SCHOOL OF ELECTRONICS ENGINEERING

Department of Electronics and Communication Engineering

Detecting Parkinson's Disease using XGBoost and Random Forest Classifier

Thanush N
(20BEC0559)

Kanishkar RC
(20BEC0625)

Kanishka Rithu D
(20BEC0730)

Project Report
of
ECE 3502 – IoT Domain Analyst

Winter Sem 2022-23

Submitted to
Faculty: Dr. R. Sujatha
Date: 14/04/2023
Slot: B2

Index

Abstract:

Parkinson's disease (PD) affects body movement, causes involuntary quivering movement and inability to move easily and without pain, and is a furtherance ailment of the central nervous system. It comprises five stages and affects more than 1 million people in India each year. There is currently no treatment available for this chronic condition. It is a neurodegenerative condition that affects brain cells that make dopamine. This disease can be identified by using machine learning algorithms called “XGboost” and “Random Forest Classifier.” Extreme Gradient Boosting, or XGBoost, is a decision tree-based algorithm. Parkinson's disease is a neurodegenerative disorder affecting dopamine-producing neurons in the brain, and XGBoost is a new Machine Learning algorithm anticipated for speed and acceptable performance. This Python project will create a model using an XGBclassifier and determine its correctness. Tremors, trouble moving, and other motor symptoms are caused by the loss of dopamine-producing cells in the brain that characterise this condition.

The substantia nigra, a region of the brain, loses nerve cells because of Parkinson's disease. As a result, the amount of dopamine in the brain decreases. Dopamine is essential for controlling how the body moves. The cause of many Parkinson's disease symptoms is a decrease in dopamine. It is unclear what specifically results in the loss of nerve cells. Most specialists believe that a confluence of hereditary and environmental variables is to blame. Early PD identification is crucial since it can help to reduce the disease's progression and enhance the quality of life for the patient. Parkinson's disease symptoms first appear modest and typically worsen over time. Parkinson's disease is accompanied by a wide range of symptoms. However, the timing of events as well as their severity vary from person to person. A person with Parkinson's disease is unlikely to encounter all or the majority of these.

In recent years, high accuracy PD diagnosis has been achieved by analysing patient data using machine learning algorithms like XGBoost. This study uses data from the Parkinson's Disease Classification Challenge dataset to investigate how well XGBoost performs in detecting PD. The findings demonstrate that XGBoost detected PD from the dataset with an accuracy of over 90%, underlining the promise of machine learning algorithms for precise diagnosis of PD.

Most people with Parkinson's start to develop symptoms when they're over 50, although some people with the condition first experience symptoms when they're under 40. Men are slightly more likely to get Parkinson's disease than women. Parkinson's disease symptoms might worsen as the condition worsens, making it harder and harder to perform daily tasks on one's own. While most patients benefit from treatment and only experience mild to moderate disability, a minority may not and may eventually experience more severe disability. Although Parkinson's disease does not directly cause death, it can put a great deal of stress on the body and increase a person's susceptibility to serious and life-threatening diseases. However, because to improvements in medicine, the majority of those who have Parkinson's disease today have normal or very normal life expectancies.

Introduction:

Parkinson’s disease is a disorder of the nervous system which affects the movement and induces tremors and stiffness. The Parkinson disease has 5 stages to it and affects more than 1 million individuals every year in India. This is a progressive disorder and has no cure yet. It is a neurodegenerative disorder affecting dopamine-producing neurons in the brain. A neurological condition affecting 60% of people over 50. Parkinson's disease (PWP) patients struggle with speech impairment and movement issues, which makes it difficult for them to travel for appointments for treatment and monitoring. Early discovery of PD enables treatment, allowing patients to live normal lives. The necessity to identify PD early, remotely, and correctly is highlighted by the world's aging population. Meanwhile, in the early non-muscular symptoms of Parkinson's disease (PD) might be not severe and brought on by a variety of other illnesses. Thus, it might be difficult to diagnose PD at an early stage because these symptoms are frequently disregarded. Machine learning algorithms have been used for the identification of PD and patients who are healthy or doesn’t have PD with reasonable clinical presentations to solve these issues and improve the diagnosis and assessment procedures of PD. The application of machine learning techniques in telemedicine to identify PD in its early stages is highlighted in this research. To detect this a machine algorithm called XGboost is used. XGBoost is a Machine Learning algorithm which is formulated with the features of speed and performance. XGBoost abbreviates as extreme Gradient Boosting, and it is based on decision trees. In this project, we will import the “XGBclassifier” from the “xgboost” library; this is an implementation of the “scikit-learn” API for XGBoost classification. In this Python machine learning project, we'll create a model using an XGBclassifier using the Python packages scikit-learn, “NumPy” and “pandas”. After loading the data, feature extraction will be done from the data. Also another method is used to find the presence of Parkinson with images

Literature Review:

1) Amin ul Haq et.al. Parkinson's Disease (PD) is one of the most popular conditions following Alzheimer's disease. Medical practitioners and researchers have found it difficult to identify the presence of Parkinson's disease correctly and promptly. Traditional approaches based on reduced characteristics and Statistical Learning have been developed for PD diagnosis. The literature displays that when DL-based diagnosis algorithms are used and trained on larger datasets, their performance accuracy improves.

2) Rohit Lamba et.al. The authors propose a speech signal-based hybrid PD detection system. A hybrid system for Parkinson’s disease diagnosis has shown the best performance with a very high accuracy. This result is also better than the standard methods used.

3) Mahmoud Ragab et.al. Parkinson’s disease (PD) affects the movement of people, including significant differences in writing skill, drawing, speech, and stiffness in muscles. Severe or intense stages of PD are risky as the patients get the inability of standing or walking.

