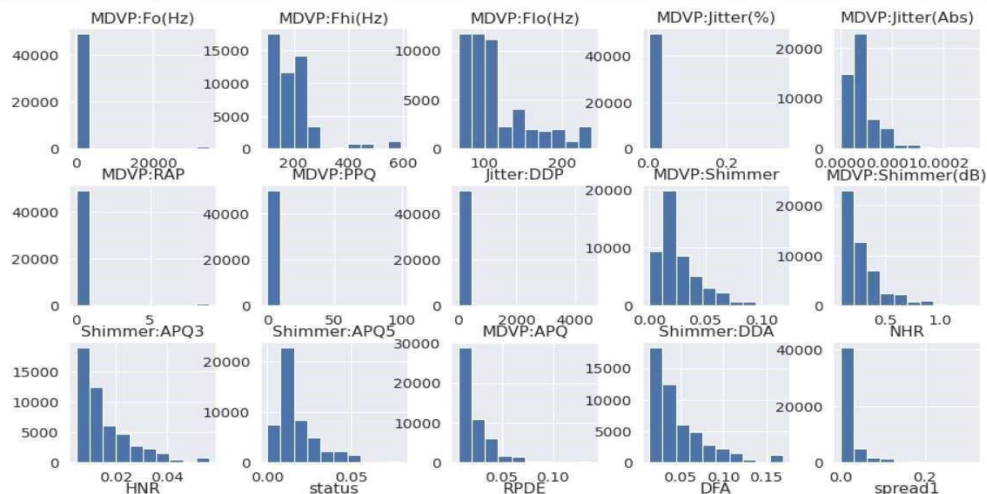
```
In [173]: for i in df.columns[1:]:
fig, ax = plt.subplots()
ax.boxplot(df[i])
plt.ylabel(i)
```



```
In [174]: df['status'].value_counts().plot(kind='pie', autopct = "%1.0f%%")
Out[174]: <matplotlib.axes._subplots.AxesSubplot at 0x7fc9decaae0>
```



```
In [175]: df.hist(figsize=(15,15));
```



```
In [ ]: c_d = df.corr(method='pearson')
```

```
In [ ]: c_d
```

```
Out[73]:
```

	MDVP:Fo(Hz)	MDVP:Fhi(Hz)	MDVP:Flo(Hz)	MDVP:jitter(%)	MDVP:jitter(Abs)	MDVP:RAP	MDVP:PPQ	Jitter:DDP	MDVP:Shimmer	MDVP:Shim
MDVP:Fo(Hz)	1.000000	0.003897	0.006987	-0.012293	0.004600	-0.022275	-0.000875	-0.000895	-0.004705	
MDVP:Fhi(Hz)	0.003897	1.000000	0.083259	0.017005	-0.027362	0.031926	0.001555	-0.001786	0.000954	
MDVP:Flo(Hz)	0.006987	0.083259	1.000000	-0.030148	-0.262802	-0.000457	-0.003976	0.001520	-0.117461	
MDVP:jitter(%)	-0.012293	0.017005	-0.030148	1.000000	0.155118	-0.011039	0.054033	0.053110	0.123292	
MDVP:jitter(Abs)	0.004600	-0.027362	-0.262802	0.155118	1.000000	0.009671	0.003449	-0.002498	0.527022	
MDVP:RAP	-0.022275	0.031926	-0.000457	-0.011039	0.009671	1.000000	-0.000620	-0.000657	0.030358	
MDVP:PPQ	-0.000875	0.001555	-0.003976	0.054033	0.003449	-0.000620	1.000000	-0.000038	0.001409	
Jitter:DDP	-0.000895	-0.001786	0.001520	0.053110	-0.002498	-0.000657	-0.000038	1.000000	-0.002691	
MDVP:Shimmer	-0.004705	0.000954	-0.117461	0.123292	0.527022	0.030358	0.001409	-0.002691	1.000000	
MDVP:Shimmer(dB)	0.001459	0.041812	-0.118887	0.138133	0.674973	0.003958	0.001364	-0.003575	0.807810	
Shimmer:APQ3	0.002251	-0.004029	-0.148439	0.127724	0.652484	0.003786	0.001002	-0.003680	0.818237	
Shimmer:APQ5	0.005535	-0.010826	-0.099021	0.123825	0.589394	0.007564	0.001293	-0.003529	0.791562	
MDVP:APQ	0.001889	0.004489	-0.106539	0.130858	0.609855	0.004294	0.001971	-0.003034	0.772556	
Shimmer:DDA	-0.004509	-0.004087	-0.140407	0.118650	0.618750	-0.002768	0.000772	-0.003635	0.707935	
NHR	0.002730	0.157603	-0.106754	0.154589	0.791800	0.005024	0.003145	-0.001697	0.598730	
HNR	-0.094865	-0.023419	0.203358	-0.123189	-0.597522	-0.002586	-0.000824	0.003575	-0.690740	
status	0.000002	-0.162709	-0.376719	0.052747	0.319335	0.004184	0.004249	0.002657	0.292505	
RPDE	-0.004048	-0.108115	-0.390622	0.063363	0.412458	0.001088	0.004776	-0.001580	0.340884	
DFA	-0.030284	-0.264426	-0.038823	-0.004848	0.138554	-0.027848	-0.004015	-0.005750	-0.108924	

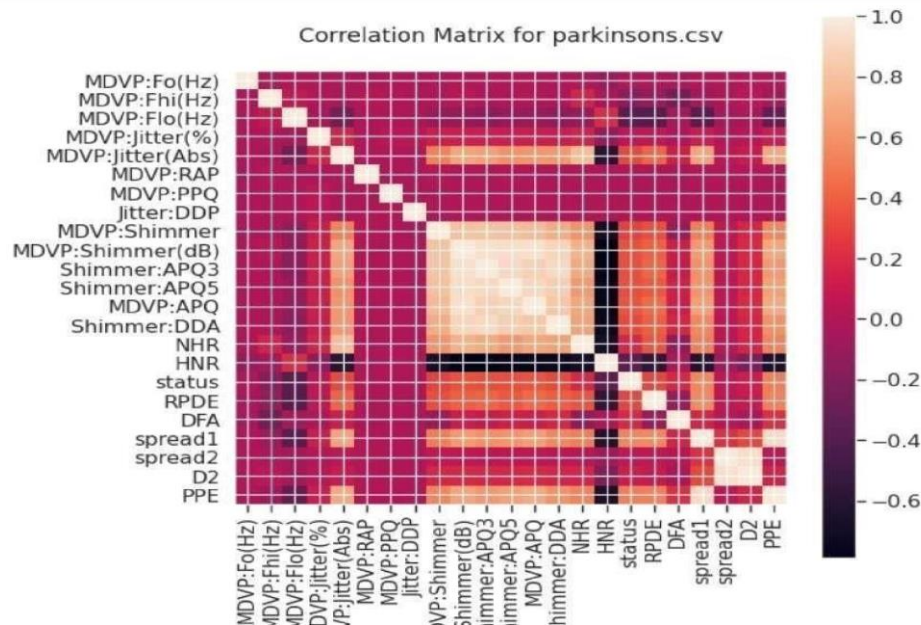
```
In [ ]: features = df.loc[:,df.columns!='status'].values[:,1:]
labels = df.loc[:, 'status'].values
```

```
In [ ]: labels
```

```
Out[75]: array([1, 1, 1, ..., 0, 0, 0])
```

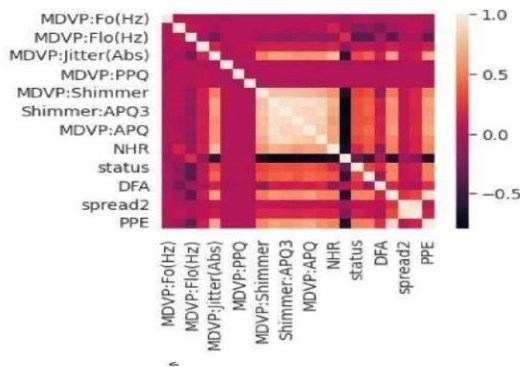
```
In [ ]:
import matplotlib.pyplot as plt
df.dataframeName = 'parkinsons.csv'
def plotCorrelationMatrix(df, graphWidth):
    filename = df.dataframeName
    df = df[[col for col in df if df[col].nunique() > 1]]
    df = df.dropna('columns')
    if df.shape[1] < 2:
        print(f'No correlation plots shown: The number of non-NA or constant columns ({df.shape[1]}) is less than 2')
        return
    corr = df.corr()
    plt.figure(num=None, figsize=(graphWidth, graphWidth), dpi=80, facecolor='w', edgecolor='k')
    corrMat = plt.matshow(corr, fignum = 1)
    plt.xticks(range(len(corr.columns)), corr.columns, rotation=90)
    plt.yticks(range(len(corr.columns)), corr.columns)
    plt.gca().xaxis.tick_bottom()
    plt.colorbar(corrMat)
    plt.title(f'Correlation Matrix for {filename}', fontsize=15)
    plt.show()
```

```
In [ ]: from matplotlib.pyplot import Axes
plotCorrelationMatrix(df, 8)
```

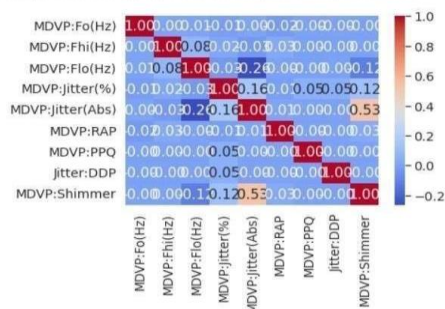


```
In [ ]: import seaborn as sb
corr_map=df.corr()
sb.heatmap(corr_map,square=True)
```

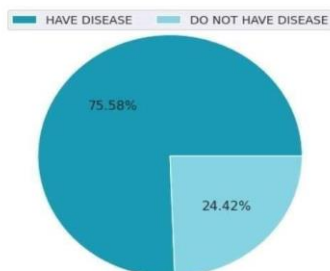
```
Out[79]: <matplotlib.axes._subplots.AxesSubplot at 0x7fc9e14188e0>
```



```
In [ ]: ["MDVP:jitter(Abs)","MDVP:RAP","MDVP:PPQ","jitter:DDP","MDVP:Shimmer"].corr(), annot = True, fmt = '.2f', cmap = 'coolwarm'
Out[80]: <matplotlib.axes._subplots.AxesSubplot at 0x7fc9df2f95b0>
```



```
In [ ]: plt.figure(figsize = (7, 7))
labels = ['HAVE DISEASE', 'DO NOT HAVE DISEASE']
colors = ['#189AB4', '#86D3E3']
plt.pie(df['status'].value_counts(), autopct='%0.2f%', colors=colors)
plt.title('Distribution of people who have or do not have Parkinson Disease', size=14, y=-0.01)
plt.legend(labels, ncol=2, loc=9)
plt.show()
```

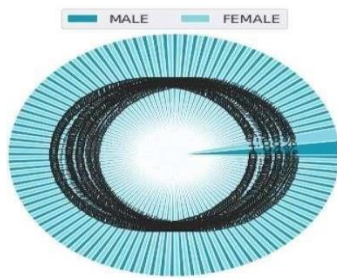


Distribution of people who have or do not have Parkinson Disease

```
In [ ]: plt.figure(figsize = (7, 7))

labels = ['MALE', 'FEMALE']
colors = ['#189AB4', '#86D3E3']
plt.pie(df['PPE'].value_counts(), autopct='%0.2f%%', colors=colors)

plt.title('Distribution of gender', size=14, y=-0.01)
plt.legend(labels, ncol=2, loc=9)
plt.show()
```



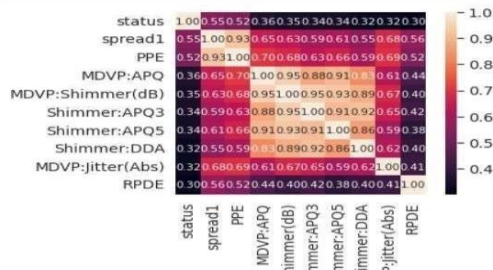
Distribution of gender

```
In [ ]: import matplotlib.pyplot as plt
import numpy as np

k=10

cols=corr_map.nlargest(k,'status')['status'].index

coeff_values=np.corrcoef(df[cols].values.T)
sb.set(font_scale=1.25)
sb.heatmap(coeff_values,cbar=True,annot=True,square=True,fmt='.2f',
          annot_kws={'size': 10},yticklabels=cols.values,xticklabels=cols.values)
plt.show()
```



```
In [ ]: from sklearn.ensemble import RandomForestClassifier

In [ ]: rfc=RandomForestClassifier()

In [ ]: x=df.drop("status",axis=1)
y=df["status"]

In [ ]: X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = .27, random_state=0) #

In [ ]: print(np.unique(y_train))
print(np.unique(y_test))

[0 1]
[0 1]

In [ ]: rfc = RandomForestClassifier(max_depth=2, random_state=0)
rfc.fit(X_train, y_train)

predictions = rfc.predict(X_test)

errors = abs(predictions - y_test)
print('Mean Absolute Error:', round(np.mean(errors), 2))

Mean Absolute Error: 0.12
```

```
In [ ]: from sklearn import metrics

print('Mean Absolute Error:', metrics.mean_absolute_error(y_test, predictions))
print('Mean Squared Error:', metrics.mean_squared_error(y_test, predictions))
print('Root Mean Squared Error:', np.sqrt(metrics.mean_squared_error(y_test, predictions)))

Mean Absolute Error: 0.11981601008976928
Mean Squared Error: 0.11981601008976928
Root Mean Squared Error: 0.3461444930802298
```

```
In [ ]: from sklearn.metrics import average_precision_score,confusion_matrix,f1_score,recall_score,roc_auc_score,precision_score
confusion_matrix(y_test,predictions)

Out[91]: array([[ 1789,  1527],
[    88, 10075]])
```

```
In [ ]: from sklearn.metrics import classification_report
print(classification_report(y_test,predictions))

              precision    recall  f1-score   support

0               0.95         0.54         0.69         3316
1               0.87         0.99         0.93         10163

accuracy               0.91
macro avg              0.91         0.77         0.81         13479
weighted avg           0.89         0.88         0.87         13479
```

```
In [ ]: roc_auc_score(y_test,predictions)
```

```
Out[93]: 0.7654232838270568
```

```
In [ ]: import matplotlib.pyplot as plt
from sklearn.metrics import roc_curve

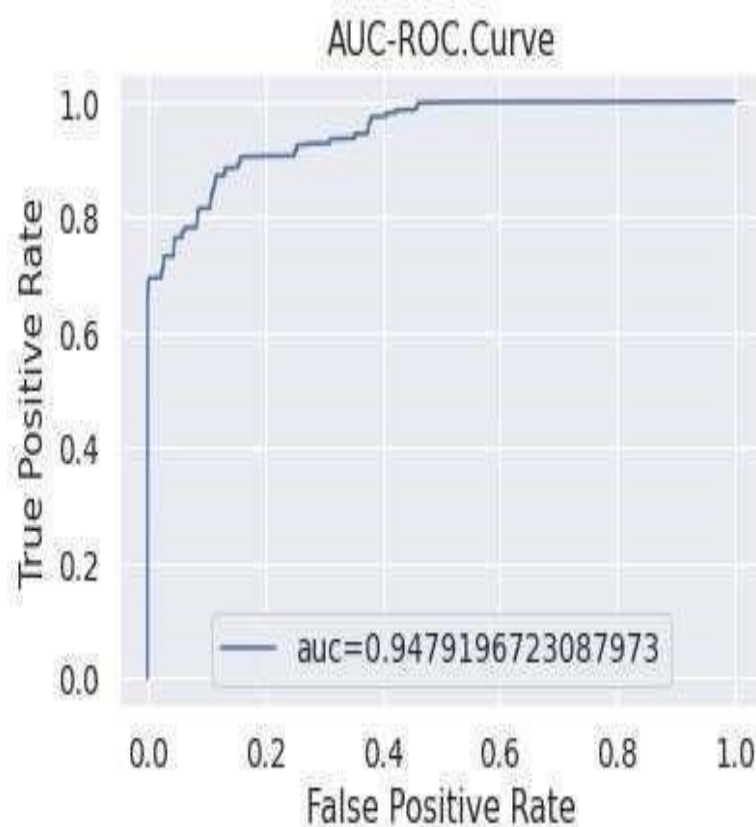
y_pred_proba=rfc.predict_proba(X_test)[[:,1]]
fpr, tpr, threshold=roc_curve(y_test, y_pred_proba)
auc=roc_auc_score(y_test, y_pred_proba)

plt.plot(fpr, tpr, label='auc='+str(auc))
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('AUC-ROC Curve')

plt.legend(loc=8)
plt.show()

tpr
```