4) Zhang, J. Parkinson's disease (PD) is a popular chronic condition that has no known cure as of now. It is a neurodegenerative movement disorder. Machine-learning techniques have been increasingly applied such as multi-modal techniques to the diagnosis of PD which enhances the performance of diagnosis in terms of time and accuracy

5) Emuel Daniel Pah et.al. Parkinson's disease (PD) is a central nervous system-based disorder in which a particular region called the “substantia nigra” region of the brain is affected. PD is diagnosed with reference on clinical observation of the symptoms and self-reported functional impairments. Using voice as an input has the advantage of recording it easily as no special apparatus of machines are required. People with Parkinson's disease (PD) often have dysarthria or slurring of the voice. There are significant differences between the voice parameters of people with PD and healthy participants. Age, gender, and ethnicity are confounding factors. Computational approaches aid in PD diagnosis and rehabilitation are being actively investigated.

6) Gunjan Pahuja et.al. Two frameworks, feature-level and modal-level are presented to classify the given subjects into PD and healthy. Neuroimaging (T1 weighted MRI scans and SPECT) and biological (CSF) features are used to form a heterogeneous dataset for deep learning models to diagnose PD.

7) Majid Aljalal et.al. Parkinson's disease is a central nervous system-based condition that affects people all over the world. Electroencephalography (EEG) is a method that is used to analyze brain activity which therefore can be used to diagnose PD.

8) Ajay Sankar Gullapalli et.al. The speech of a person can be an efficient tool for the early detection of diseases like Parkinson's, Alzheimer's, Autism and several other similar conditons in elderly people. Review of speech features can be analyzed using machine learning algorithms which is presented to help in development of non-invasive signal processing techniques.

9) Tarjni Vyas .et.al.Parkinson's disease (PD) is a central nervous system based disorder. MRI scans are one of the tools that can identify biomarkers that can be used to know how the disease spreads, leading to a cure in the future. These MRI scans can then be used an input for CNN models to predict the presence of PD

10) Changqin Quan et.al. One of the popular symptom with PD patients is that they suffer from hypokinetic dysarthria which is a speech-based condition. This paper gives a method with a deep learning model for diagnosis of patients with PD from speech signals.

11) John M.Tracy et.al. This research demonstrates the potential of speech signals of an individual as a deep phenotype for Parkinson's Disease (PD), the second most common neurodegenerative disorder worldwide. It analyzed the audio recordings of healthy patients and patients with PD and then this was analyzed using machine learning models to display the severity of PD.

12) H. Gunduz et.al. This study suggests two methods to classify a patient with the Parkinson condition where the first uses CNN with a 8 layered model to differentiate the other one uses deep learning and linear convolution to diagnose a patient

13) Turker Tuncer et.al. This study proposes a new method to autonomously detect the presence of Parkinson's disease (PD) using a combination of minimum average maximum average algorithm (MAMa) tree and singular value decomposition (SVD). The highest classification accuracy rate was achieved with KNN classifiers. The proposed method has proven to work with a huge database, and this can help neurologists in the diagnosis of PD.

14) Laureano Moro-Velazquez et.al. This review paper identifies the most common speech features like the phonatory and stuttering that can be analyzed through various machine learning techniques to assess the severity of PD. It has proven that these aspects of speech analysis are relevant for the automatic detection and severity assessment of PD.

15) T.J.Wroge et.al. This study examined how well supervised classification algorithms perform when used to accurately identify Parkinson's disease (PD). The models have proven to outperform many medical and clinical specialists with a higher percentage for diagnosis. This suggests that trustworthy techniques that can convert audio data into a diagnostic tool for medical experts may be able to deliver diagnoses that are more affordable and precise.

16) Imanne El.Maachi et.al. This research paper introduces a new intelligent technique based on deep learning by analyzing gait information. It uses a 1-dimensional convolutional neural network to build a Deep Neural Network (DNN) classifier that processes and aids in the diagnosis of the presence of parkinson's condition

17) Jie Mei et.al. Parkinson's disease (PD) is typically diagnosed primarily on clinical indicators and medical observations, however conventional diagnostic methods may be vulnerable to subjectivity. ML techniques are widely used to segregate people with Parkinson’s and healthy patients. A study of 210 papers in the literature revealed a great potential for machine learning techniques and novel biomarkers used in clinical decision helping, resulting in an ordered thorough diagnosis of Parkinson's disease.

18) Gabriel Solana-Lavalle et.al Using a very large and new dataset this research reduces the vocal features and also increases accuracy of Parkinson’s Disease detection. The Four classifiers used to give an accuracy of 95.4%, sensitivity of 97.2%, specificity of 91.88%, and precision of 98.11%. The computational complexity is greatly reduced by selecting no more than 19 features.

19) Zehra Karapinar Senturk et.al. This disease is brought on by damage to the brain cells that make dopamine, which is essential for movement control, flexibility, and adaptation. In this research, recursive feature elimination and feature importance approaches are proposed for a machine learning based diagnosis of Parkinson's disease. 92.99 percent accuracy was obtained with a very small number of features.

20) Prof Eduardo Tolosa PhD et.al.This Parkinson's disease is a major neurological disorder and is said to quadruple over the next 30 years. The validation of clinical research and testing of research criteria for this disease, and the discovery of genetic subtypes and variants linked to risk of Parkinson's disease are considered as recent breakthroughs. Genetic and imaging studies are currently a standard part of clinical practice protocols, and biomarkers discovery is a significant progress. The disease has changed to a biomarker-supported entity, and new disease-modifying therapies are now being developed.

Problem statement:

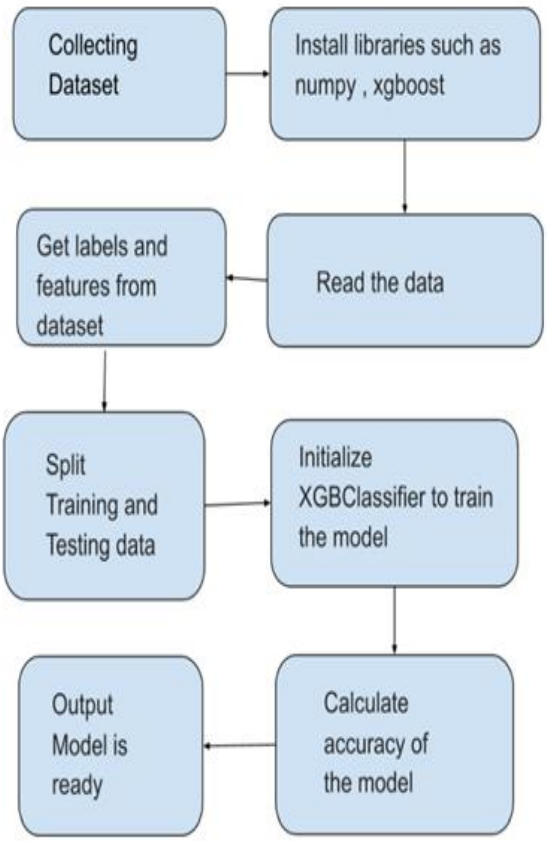
Parkinson's disease includes recognition of a stable and accurate method for diagnosing the disease in patients. Parkinson's disease is a neurological disorder that affects body movement, causes involuntary quivering movement and inability to move easily and without pain and difficulty with coordination and balance. However, these symptoms are the indication of the disease and there are many stages in it and it is very important to look after it before it attains the last stage and it is very difficult to diagnose this disease because there is more chance for it to be a different disease because these all are the common symptoms for most of the disease which is making it challenging to diagnose Parkinson's disease accurately. The problem statement, therefore, involves recognising a set of diagnostic criteria or tests that can differentiate Parkinson's disease from other similar conditions.

Proposed Methodology:

XG Boost:

The XGBoost (extreme Gradient Boosting) is a popular and efficient open-source implementation of the gradient boosted trees algorithm. Gradient boosting is a supervised learning algorithm that attempts to accurately predict a target variable by combining an ensemble of estimates from a set of simpler and weaker models. The XGBoost algorithm performs well in machine learning competitions because of its robust handling of a variety of data types, relationships, distributions, and the variety of hyperparameters that you can fine-tune. You can use XGBoost for regression, classification (binary and multiclass), and ranking problems. You can use the new release of the XGBoost algorithm either as an Amazon SageMaker built-in algorithm or as a framework to run training scripts in your local environments. It provides an XGBoost estimator that executes a training script in a managed XGBoost environment. The current release of SageMaker XGBoost is based on the original XGBoost versions 1.0, 1.2, 1.3, and 1.5.

System Flow Diagram:



Project Description:

XG Boost:

A well-liked machine learning technique called XGBoost (Extreme Gradient Boosting) is utilized for supervised learning tasks like classification and regression. A final prediction is made by combining the predictions of various base models using this type of ensemble learning technique. It employs a gradient descent approach to minimize a loss function and gradient boosting to enhance the performance of each decision tree individually. Due to its versatility in terms of the kinds of problems it can be used to, its high level of accuracy, capacity to handle enormous datasets, and popularity among data scientists. In this machine learning based decision trees algorithm, we combine many weak trees or branches to come up with one strong tree or branches. The weak trees here are the individual decision trees. All the trees are connected in series and each tree tries to minimize the error of the previous tree. Due to this type of connection, boosting algorithms usually tend to be slow to learn, but also highly accurate in providing a decision. In statistical learning models it has been proved that the models which learn slowly perform better.

- Initializing the learning model with a constant value

$$F_0(x) = \arg\min_{\gamma} \sum_{i=1}^{n} L(y_i, \gamma)$$

- For m = 1 to M, pseudo-residuals are computed

$$r_{im} = -\left[\frac{\partial L(y_i, F(x_i))}{\partial F(x_i)}\right]_{F(x)=F_{m-1}(x)} \quad \text{for } i = 1, \ldots, n.$$

- Fit a base learner (or weak learner, e.g. tree) closed under scaling $h_m(x)$ to pseudo-residuals, i.e. train it using the training set $\{(x_i, r_{im})\}_{i=1}^{n}$

- Compute multiplier γ_m by solving the following one-dimensional optimization problem

$$\gamma_m = \arg\min_{\gamma} \sum_{i=1}^{n} L\left(y_i, F_{m-1}(x_i) + \gamma h_m(x_i)\right)$$

- Update the model

$$F_m(x) = F_{m-1}(x) + \gamma_m h_m(x)$$

- Output F_m(x)

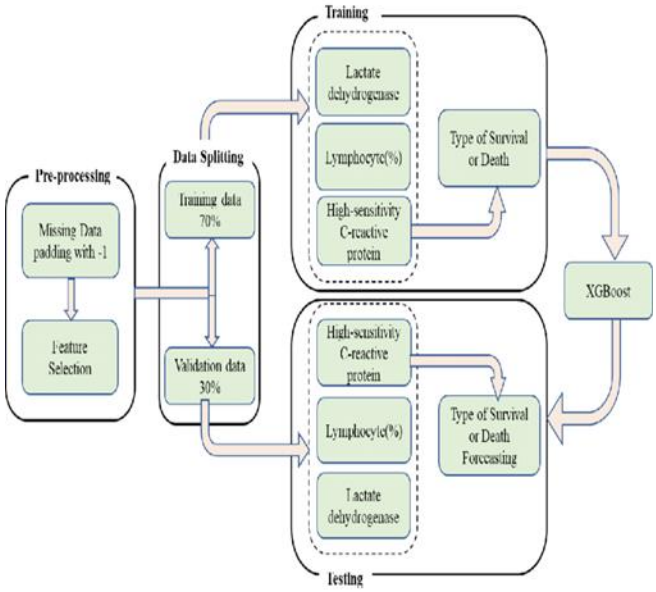


Fig.1 XGboost algorithm

NumPy:

The Python library NumPy offers a potent collection of tools for working with numerical data, including exponential, logarithmic, trigonometric, and arithmetic functions. Several Python libraries for machine learning, data analysis, and scientific computing are built on top of it as well. The Python package NumPy is used to handle and control arrays in a skilful manner. Moreover, it

has matrices, Fourier transform, and functions for working in linear algebra. The equivalent of arrays in Python are lists, although they take more time to implement. The intention of NumPy is to provide array objects that are even better and 50 times faster than conventional Python lists. The NumPy array object is specified as ndarray, and it has several supporting methods that make using ndarray relatively simple. In data research, where speed and resources are vital, the arrays are the ones which are mostly occupied.