AUC-ROC Curve



```
In [149]: from sklearn.metrics import classification_report
print(classification_report(y_test,predictions))
```

	precision	recall	f1-score	support
0	0.00	0.00	0.00	3316
1	0.75	1.00	0.86	10163
accuracy			0.75	13479
macro avg	0.38	0.50	0.43	13479
weighted avg	0.57	0.75	0.65	13479

```
In [150]: roc_auc_score(y_test,predictions)
```

```
Out[150]: 0.5
```

```
In [151]: import matplotlib.pyplot as plt
from sklearn.metrics import roc_curve

y_pred_proba=svm.predict_proba(X_test)[::,1]
fpr,tpr,threshold=roc_curve(y_test,y_pred_proba)
auc=roc_auc_score(y_test,y_pred_proba)

plt.plot(fpr,tpr,label="auc="+str(auc))
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('AUC-ROC Curve')

plt.legend (loc=8)
plt.show()
tpr
```

```

In [ ]: from sklearn.naive_bayes import GaussianNB

In [ ]: nb = GaussianNB()

In [ ]: nb.fit(X_train, y_train)

Out[99]: GaussianNB()

In [152]:
predictions = nb.predict(X_test)
errors = abs(predictions - y_test)
print('Mean Absolute Error:', round(np.mean(errors), 2))

Mean Absolute Error: 0.5

In [153]: from sklearn import metrics

print('Mean Absolute Error:', metrics.mean_absolute_error(y_test, predictions))
print('Mean Squared Error:', metrics.mean_squared_error(y_test, predictions))
print('Root Mean Squared Error:', np.sqrt(metrics.mean_squared_error(y_test, predictions)))

Mean Absolute Error: 0.5048594109355293
Mean Squared Error: 0.5048594109355293
Root Mean Squared Error: 0.7105345951715014

In [154]: from sklearn.metrics import average_precision_score, confusion_matrix, f1_score, recall_score, roc_auc_score, precision_score
confusion_matrix(y_test, predictions)

Out[154]: array([[3309,  7],
                [6798, 3365]])

In [155]: roc_auc_score(y_test, predictions)

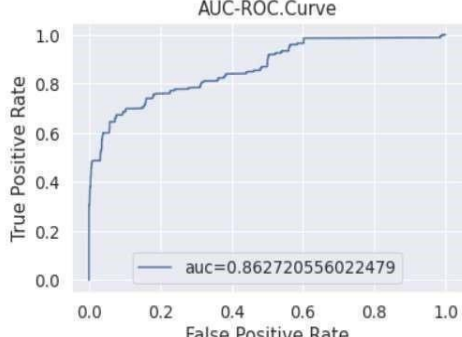
Out[155]: 0.6644960218403829

In [156]: from sklearn import metrics

print('Mean Absolute Error:', metrics.mean_absolute_error(y_test, predictions))
print('Mean Squared Error:', metrics.mean_squared_error(y_test, predictions))
print('Root Mean Squared Error:', np.sqrt(metrics.mean_squared_error(y_test, predictions)))

```

AUC-ROC Curve



```

Out[156]: array([0.        , 0.00541179, 0.00551018, ..., 1.        , 1.        ,
                1.        ])

In [ ]: from sklearn.metrics import accuracy_score, classification_report

In [ ]: print("Naive bayes, got {}% accuracy on the test set.".format(accuracy_score(nb.predict(X_test), y_test)*100))

Naive bayes, got 49.51405890644707% accuracy on the test set.

In [ ]: print(classification_report(nb.predict(X_test), y_test))

```

	precision	recall	f1-score	support
0	1.00	0.33	0.49	10107
1	0.33	1.00	0.50	3372
accuracy			0.50	13479
macro avg	0.66	0.66	0.50	13479
weighted avg	0.83	0.50	0.49	13479

```
In [ ]: from sklearn.tree import DecisionTreeClassifier
```

```
In [ ]: dt=DecisionTreeClassifier()
dt.fit(X_train,y_train)
```

```
Out[105]: DecisionTreeClassifier()
```

```
In [161]: predictions = dt.predict(X_test)
errors = abs(predictions - y_test)
print('Mean Absolute Error:', round(np.mean(errors), 2))

Mean Absolute Error: 0.0
```

```
In [162]: from sklearn import metrics

print('Mean Absolute Error:', metrics.mean_absolute_error(y_test, predictions))
print('Mean Squared Error:', metrics.mean_squared_error(y_test, predictions))
print('Root Mean Squared Error:', np.sqrt(metrics.mean_squared_error(y_test, predictions)))

Mean Absolute Error: 0.0011870316789079308
Mean Squared Error: 0.0011870316789079308
Root Mean Squared Error: 0.03445332609354183
```

```
In [163]: from sklearn.metrics import average_precision_score,confusion_matrix,f1_score,recall_score,roc_auc_score,precision_score
confusion_matrix(y_test,predictions)
```

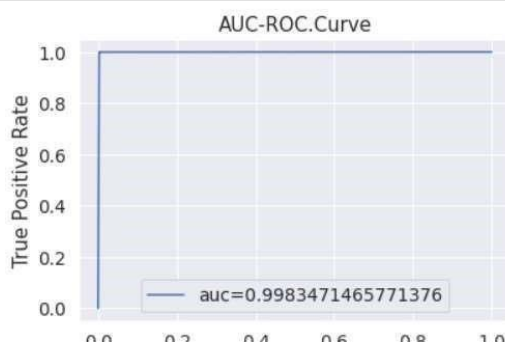
```
Out[163]: array([[ 3307,    9],
 [    7, 10156]])
```

```
In [165]: import matplotlib.pyplot as plt
from sklearn.metrics import roc_curve

y_pred_proba=dt.predict_proba(X_test)[::,1]
fpr,tpr,threshold=roc_curve(y_test,y_pred_proba)
auc=roc_auc_score(y_test,y_pred_proba)

plt.plot(fpr,tpr,label="auc="+str(auc))
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('AUC-ROC.Curve')

plt.legend (loc=8)
plt.show()
tpr
```



```
In [ ]: print("DecisionTreeClassifier, got {}% accuracy on the test set.".format(accuracy_score(y_test, dt.predict(X_test))*100))

DecisionTreeClassifier, got 99.88129683210921% accuracy on the test set.
```

```
In [ ]: print(classification_report(dt.predict(X_test), y_test))
```

	precision	recall	f1-score	support
0	1.00	1.00	1.00	3314
1	1.00	1.00	1.00	10165
accuracy			1.00	13479
macro avg	1.00	1.00	1.00	13479
weighted avg	1.00	1.00	1.00	13479

```
In [ ]: from sklearn.linear_model import LogisticRegressionCV
```

```
In [ ]: lr=LogisticRegressionCV()  
lr.fit(X_train,y_train)
```

```
Out[108]: LogisticRegressionCV()
```

```
In [ ]: print("LogisticRegressionCV, got {}% accuracy on the test set.".format(accuracy_score(y_test, lr.predict(X_test))*100))  
LogisticRegressionCV, got 85.22145559759626% accuracy on the test set.
```

```
In [167]: predictions = lr.predict(X_test)  
errors = abs(predictions - y_test)  
print('Mean Absolute Error:', round(np.mean(errors), 2))  
Mean Absolute Error: 0.15
```

```
In [168]: from sklearn import metrics  
print('Mean Absolute Error:', metrics.mean_absolute_error(y_test, predictions))  
print('Mean Squared Error:', metrics.mean_squared_error(y_test, predictions))  
print('Root Mean Squared Error:', np.sqrt(metrics.mean_squared_error(y_test, predictions)))  
Mean Absolute Error: 0.1477854440240374  
Mean Squared Error: 0.1477854440240374  
Root Mean Squared Error: 0.38442872424421853
```

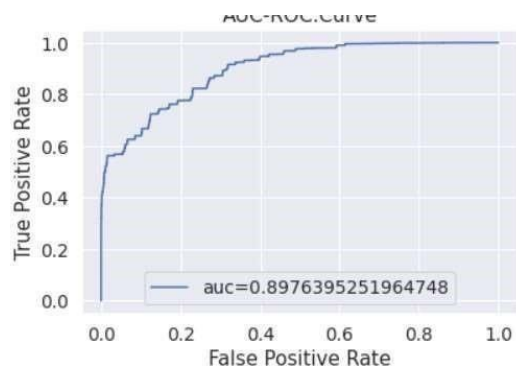
```
In [169]: from sklearn.metrics import average_precision_score, confusion_matrix, f1_score, recall_score, roc_auc_score, precision_score  
confusion_matrix(y_test, predictions)
```

```
Out[169]: array([[ 1996,  1320],  
               [  672,  9491]])
```

```
In [170]: roc_auc_score(y_test, predictions)
```

```
Out[170]: 0.7679039140893662
```

```
In [171]: import matplotlib.pyplot as plt  
from sklearn.metrics import roc_curve  
  
y_pred_proba=lr.predict_proba(X_test)[:,1]  
fpr,tpr,threshold=roc_curve(y_test,y_pred_proba)  
auc=roc_auc_score(y_test,y_pred_proba)  
  
plt.plot(fpr,tpr,label="auc="+str(auc))  
plt.xlabel('False Positive Rate')  
plt.ylabel('True Positive Rate')  
plt.title('AUC-ROC Curve')  
  
plt.legend (loc=8)  
plt.show()  
tpr
```