Pandas:

The open-source Python data manipulation and analysis toolkit Pandas offers strong and adaptable data structures including Series and Data Frame. The ability to read and write data from a range of file types is also built in. It can work with other Python packages, making it a potent tool for modelling and data analysis. Pandas is a Python module used to expedite data cleaning, pre-processing, and analysis. NumPy, Python's mathematical library, is the foundation upon which Pandas is constructed. The pandas library enables you to work with tabular data with columns of various data types, including those from Excel spreadsheets, CSV files downloaded from the internet, SQL database tables, Time series data, either at fixed-frequency or not, and other structured datasets, including those derived from web data, like JSON files.

Sklearn:

A well-established accessible source machine learning toolkit for Python called Scikit-Learn which provides a diverse set of tools for setting up several machine learning algorithms or techniques, like classification models, regression models, clustering models, and dimensionality reduction, into practice. It offers a variety of integrated machine learning models, including support vector machines (SVM), logistic regression, decision trees, random forests, and k-nearest neighbours (KNN). Also, it functions well with other libraries like Pandas and Seaborn.

UCI Machine Learning Repository:

A well-known online repository called the UCI Machine Learning Repository offers a variety of datasets and other resources for testing and comparing machine learning methods. It is free to use and is managed by the University of California, Irvine. It provides the machine learning community with many datasets that are currently maintained. The machine learning community uses it as a collection of databases, domain theories, and data generators for the empirical investigation of machine learning algorithms.

Random Forest Classifier:

The random forest classifier can be used to resolve classification or regression issues. Each decision tree in the ensemble that makes up the random forest method is composed of a bootstrap sample of data that is taken from a training set with replacement. One of the most highly preferred and frequently used algorithms among data scientists is random forest. Supervised machine learning algorithms like random forest are frequently employed when the requirements of classification and regression problems are issued. Using various types of samples, this algorithm constructs and builds decision trees and uses their average for classification and majority vote for

regression. One of the most predominant features of the Random Forest Algorithm is its ability to handle data sets with continuous variables, as in regression, and categorical variables, as in classification. For classification and regression tasks, it performs better.

Steps involved in Random Forest Algorithm:

- In the Random Forest model, a small set of data points and a small set of features are selected for constructing and building each decision tree. Simply put, "x" random records and "y" features are taken from the data set having "z" number of records.

- Individual decision trees are constructed for each sample to determine a particular decision in a case

- Each decision tree will generate an output acting as the input for the following successive tree

- Final output is considered based on Majority Voting or Averaging for Classification and regression, respectively.

Mathematics behind Random Forest:

- **Regression Problem:**

  To predict how the data branches out from each node when utilizing the Random Forest Algorithm to solve regression problems, there must be prior knowledge about the mean squared error (MSE). To predict which branch is the best choice for your forest, the formula below is used which estimates the distance between each node and the expected actual value. In this case, "fi" is taken as the value which the decision tree returned, and "yi" is the value of the data point that is tested at a particular node.

$$MSE = \frac{1}{N} \sum_{i=1}^{N} (fi - yi)^2$$

- **Classification Problem:**

  We should be conscious that the Gini index, or the algorithm which is used for identifying how nodes on a decision tree branch are controlled and organized and how frequently it is occupied while implementing the random forests based upon the designated data. The Gini of each branch on a node is determined using this formula that is using the class and probability and mentioning which branch is more likely to be established. In this case, the c term stands for the number of classes, and pi is the relative frequency of the class which we are observing in the dataset.

$$Gini = 1 - \sum_{i=1}^{C} (p_i)^2$$

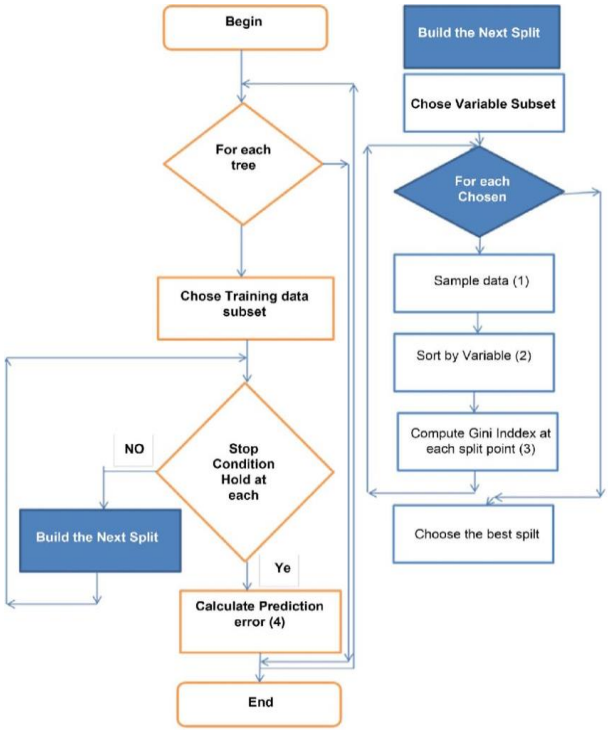


Fig.2 Random classifier algorithm

Results and Discussion:

In this project, we have analysed and learnt how to use a variety of indicators to identify individuals who seem to suffer from Parkinson's disease. To do this, we used an XGBclassifier algorithm and the sklearn library to prepare the dataset. This machine learning results in an accuracy of **94.87%** in predicting the presence of PD in a particular individual which is excellent given that the size of the code is very small.

	name	MDVP:Fo(Hz)	MDVP:Fhi(Hz)	MDVP:Flo(Hz)	MDVP:Jitter(%)	MDVP:Jitter(Abs)	M
0	phon_R01_S01_1	119.992	157.302	74.997	0.00784	0.00007	
1	phon_R01_S01_2	122.400	148.650	113.819	0.00968	0.00008	
2	phon_R01_S01_3	116.682	131.111	111.555	0.01050	0.00009	
3	phon_R01_S01_4	116.676	137.871	111.366	0.00997	0.00009	
4	phon_R01_S01_5	116.014	141.781	110.655	0.01284	0.00011	

Fig.3 Dataset

Fig.3:

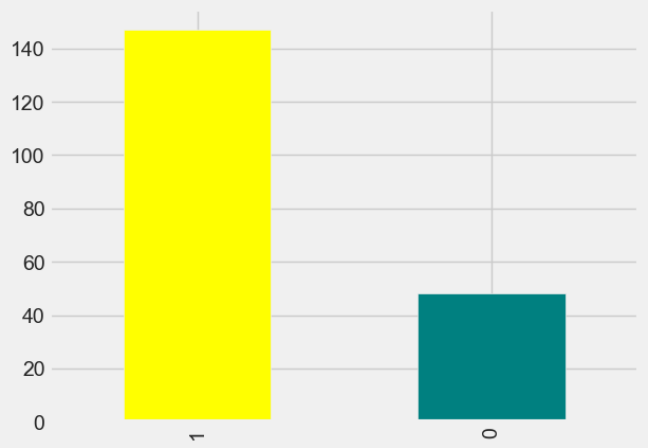


Fig.4 Bar chart (status)

Fig.4: ‘1’ represents the people with Parkinson’s and ‘0’ represents Healthy status.

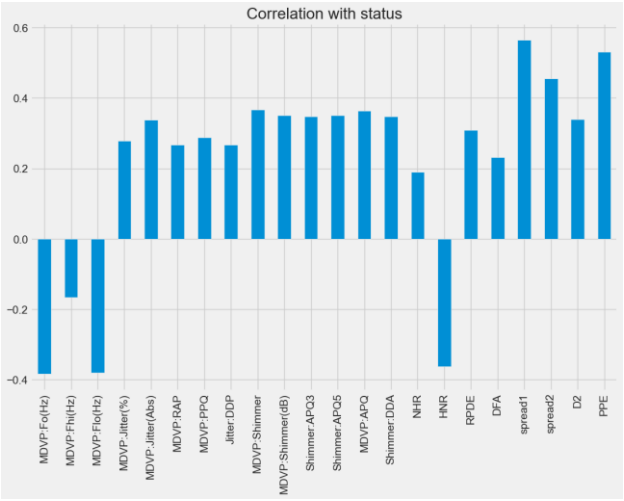


Fig.5 Correlation with Status

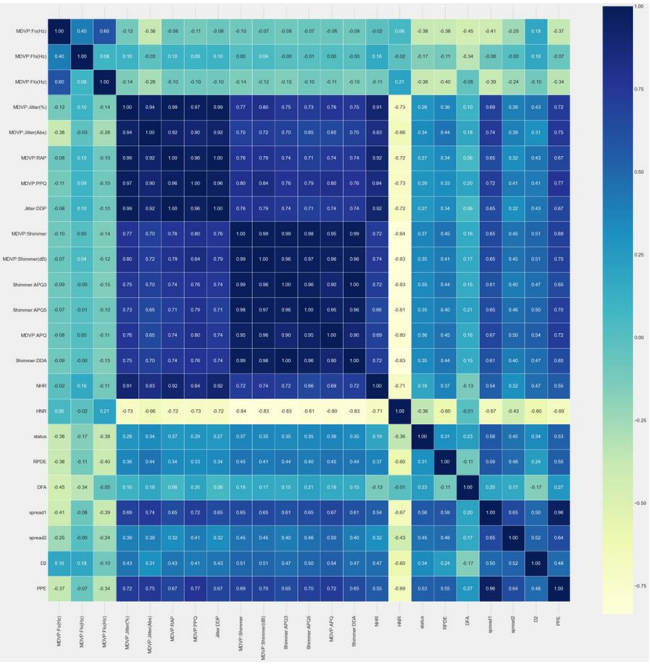


Fig.6 Correlation Matrix

SPIRAL		
	XGBOOST	RANDOM FOREST CLASSIFIER
ACCURACY	73..33 %	86.67 %
SENSITIVITY	73.33 %	80.00 %
SPECIFICITY	73.33 %	93.33 %

Table I

WAVE		
	XGBOOST	RANDOM FOREST CLASSIFIER
ACCURACY	76.67 %	70.00 %
SENSITIVITY	73.33 %	73.33 %
SPECIFICITY	80.00 %	66.67 %