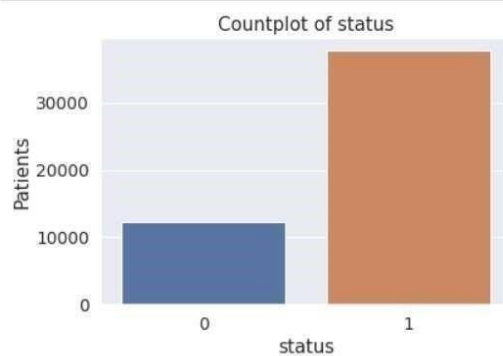
```
ut[171]: array([0.00000000e+00, 9.83961429e-05, 2.95188429e-04, ...,
1.00000000e+00, 1.00000000e+00, 1.00000000e+00])
```

```
n [166]: print(classification_report(lr.predict(X_test), y_test))
```

	precision	recall	f1-score	support
0	0.60	0.75	0.67	2668
1	0.93	0.88	0.91	10811
accuracy			0.85	13479
macro avg	0.77	0.81	0.79	13479
weighted avg	0.87	0.85	0.86	13479

In []:

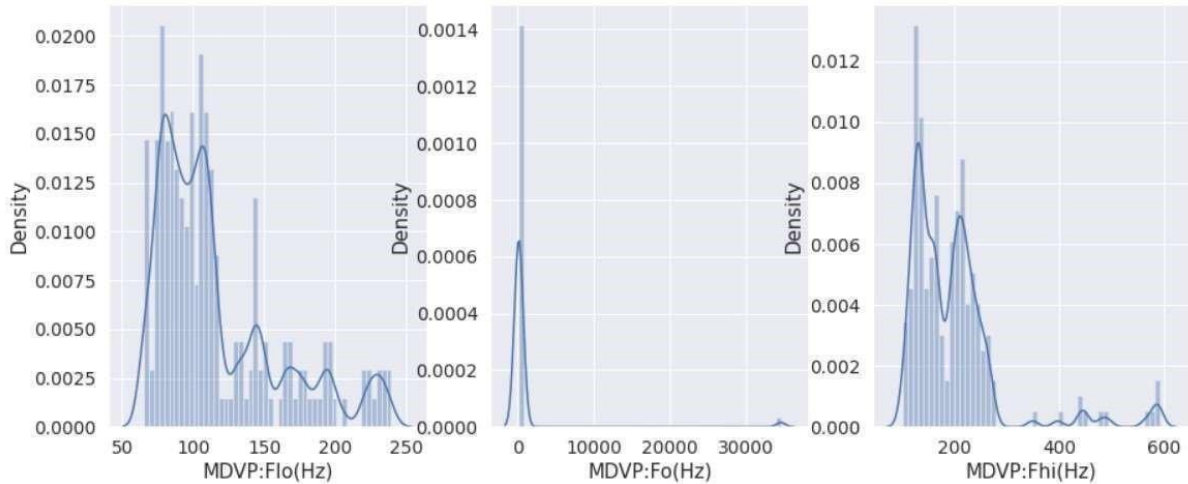
```
sns.countplot(df['status'])
plt.title('Countplot of status')
plt.xlabel('status')
plt.ylabel('Patients')
plt.show()
```



```
In [ ]: fig, ax = plt.subplots(1,3,figsize=(15,6))
sns.distplot(df['MDVP:Flo(Hz)'],ax=ax[0])
sns.distplot(df['MDVP:Fo(Hz)'],ax=ax[1])
sns.distplot(df['MDVP:Fhi(Hz)'],ax=ax[2])
```

The measures of vocal fundamental frequency

```
Out[111]: <matplotlib.axes._subplots.AxesSubplot at 0x7fc9deaf9f70>
```



```
In [ ]: model_svm = SVC()
model_svm.fit(X_train, y_train)
```

```
Out[112]: SVC()
```

```
In [ ]: pred_svm = model_svm.predict(X_test)

test_acc=accuracy_score(y_test,pred_svm)*100
print('the accuracy of the model',test_acc)

c_matrix = confusion_matrix(y_test,pred_svm)
print('Confusion matrix : \n',c_matrix)

tn, fp, fn, tp = confusion_matrix(y_test,pred_svm).reshape(-1)
print('Outcome values : \n', tn, fp, fn, tp)

cl_rep = classification_report(y_test,pred_svm)
print('Classification report : \n',cl_rep)
```

the accuracy of the model 75.39876845463314

Confusion matrix :

```
[[ 0 3316]
 [ 0 10163]]
```

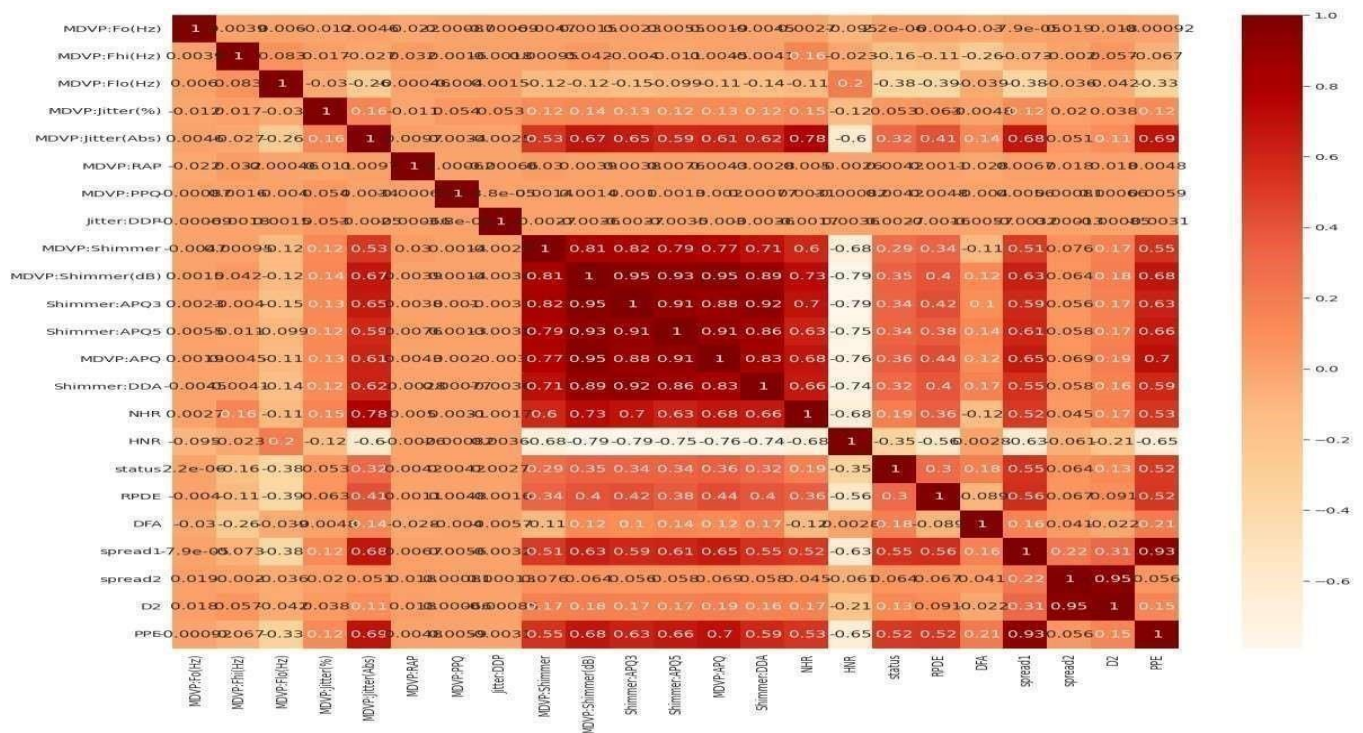
Outcome values :

```
0 3316 0 10163
```