Table II

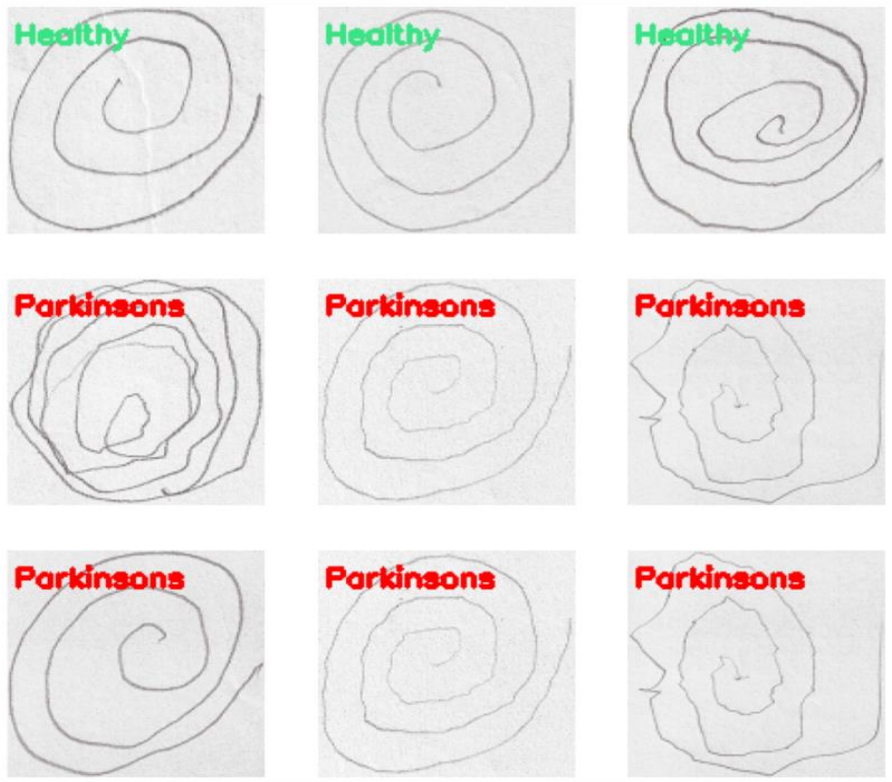


Fig.7 Spiral Output

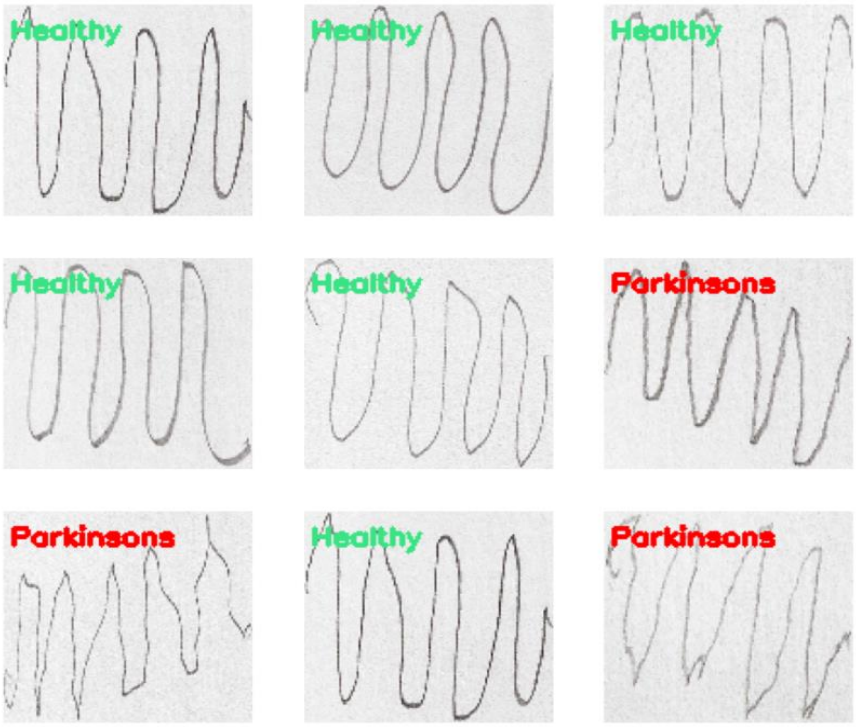


Fig.8 Wave Output

Conclusion:

Parkinson's disease, which affects the central nervous system (CNS) of the brain, is fatal if not detected early. There are various stages of Parkinson's. If it's found in early stages, it is possible to cure or extend life. Due to late finding, treatment is not provided, and lives are lost. It is crucial to make an early diagnosis as a result. For early disease diagnosis, we employed machine learning techniques like XGBoost and Random Forest classifiers. According to our analysis of our data on Parkinson's disease, the algorithm known as XGBoost is the most accurate in identifying the condition's onset. XGBoost finds the status of whether a person has Parkinson's or not by evaluating the dataset of numeric values such as Mdvp are given, and it gives the accuracy of how correctly it predicts the outcome. While the Random Forest classifier uses image dataset, where the patient is required to draw a spiral and wave drawing. The Random Forest classifier then differentiates the people who have Parkinson's and those who don't have Parkinson's. The output is labelled as either Parkinson's or Healthy. With the help of Machine learning techniques we can train the model to be more accurate which in turn gives better results. Early treatment will be possible as a result, perhaps saving lives.

Future Scope:

In this project we have developed two different ways to recognize Parkinson's Disease using XGB & Random Forest Classifier from the data set of both numeric data and pictorial image-type data. We can also compare the given data with classifiers like SVM, Light GBM and CAT Boost to predict accuracy, sensitivity, specificity of the outcomes. This can also be taken to another level by including camera module to detect patients using open CV and ML techniques to train the model to detect patients whether they have Parkinson's or not.

References:

1. Lamba, Rohit, et al. "A hybrid system for Parkinson’s disease diagnosis using machine learning techniques." International Journal of Speech Technology 25.3 (2022): 583-593.
2. ul Haq, Amin, et al. "A survey of deep learning techniques based Parkinson’s disease recognition methods employing clinical data." Expert Systems with Applications 208 (2022): 118045.
3. Bahaddad, Adel A., et al. "Metaheuristics with Deep Learning-Enabled Parkinson’s Disease Diagnosis and Classification Model." Journal of Healthcare Engineering 2022 (2022).
4. Zhang, Jing. "Mining imaging and clinical data with machine learning approaches for the diagnosis and early detection of Parkinson’s disease." npj Parkinson's Disease 8.1 (2022): 1-15.
5. Hireš, Máté, et al. "Convolutional neural network ensemble for Parkinson's disease detection from voice recordings." Computers in biology and medicine 141 (2022): 105021.
6. Gunjan Pahuja, Bhanu Prasad,Deep learning architectures for Parkinson's disease detection by using multi-modal features,Computers in Biology and Medicine, Volume146, 2022,105610,ISSN 0010-4825.
7. Aljalal, M.; Aldosari, S.A.; AlSharabi, K.; Abdurraqeeb, A.M.; Alturki, F.A. Parkinson’s Disease Detection from Resting-State EEG Signals Using Common Spatial Pattern, Entropy, and Machine Learning Techniques. Diagnostics **2022**, 12, 1033.
8. Gullapalli, A.S., Mittal, V.K. (2022). Early Detection of Parkinson’s Disease Through Speech Features and Machine Learning: A Review. In: Senjyu, T., Mahalle, P.N., Perumal, T., Joshi, A. (eds) ICT with Intelligent Applications. Smart Innovation, Systems and Technologies, vol 248. Springer, Singapore.
9. Kirti Raj Bhatele, Anand Jha, Kavish Kapoor, Devanshu Tiwari, Neurodegenerative diseases-Caps: a capsule network based early screening system for the classification of neurodegenerative diseases, Cognitive Neurodynamics, 10.1007/s11571-022-09787-1, (2022).