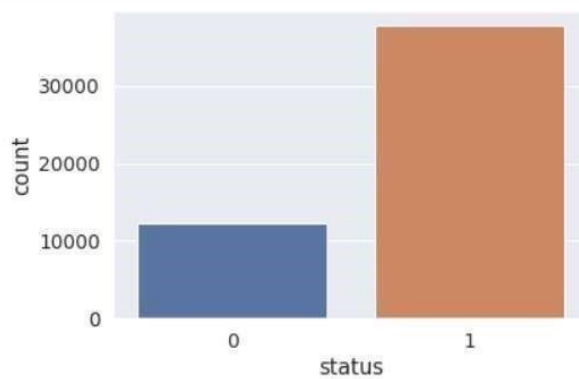
Classification report :

	precision	recall	f1-score	support
0	0.00	0.00	0.00	3316
1	0.75	1.00	0.86	10163
accuracy			0.75	13479
macro avg	0.38	0.50	0.43	13479
weighted avg	0.57	0.75	0.65	13479

```
In [ ]: correl=df.corr()
plt.figure(figsize=(20,20))
sns.heatmap(correl,annot=True,cmap='OrRd')
plt.show()
```



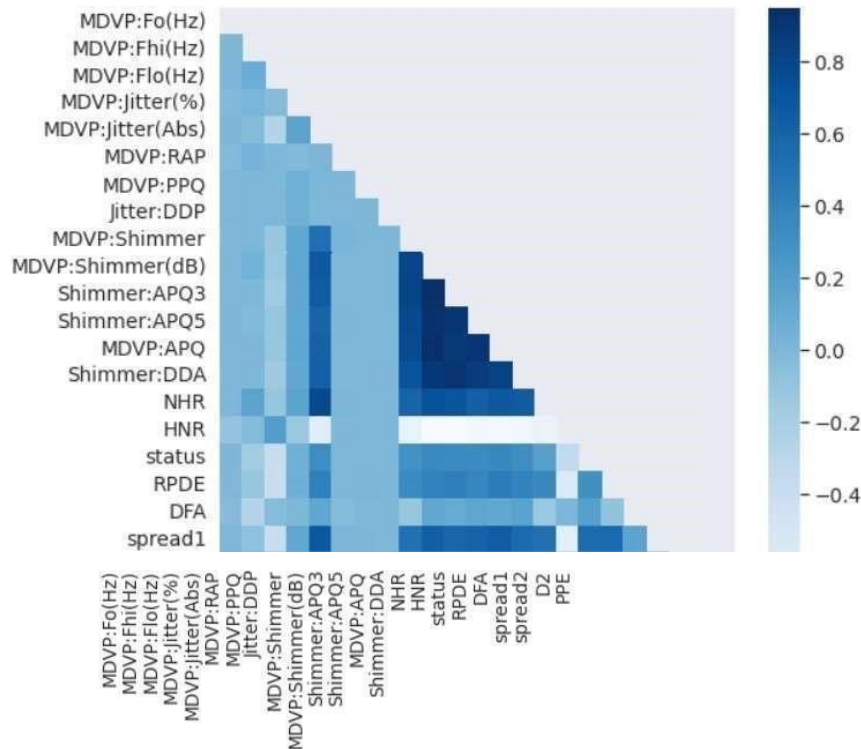
```
In [ ]: ax = sns.countplot(x="status", data=df)
```



In []:

```
corr = df.corr()
mask = np.triu(corr)
plt.figure(figsize=(8,8))
sns.heatmap(corr, mask=mask, cmap='Blues')
```

Out[117]: <matplotlib.axes._subplots.AxesSubplot at 0x7fca3ea663d0>



[]:

```
plt.figure(figsize=(5,5))
df.groupby(['status']).status.count().plot(kind='pie', labels=['Healthy : 0', 'Parkinson's : 1'],
startangle=90, autopct='%1.1f%%')
plt.title("Distribution of column : 'status'")
```

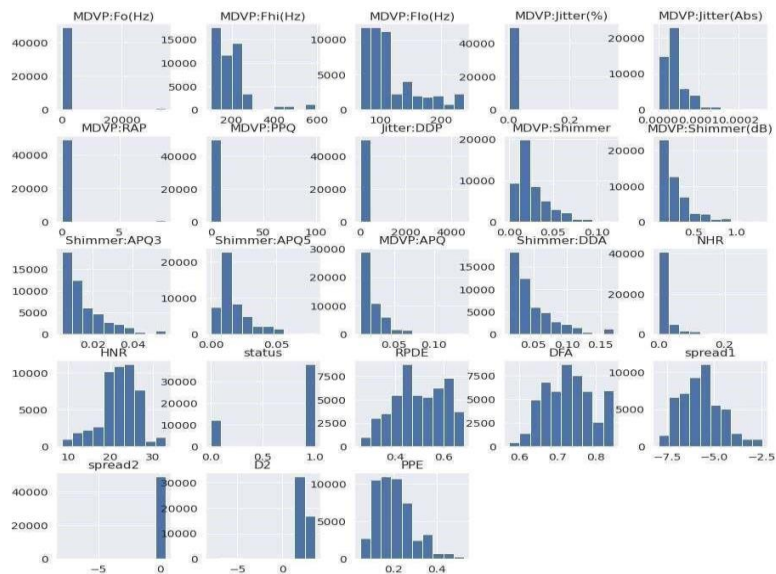
[19]: Text(0.5, 1.0, "Distribution of column : 'status'")

Distribution of column : 'status'

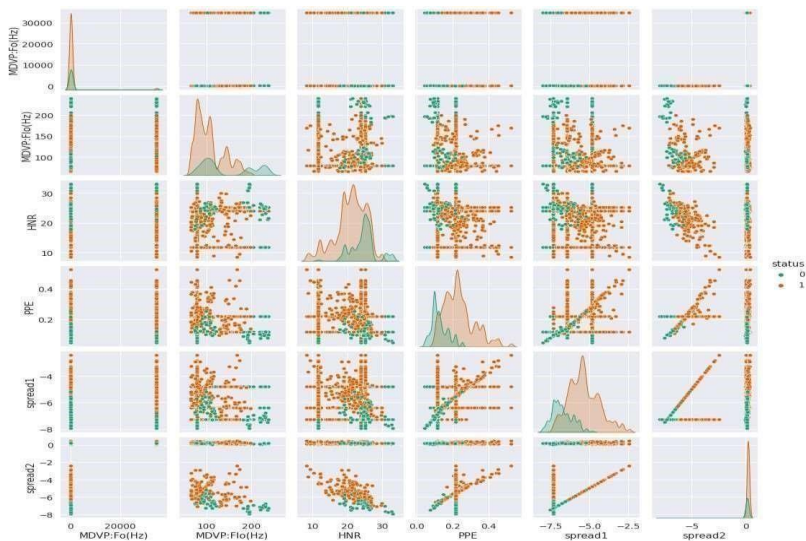


In []:

```
df.hist(figsize=(15,15));
```



```
In [ ]: plt.figure(figsize = (15,10))
sns.pairplot(df, vars=['MDVP:Fo(Hz)', 'MDVP:Fhi(Hz)', 'MDVP:Flo(Hz)', 'HNR', 'PPE', 'spread1', 'spread2'], hue='status', palette='Dark2')
plt.savefig('status')
plt.show()
```

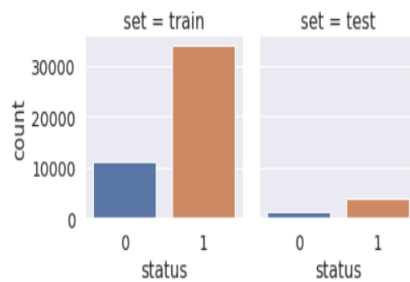


```
In [ ]: test_df = df.sample(frac=0.1)
train_df = df.drop(test_df.index).reset_index(drop=True)
test_df = test_df.reset_index(drop=True)

test_df['set'] = 'test'
train_df['set'] = 'train'

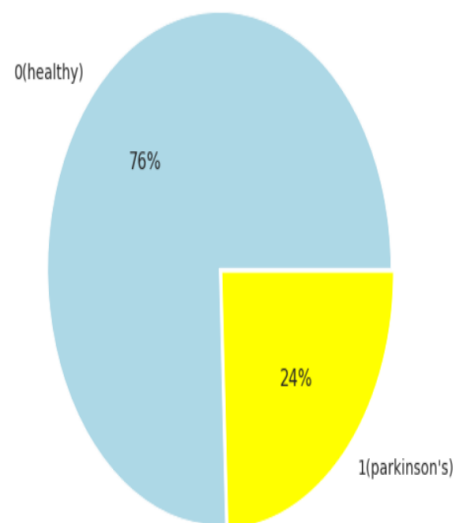
ax = sns.catplot(x="status", col="set", data=pd.concat([train_df, test_df]), kind="count", height=3)
plt.show()

print('Training dataset      : ', len(train_df))
print('Unseen data for validation : ', len(test_df))
```