10. Changqin Quan, Kang Ren, Zhiwei Luo, Zhonglue Chen, Yun Ling,End-to-end deep learning approach for Parkinson's disease detection from speech signals,Biocybernetics and Biomedical Engineering,Volume 42, Issue 2,2022,Pages 556-574,ISSN 0208-5216.
11. H. Gunduz, "Deep Learning-Based Parkinson's Disease Classification Using Vocal Feature Sets," in IEEE Access, vol. 7, pp. 115540-115551, 2019, doi: 10.1109/ACCESS.2019.2936564.
12. John M. Tracy, Yasin Özkanca, David C. Atkins, Reza Hosseini Ghomi, Investigating voice as a biomarker: Deep phenotyping methods for early detection of Parkinson's disease.
13. T. J. Wroge, Y. Özkanca, C. Demiroglu, D. Si, D. C. Atkins and R. H. Ghomi, "Parkinson's Disease Diagnosis Using Machine Learning and Voice," 2018 IEEE Signal Processing in Medicine and Biology Symposium (SPMB), Philadelphia, PA, USA, 2018, pp. 1-7, doi: 10.1109/SPMB.2018.8615607.
14. Laureano Moro-Velazquez, Jorge A. Gomez-Garcia, Julian D. Arias-Londoño, Najim Dehak, Juan I. Godino-Llorente, Advances in Parkinson's Disease detection and assessment using voice and speech: A review of the articulatory and phonatory aspects, Biomedical Signal Processing and Control, Volume 66, 2021
15. Turker Tuncer, Sengul Dogan, Udyavara Rajendra Acharya, Automated detection of Parkinson's disease using minimum average maximum tree and singular value decomposition method with vowels, 2020
16. Mei Jie, Desrosiers Christian, Frasnelli JohannesTITLE=Machine Learning for the Diagnosis of Parkinson's Disease: A Review of LiteratureJOURNAL=FrontiersinAgingNeuroscienceVOLUME=13YEAR=2021DOI=10.3389/fnagi.2021.633752 ISSN=1663-4365
17. Imanne El Maachi, Guillaume-Alexandre Bilodeau, Wassim Bouachir, Deep 1D-Convnet for accurate Parkinson disease detection and severity prediction from gait, Expert Systems with Applications, Volume 143, 2020
18. Gabriel Solana-Lavalle, Juan-Carlos Galán-Hernández, Roberto Rosas-Romero, Automatic Parkinson disease detection at early stages as a pre-diagnosis tool by using classifiers and a small set of vocal features, Biocybernetics and Biomedical Engineering, Volume 40, Issue 1, 2020
19. Zehra Karapinar Senturk,Early diagnosis of Parkinson's disease using machine learning algorithms,Medical Hypotheses,Volume 138,2020,109603,ISSN 0306-9877.
20. Eduardo Tolosa, Alicia Garrido, Sonja W Scholz, Werner Poewe,Challenges in the diagnosis of Parkinson's disease,The Lancet Neurology,Volume 20, Issue 5,2021,Pages 385-397,ISSN 1474-4422.
21. Brooks, D. J. (1991). Detection of preclinical Parkinson's disease with PET. Neurology, 41(5 Suppl 2), 24-27.

22. Sama, A., Pérez-López, C., Romagosa, J., Rodriguez-Martin, D., Catala, A., Cabestany, J., ... & Rodríguez-Molinero, A. (2012, August). Dyskinesia and motor state detection in Parkinson's disease patients with a single movement sensor. In 2012 Annual International Conference of the IEEE Engineering in Medicine and Biology Society (pp. 1194-1197). IEEE.

23. Nilashi, M., Ibrahim, O., & Ahani, A. (2016). Accuracy improvement for predicting Parkinson's disease progression. Scientific reports, 6(1), 34181.

24. Williams, J. R., Hirsch, E. S., Anderson, K., Bush, A. L., Goldstein, S. R., Grill, S., ... & Marsh, L. (2012). A comparison of nine scales to detect depression in Parkinson disease: which scale to use?. Neurology, 78(13), 998-1006.

25. Narendra, N. P., Schuller, B., & Alku, P. (2021). The detection of parkinson's disease from speech using voice source information. IEEE/ACM Transactions on Audio, Speech, and Language Processing, 29, 1925-1936.