Training dataset : 44928
Unseen data for validation : 4992

```
In [ ]: plt.figure(figsize=(8,8))
plt.pie(df.status.value_counts(), colors=['lightblue', 'yellow'], explode=[0, 0.02], autopct='%1.0f%%', labels=['0(healthy)'])
```



In []:

```
correlation_values=df.corr()['status']  
pd.DataFrame(correlation_values.sort_values(ascending=False))
```

it[124]:

	status
status	1.000000
spread1	0.546850
PPE	0.518188
MDVP:APQ	0.359736
MDVP:Shimmer(dB)	0.347113
Shimmer:APQ3	0.342051
Shimmer:APQ5	0.336593
Shimmer:DDA	0.319584
MDVP:Jitter(Abs)	0.319335
RPDE	0.300746
MDVP:Shimmer	0.292505
NHR	0.185635
DFA	0.181694
D2	0.126406
spread2	0.063628
MDVP:Jitter(%)	0.052747
MDVP:PPQ	0.004249
MDVP:RAP	0.004184
Jitter:DDP	0.002657

C.RESEARCH PAPER

DEEP LEARNING BASED EARLY PARKINSON'S DISEASE DETECTION FROM BRAIN MRI IMAGE

K.SOWJANYA DURGA

K.JYOTHI

Ms.DHARANI.V

Department of Computer Science

Department of Computer Science

Department of Computer Science

and Engineering

and Engineering

and Engineering

Sathyabama Institute of Science

Sathyabama Institute of Science

Sathyabama Institute of Science

and Technology

and Technology

and Technology

Chennai, Tamil Nadu

Chennai, Tamil Nadu

Chennai, Tamil Nadu

Sowjanyaakasani98@gmail.com

katajyothi80@gmail.com

vdharani.cse@sathyabama.ac.in

Parkinson's disease: a degenerative brain

ABSTRACT:

A hallmark The progressive loss of dopaminergic neurons in the substantia nigra of the brain is the hallmark of Parkinson's disease (PD), a chronic neurodegenerative condition. The intricate array of clinical presentations in PD poses challenges for categorization and diagnosis. The development of effective classification systems is imperative for early and accurate diagnosis, as well as for comprehending disease progression and treatment responsiveness. Numerous classification frameworks have emerged, encompassing factors such as clinical features, age of onset, motor symptoms, disease duration, and therapeutic outcomes. The integration of modern methods, leveraging genetic and molecular characteristics, has enhanced PD classification, aiming to delineate distinct PD subtypes and unravel underlying mechanisms for potential therapeutic interventions. These classification systems play a pivotal role in patient stratification for clinical trials, facilitating the design of more tailored treatments. Despite advancements, the quest persists for an optimal classification scheme that comprehensively considers the multidimensional features of PD, incorporating novel biomarkers and condition-of-the-artwork imaging methods. Terms: Parkinson's illness, Classification diagnosis, Motor symptoms, Clinical features, treatment response, biomarkers.

condition impacting movement control, was first identified by British physician James Parkinson in 1817.

Since then, advancements in classification and understanding have significantly enhanced the ability of medical practitioners to provide precise diagnoses and explore various treatment options.

The way Parkinson's disease is categorized predominantly relies on clinical characteristics and the progression rate of the condition. The Hoehn and Yahr scale, established in the 1960s, categorizes patients into different stages based on symptom severity, ranging from unilateral, minor disease in stage 1 to severe disability and wheelchair dependence in stage 5.

Another popular classification system that thoroughly evaluates The Movement Disorder Society Unified Parkinson's Disease Rating Scale (MDSUPDRS) measures both motor and non-motor symptoms..., as well as daily activities. This scale aids in a more nuanced evaluation of the patient's condition, providing healthcare professionals with a clearer understanding of disease progression and guiding treatment decisions.

Parkinson's disease can be further classified according to the main motor symptoms that individuals present with, such as bradykinesia, resting tremor, muscle rigidity, and postural

I.INTRODUCTION

instability Tremor-dominant and nontremordominant subtypes are identified, each presenting distinct motor characteristics.

Genetic factors also contribute in relation to Parkinson's disease categorization, with certain genetic mutations linked to its development. The SNCA gene, associated with abnormal alphasynuclein protein accumulation in the brain, is a well-known genetic marker. Other genetic changes, such as LRRK2, PARKIN, and PINK1, impact disease progression and treatment efficacy.

Advancements in neuroimaging techniques enable further categorization based on specific imaging findings. Dopamine transporter imaging using For example, SPECT or PET scans can be used to differentiate Parkinson's illness from further Parkinsonian conditions such as Multiple system atrophy with progressive supranuclear palsy.

In conclusion, Multiple clinical, genetic, and imaging criteria are integrated in the category for Parkinson's illness. This comprehensive approach enhances healthcare providers' understanding of disease progression, facilitating tailored treatment strategies for individual patients. As our knowledge of the condition expands, further refinements in classification are anticipated, contributing to advancements in diagnosis, prognosis, and treatment options.

II . RELATED WORK

[1] In the recent work by Gunduz (2021), a fresh method for dimensionality reduction for Parkinson's disease classification was introduced. The method ingeniously combines Feature selection based on filters and variational autoencoders to enhance efficiency. The study's objective was to streamline the dataset's complexity, thereby improving the diagnostic precision of Parkinson's disease. The encouraging outcomes indicate the potential applicability of this innovative approach in clinical settings.

[2] Rezaee, Savarkar, Yu, and Zhang (2022) created a distinctive technique for classifying

Parkinson's illness via means of surface electromyography (sEMG) signals. Their research aimed at improving the precision and efficacy of sEMG-based diagnosis by employing a combination of deep learning and transfer learning approaches. The study successfully demonstrated the viability of their strategy, paving the way for the development of tools for monitoring and diagnosing Using sEMG data, Parkinson's disease is determined

[3] In a study published in IEEE Access in 2019, Gunduz utilized vocal feature sets for deep learning-oriented classification of Parkinson's illness. The investigation explored the potential of utilizing vocal data to construct a reliable classification model. The findings indicated promise for the application of deep techniques for learning in accurately categorizing centered on Parkinson's illness on vocal traits.

[4] Mei, Desrosiers, and Frasnelli (2021) We out a thorough examination of the research on the application of artificial intelligence algorithms to the diagnosis of Parkinson's illness. Their analysis emphasized the significance of early and accurate disease identification, showcasing various studies utilizing machine learning To differentiate between individuals with Parkinson's illness as well as healthy counterparts. The paper offers a detailed summary of the situation as of right now of machine learning in Parkinson's disability classification.

[5] A hybrid approach to the categorization of Parkinson's illness utilizing speech was presented by Polat (2019). signals, incorporating random forests and SMOTE (Synthetic Minority Over-sampling Technique). By employing oversampling as well as machine learning algorithms, The objective of the research intended to increase the classification of Parkinson's disease's accuracy presented at the 2019 Scientific Conference on Computer Science, which was sponsored by IEEE, Biomedical Science, Electrical and Electronics the research suggested a promising avenue for early detection and treatment.

[6] An index of Parkinson's disease technique based on a sampling of one against all data and acoustic characteristics extracted from voice signals was presented by Polat and Nour (2020).. The study highlighted the possibility of accurately identifying individuals with Parkinson's disease by analyzing

specific acoustic characteristics in speech signals, emphasizing the potential for early diagnosis and intervention.

[7] Rehman et al. (2019) used a thorough machine learning methodology to pinpoint early Parkinson's disease gait characteristics that are clinically significant. The study identified stride length, stance-to-swing ratio, gait speed, and cadence as crucial gait traits for accurate early diagnosis. This underscores the promise of machine intelligence in enhancing the categorization based on Parkinson's illness on gait analysis.

[8] Balaji, A supervised machine learningbased gait classification method for Parkinson's disease early detection and stage classification was presented by Brindha and Balakrishnan (2020). The study used machine learning methods for feature extraction and classification, with an emphasis on the efficacy of gait analysis as a diagnostic instrument.. The findings highlighted the system's efficacy in accurately classifying Parkinson's disease stages, suggesting its potential for clinical applications.

[9] Maitn and others. (2020) carried out a comprehensive review on the application of machine learning methods for identifying Parkinson's illness from EEG analysis. The paper not only demonstrated the promise of these techniques for precise Classification of Parkinson's illness using EEG data but also provided a comprehensive overview of the current research landscape in this domain.

[10] Solana-Lavalle and Rosas-Romero (2020) applied Morphometry based on voxels (VBM) and artificial intelligence methodologies To classify Parkinson's illness patients based on PPMI MRI images. Their strategy presented a potential tool for early detection and intervention, contributing to the diagnostic capabilities for Parkinson's disease.

III . EXISTING SYSTEM

The current classification scheme for Parkinson's disease exhibits several limitations that compromise its utility and overall effectiveness. One significant drawback is the subjective nature of the assessment process, heavily reliant on the clinician's expertise

and observational skills. This subjectivity introduces considerable inter-rater variability, as different clinicians may interpret and evaluate the same patient's symptoms differently, leading to conflicting classification results. The lack of uniformity in this approach diminishes its reliability and repeatability, making it challenging to compare and generalize data across diverse healthcare settings.

Moreover, the reliance on clinical assessments during in-person visits in the existing system is associated with drawbacks. These assessments are often expensive, laborious, and might not accurately capture the entirety spectrum of symptoms experienced by Parkinson's patients in their daily lives. Subjective reports and qualitative observations are susceptible to error and bias, lacking the capacity for ongoing, real-time monitoring of patients' symptoms and potentially missing significant fluctuations throughout the day.

The current approach also struggles to differentiate between various phases and subtypes of Parkinson's disease, failing to account for the complexity of this neurological condition. With patients exhibiting a diverse range of symptoms, the existing classification system often overlooks these nuances, resulting in inaccurate diagnoses and ineffective management strategies. This limitation hinders healthcare professionals in tailoring treatment programs to meet individual patient needs, thereby impacting the overall quality of life and treatment outcomes.

Furthermore, the current methodology fails to leverage cutting-edge technology that could enhance the accuracy and objectivity of Parkinson's disease classification. Technologies such as wearable sensors, smartphone apps, and machine learning algorithms have the potential to continuously and objectively track motor symptoms, speech patterns, and other relevant biomarkers throughout the day. Integrating these advancements into the classification process could significantly improve diagnostic precision and facilitate personalized therapeutic approaches.

In conclusion, the existing challenges in Parkinson's disease classification, including subjectivity, lack of uniformity, limited monitoring capabilities, imprecise discrimination between subtypes, and

underutilization of evolving technology, highlight the need for a more reliable and effective classification system. Addressing these limitations is crucial to developing a classification framework capable of accurately diagnosing and monitoring the illness, which ultimately leads to improved treatment results and an improved quality of life for Parkinson's disease sufferers.

IV . PROPOSED SYSTEM

Using a vast database of human brain MRI images, The project's goal is to develop an approach based on deep learning for identifying and classifying normal and diseased brain cells by extracting attributes from a segmented region. The proposed technique extracts the form and texture characteristics Using the brain region from MRI scans to help with the multi-class classifier application of a neural network to identify the various stages of Parkinson's disease. It is believed that the suggested methodology will deliver better accuracy when compared to established methodologies.

V. METHODS

A. Data Acquisition

, and testing subsets with respective ratios of 75%, 15%, and 10%.

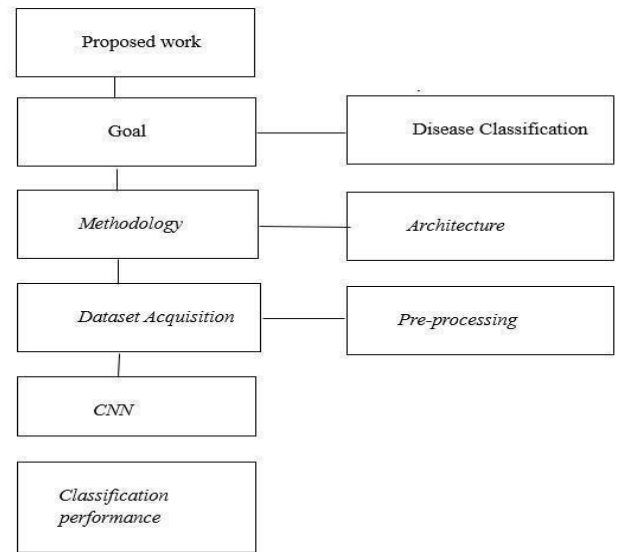
B. Pre-processing

To minimize image noise, a nonlinear approach known as median filtering is employed in the

VI. CONVOLUTION(CNN)

CNNs, or convolutional neural networks, are central to our approach, specifically designed for handling two-dimensional data. Inspired by biological processes, CNNs excel in uncovering hidden features. The architecture reduces the need for extensive hyperparameter tuning due to its spatial nature.

The initiative known as Parkinson's Progression Markers (PPMI), a ground-breaking program aimed at finding trustworthy biomarkers and facilitating an Parkinson's disease diagnosis, provided the dataset used in this investigation. include In the framework of Digital Imaging and Communications in Medicine, 458 T2 weighted MRI scans (DICOM)., the data includes 229 MRI sets for Parkinson's disease (PD) and an equal number for healthy controls (HC). The dataset is meticulously Parkinson's disease (PD) and an equal number for healthy controls (HC). The dataset is meticulously



preprocessing phase. This technique proves effective in noise reduction and edge preservation, particularly adept at addressing "salt and pepper" noise. The median filter traverses each pixel of the image, replacing each value with the middle value of its neighbors

TABLE : COMPARISION TABLE

A. Proposed Network Architecture

The suggested CNN architecture processes MR images and categorizes them into Parkinson's disease (PD) or healthy controls (HC). It incorporates five convolutional layers with varying filter sizes, ReLU activation, and max pooling. A flattening layer precedes two dense layers with ReLU and SoftMax activations. provides an overview of the network parameters.

Technique	Dataset type	Accuracy
SVM[37]	Structural MRI	92.35%
GA-ELM[3]	Resting-state fMRI	89.22%
SVM[38]	Diffusion Weighted Imaging (DWI)	86.67%
SVM[17]	Susceptibility Weighted Imaging (SWI)	86%
SVM[13]	Multimodal MRI scans (T1-weighted, T2-weighted, FLAIR)	86.96%
SVM[16]	High-Resolution MRI Scans	89%
SVM[36]	Positron Emission Tomography (PET)	90%
RVM[36]	Functional MRI (fMRI)	90%
Decision tree[15]	3-Tesla Magnetic Resonance Imaging (3T-MRI)	92%
Naive Bayes[14]	Dynamic Contrast-Enhanced MRI	93%
ANN[12]	Single Photon Emission Computed Tomography (SPECT) and TRODAT imaging	94%
CNN(proposed)	Quantitative Susceptibility Mapping (QSM) MRI Scans	96%