Appendix 1 (code):

```
import numpy as np
import pandas as pd
import os, sys
from sklearn.preprocessing import MinMaxScaler
from xgboost import XGBClassifier
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score

df=pd.read_csv('C:\\Users\\DELL\\Downloads\\Parkinsson disease.csv')
df.head()

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

print(labels[labels==1].shape[0], labels[labels==0].shape[0])

scaler=MinMaxScaler((-1,1))
x=scaler.fit_transform(features)
y=labels

x_train,x_test,y_train,y_test=train_test_split(x, y, test_size=0.2, random_state=7)
```

```
model=XGBClassifier()

model.fit(x_train,y_train)

y_pred=model.predict(x_test)
print(accuracy_score(y_test, y_pred)*100)

corr_matrix = df.corr()
fig, ax = plt.subplots(figsize=(30, 30))
ax = sns.heatmap(corr_matrix,
                 annot=True,
                 linewidths=0.5,
                 fmt=".2f",
                 cmap="YlGnBu");
bottom, top = ax.get_ylim()
ax.set_ylim(bottom + 0.5, top - 0.5)

df.status.value_counts().plot(kind="bar", color=["yellow", "teal"])
```

Appendix 2 (code):

```
import os
import cv2
import numpy as np
from skimage import feature
import random
import matplotlib.pyplot as plt
from xgboost import XGBClassifier
from sklearn.ensemble import RandomForestClassifier
%matplotlib inline

def quantify_image(image):
    features = feature.hog(image, orientations=9,
                           pixels_per_cell=(10, 10), cells_per_block=(2, 2),
                           transform_sqrt=True, block_norm="L1")
    return features
```

```
def load_split(path):
    # grab the list of images in the input directory, then initialize
    # the list of data (i.e., images) and class labels
    imagePaths = list(paths.list_images(path))
    data = []
    labels = []
    # loop over the image paths
    for imagePath in imagePaths:
        # extract the class label from the filename
        label = imagePath.split(os.path.sep)[-2]
        # load the input image, convert it to grayscale, and resize
        # it to 200x200 pixels, ignoring aspect ratio
        image = cv2.imread(imagePath)
        image = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
        image = cv2.resize(image, (200, 200))
        # threshold the image such that the drawing appears as white
        # on a black background
        image = cv2.threshold(image, 0, 255,
                              cv2.THRESH_BINARY_INV | cv2.THRESH_OTSU)[1]
        # quantify the image
        features = quantify_image(image)
        # update the data and labels lists, respectively
        data.append(features)
        labels.append(label)
    return (np.array(data), np.array(labels))

!pip install imutils
from imutils import paths
from sklearn.preprocessing import LabelEncoder
from sklearn.ensemble import RandomForestClassifier
from xgboost import XGBClassifier
from sklearn.metrics import confusion_matrix

def train_models(dataset):
    # initialize the models
    models = {
        "Rf": {
            "classifier": RandomForestClassifier(random_state=1),
            "accuracy": 0,
```

```
            "sensitivity": 0,
            "specificity": 0,
        },
        "Xgb": {
            "classifier": XGBClassifier(),
            "accuracy": 0,
            "sensitivity": 0,
            "specificity": 0,
        }
    }
    # define the path to the testing and training directories
    path = "C:\\Users\\DELL\\Downloads\\drawings\\" + dataset
    trainingPath = os.path.sep.join([path, "training"])
    testingPath = os.path.sep.join([path, "testing"])
    # load the data
    (trainX, trainY) = load_split(trainingPath)
    (testX, testY) = load_split(testingPath)
    # encode the labels
    le = LabelEncoder()
    trainY = le.fit_transform(trainY)
    testY = le.transform(testY)

    # train each model and calculate its metrics
    for model in models:
        models[model]["classifier"].fit(trainX, trainY)
        predictions = models[model]["classifier"].predict(testX)
        cm = confusion_matrix(testY, predictions).ravel()
        tn, fp, fn, tp = cm
        models[model]["accuracy"] = (tp + tn) / float(cm.sum())
        models[model]["sensitivity"] = tp / float(tp + fn)
        models[model]["specificity"] = tn / float(tn + fp)

    return models

spiralModels = train_models('spiral')
waveModels = train_models('wave')

print("Random Forrest vs XGBoost Classifier\n\n")
for metric in ("accuracy", "sensitivity", "specificity"):
```

```
    print(f"{metric.capitalize()}: ")
    print("Random Forrest={:.2f}%, XGBoost={:.2f}% \n".format(
        spiralModels['Rf'][metric]*100, spiralModels['Xgb'][metric]*100))

print("Random Forrest vs XGBoost Classifier\n\n")
for metric in ("accuracy", "sensitivity", "specificity"):
    print(f"{metric.capitalize()}: ")
    print("Random Forrest={:.2f}%, XGBoost={:.2f}% \n".format(
        waveModels['Rf'][metric]*100, waveModels['Xgb'][metric]*100))

def test_prediction(model, testingPath):
    # get the list of images
    testingPaths = list(paths.list_images(testingPath))
    output_images = []
    # pick 15 images at random
    for _ in range(15):
        image = cv2.imread(random.choice(testingPaths))
        output = image.copy()
        output = cv2.resize(output, (128, 128))
        # pre-process the image
        image = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
        image = cv2.resize(image, (200, 200))
        image = cv2.threshold(image, 0, 255,
                        cv2.THRESH_BINARY_INV | cv2.THRESH_OTSU)[1]
        # quantify the image and make predictions based on the extracted features
        features = quantify_image(image)
        preds = model.predict([features])
        label = "Parkinsons" if preds[0] else "Healthy"

        # draw the colored class label on the output image and add it to
        # the set of output images
        color = (58,228,132) if label == "Healthy" else (255, 0, 0)
        cv2.putText(output, label, (3, 20), cv2.FONT_HERSHEY_SIMPLEX, 0.5,
                color, 2)
        output_images.append(output)
    plt.figure(figsize=(20, 20))
    for i in range(len(output_images)):
        plt.subplot(5, 5, i+1)
        plt.imshow(output_images[i])
```

```
        plt.axis("off")
    plt.show()

testingPath = os.path.sep.join(['C:\\Users\\DELL\\Downloads\\drawings\\spiral', 'testing'])
test_prediction(spiralModels['Rf']['classifier'], testingPath)

testingPath = os.path.sep.join(["C:\\Users\\DELL\\Downloads\\drawings\\wave", "testing"])
test_prediction(waveModels['Rf']['classifier'], testingPath)
```