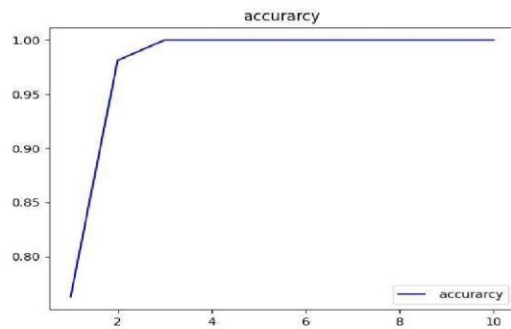


Figure 1: Accuracy

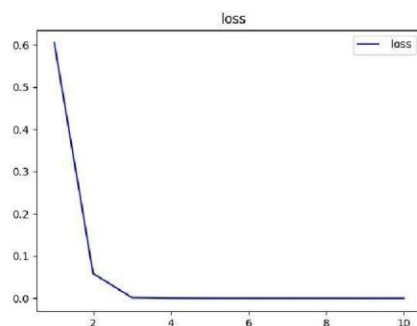
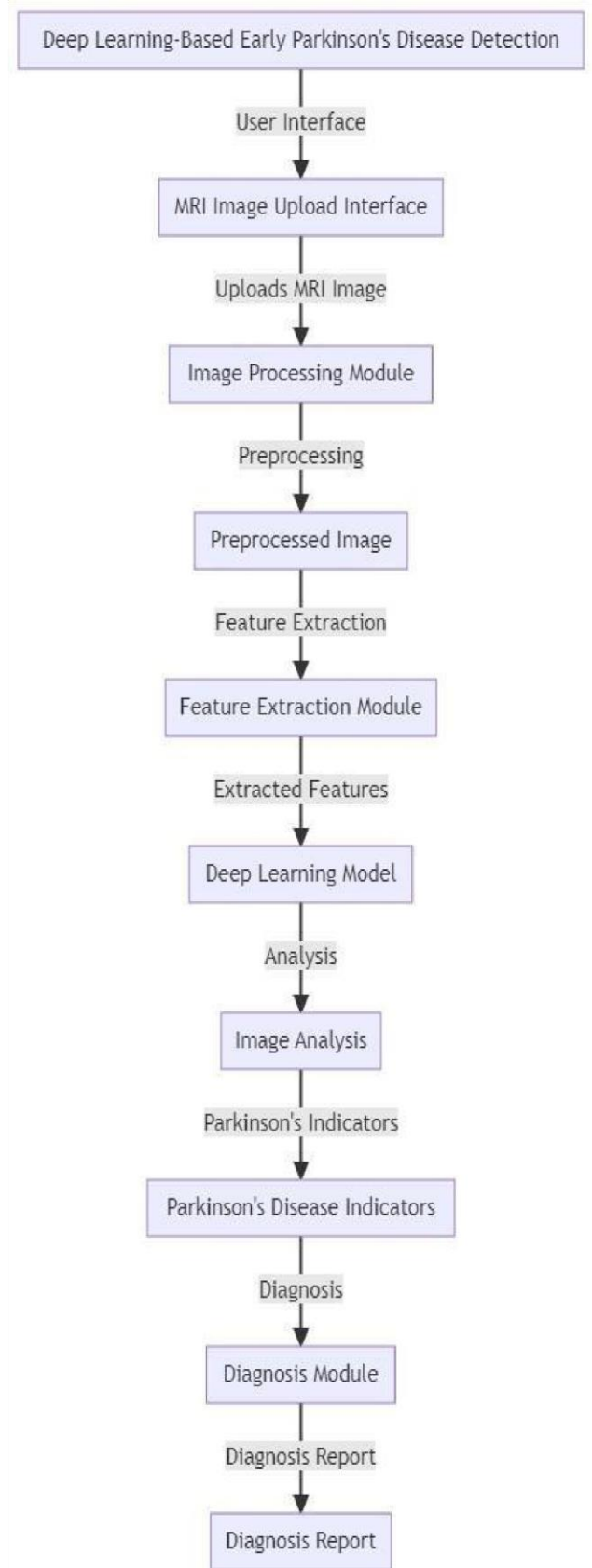


Figure 2: Loss



VII. LITERATURE SURVEY

Nilashi, M., Ghane, M., Ang, M. C., Sorooshian, S., et al.[1] Improved decision tree induction Parkinson's illness categorization through utilizing evolutionary approaches

Advanced Automated Learning Systems

The review of the literature on The classification system for Parkinson's illness reveals the increasing utilization Using cutting-edge machine-learning techniques for precise categorization and diagnosis. Numerous studies and articles highlight the effectiveness of algorithms such as support vector machines ns and features from medical data, including imaging, speech, and clinical measurements.

Multimodal Data Fusion

Another key inference from the literature is the adoption of multimodal data fusion techniques to improve classification accuracy. Researchers have explored the integration of various data sources, such as neuroimaging, voice recordings, and patient clinical data, to create a comprehensive feature set for Parkinson's disease classification. Inferences suggest that the fusion of diverse data modalities enhances the discriminative power of machine learning models, allowing for more accurate and reliable diagnoses.

Early Diagnosis and Disease Progression Monitoring

The literature survey also underscores the significance of

Parkinson's disease classification to track the disease's course and make an early diagnosis.. Researchers have emphasized the potential of machine learning-based .

classification models to detect the disease at its onset and track its progression over time. Inferences suggest that early diagnosis and continuous monitoring are crucial for timely interventions and personalized treatment strategieseventually enhancing the quality of life for those suffering with Parkinson's disease.

VIII. RESULTS AND ANALYSIS

To effectively diagnose and classify Parkinson's disease, a comprehensive classification system integrates various clinical and neuroimaging approaches. Alzheimer's disease is a complicated neurological illness that results in bradykinesia, postural instability, tremors, and stiffness as motor symptoms.. The classification system's principal goal is to differentiate Parkinson's disease from other Parkinson "disorders that share similar symptoms but originate from distinct underlying causes.

Clinical assessments involve a detailed medical history, comprehensive physical examinations, as well as the assessment of a range of symptoms, both motor and nonmotor. Based on particular motor symptoms and how well they respond to dopaminergic medications, The diagnosis of Parkinson's disease depends on diagnostic standards, such as the UK Parkinson's Disease Society Brain Bank criteria.

$$ACC = \frac{TP + TN}{TP + FP + FN + TN} \times 100$$

$$Perceptiveness = \frac{TP}{TP + FN} \times 100\%$$

$$Particulars = \frac{TN}{FP + TN} \times 100\%$$

IX . CONCLUSION

In summary, the Parkinson's disease categorization system serves as a valuable tool in accurately identifying and between individuals with Parkinson's disease and those in good health. This method offers a thorough framework for the categorization and

identification of Parkinson's disease, biomarker analyses. By considering both motor and nonmotor symptoms, healthcare practitioners can make informed

decisions regarding treatment options and management strategies, ultimately elevating the standard of patient care and improving overall quality of life.

The incorporation of technology in this methodology holds promise for Parkinson's disease early detection, which allows for prompt therapies and may change the course of the illness. Our understanding of Parkinson's disease will advance with further research and improvement of this classification system, resulting in improved diagnostic accuracy and more successful treatment strategies. The continued development of this categorization approach is expected to yield major advances in both diagnosis and treatment strategies as we learn more about the complexities of Parkinson's illness.

X. FUTURE WORK

Parkinson's disease's future classification systems offers ample opportunities for advancement in the future, with a focus on establishing a reliable and precise framework. Leveraging state-of-the-art machine learning methodologies, such as deep learning, ensemble learning, and hybrid models, holds the potential to elevate classification performance significantly. Introducing supplementary data sources, including genetic data, neuroimaging data, and comprehensive patient medical

histories, has the capacity to enhance the accuracy of the system.

The integration of wearable technology, such as smartwatches or accelerometers, stands out as a key avenue for progress. Enabling continuous monitoring of motor symptoms and real-time data collection can revolutionize the classification process, providing a more dynamic and holistic perspective.

REFERENCES

- [1] Gunduz, H. (2021). An efficient dimensionality reduction method using filterbased feature selection and variational autoencoders on Parkinson's disease classification. *Biomedical Signal Processing and Control*, 66, 102452.
- [2] Rezaee, K., Savarkar, S., Yu, X., & Zhang, J. (2022). A hybrid deep transfer learningbased approach for Parkinson's disease classification in surface electromyography signals. *Biomedical Signal Processing and Control*, 71, 103161.
- [2] Gunduz, H. (2019). Deep learning-based Parkinson's disease classification using vocal feature sets. *IEEE Access*, 7, 115540-115551.
- [3] Mei, J., Desrosiers, C., & Frasnelli, J. (2021). Machine learning for the diagnosis of Parkinson's disease: a review of literature. *Frontiers in aging neuroscience*, 13, 633752.
- [4] Polat, K. (2019, April). A hybrid approach to Parkinson disease classification using speech signal: the combination of smote and random forests. In *2019 Scientific Meeting on Electrical Electronics & Biomedical Engineering and Computer Science (EBBT)* (pp. 1-3). Ieee.
- [5] Polat, K., & Nour, M. (2020). Parkinson disease classification using one against all based data sampling with the acoustic features from the speech signals. *Medical Hypotheses*, 140, 109678.
- [6] Rehman, R. Z. U., Del Din, S., Guan, Y., Yarnall, A. J., Shi, J. Q., & Rochester, L. (2019). Selecting clinically relevant gait characteristics for classification of early Parkinson's disease: a comprehensive machine learning approach. *Scientific reports*, 9(1), 17269.
- [7] Balaji, E., Brindha, D., & Balakrishnan, R. (2020). Supervised machine learning based gait classification system for early

D. CERTIFICATE



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)
CATEGORY-1 UNIVERSITY BY UGC
Accredited 'A++' Grade by NAAC [12B Status by UGC] Approved by AICTE



www.sathyabama.ac.in

SCHOOL OF COMPUTING

Department of Computer Science and Engineering

Certificate of Presentation

This is to certify that

K. JYOTHI

of

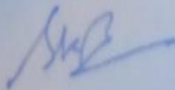
Sathyabama Institute of Science and Technology

has presented a paper entitled

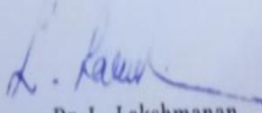
Early Parkinson's Disease Detection from Brain MRI Images

in the

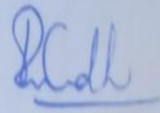
International Conference on Cognitive Computing and Artificial Intelligence (ICCCAI-2024) in association with
Taylor's University, Malaysia held on 7th & 8th March 2024.



Dr. Sayan Kumar Ray
Head, School of Computer Science
Taylor's University



Dr. L. Lakshmanan
HOD, CSE Dept
Sathyabama Institute of Science and Technology



Dr. T. Sasikala
Dean, SOC
Sathyabama Institute of Science and Technology