iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 10 **Issue:** VI **Month of publication:** June 2022

DOI: https://doi.org/10.22214/ijraset.2022.44075

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 10 Issue VI June 2022- Available at www.ijraset.com

Parkinsons Disease Prediction Using Machine Learning

Mohesh. T[1], Gowtham. K[2], Vijeesh. P[3], Arun Kumar. S[4]
[1, 2, 3, 4]*Bachelor of Engineering, Computer Science and Engineering, Anna University*

Abstract: Diagnosis of Parkinson's disease (PD) is commonly based on medical observations and assessment of clinical signs, including the characterization of a variety of motor symptoms. However, traditional diagnostic approaches may suffer from subjectivity as they rely on the evaluation of movements that are sometimes subtle to human eyes and therefore difficult to classify, leading to possible misclassification. In the meantime, early non-motor symptoms of PD may be mild and can be caused by many other conditions. Therefore, these symptoms are often overlooked, making diagnosis of PD at an early stage challenging. To address these difficulties and to refine the diagnosis and assessment procedures of PD, machine learning methods have been implemented for the classification of PD and healthy controls or patients with similar clinicalpresentations.

I. INTRODUCTION

Parkinson's Disease creates neural system disorder for various people. The diseaseaffects the people at different age groups around the world. Medical research works collaborate with computational intelligence techniques for predicting Parkinson symptoms. PD has numerous types based on the human abnormalities.

Mostly it disturbs the nature of neural activities and the body movements. Researches evolvedin recent years use Machine Learning (ML) and Deep Learning (DL) approaches for finding early stages of PD. The research works used different types of medical observations such as voice levels, handwriting variations, body movements, brain signal variations and protein aggregations. These kinds of observations are measured using various medical apparatuses.

A. Objective

The main objective of this article is to recognize what is Parkinson's sickness and to discover the early onset of the disorder. We will use here XGBoost, Random wooded area algorithm, guide Vector Machines (SVMs), and other gadget learning Algorithms make use of the data-set available on UCL Parkinson Data-set.

https://archieve.ics.uci.edu/ml/datasets/Parkinsonpercent27s+sickness+class.

B. Motivation Of The Project

Parkinson illnesses are the most vital reasons of dying and disability worldwide. According to the Parkinson sickness foundation, The affected peoples within the global of Parkinson ailment is projected that the 1 million humans are living by way of 2020 in united states of America. The scientific treatment of Parkinson disorder may be recommended on Neuropathologic and Histopathologic. Medical diagnostic detection of Parkinson sickness can be completed on enormous choice basing on the sensitivity and specificity of the feature Parkinson disease capabilities consequently, the Parkinson sickness are needed to explore the medical, pathologic, and nosology research grounded on frequency of prevalence, traits, and including hazard elements of samples. Parkinson typically affects a huge part of global sufferers over the age of fifty, which has affected to date still now there's no recognised cause of Parkinson disease, however, it is very likely possible to soothe signs and symptoms knowingly within the early stage of the subjective patients.

A have a look at claimed that around ninety% of the sufferers affected with vocal damage. The Parkinson remedy is in all likelihood very expensive.

This reasons most of the sufferers cannot have the funds for the value of the Parkinson sickness. because if the ailment is detected in the preliminary degree, then the cost will lower and it will additionally be viable to store the patient's lifestyles these days, Parkinson disorder prediction is maximum crucial rely for clinical practitioners to take correct decision of such ailment. It's a exquisite exercise at gift time, gadget learning based totally giant platform can detect Parkinson ailment.

©IJRASET: All Rights are Reserved

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 10 Issue VI June 2022- Available at www.ijraset.com

C. Scope Of The Project

In recent, machine learning knowledge of algorithms have generated a great impact and commitment in the Parkinson studies network for detection of Parkinson disorder. furthermore, system getting to know techniques are distinctive greater precise outcomes in disease prediction in comparison to others facts taxonomy techniques. prompted by way of this, the authors have used three distinguished device studying algorithms for detection and right diagnosis of Parkinson sufferers. the main purpose of this task is to look at the overall performance size of various distinguished class methods for this assignment a couple of machine studying techniques were used which includes support Vector device, Logistics Regression and different system getting to know algorithms. moreover, the overall performance of the 3 classifiers become evaluated the use of exceptional methods.

D. Proposed System

By using machine learning techniques, the hassle can be solved with minimum mistakes price. The voice dataset of Parkinson's ailment from the UCI device mastering library is used as input. additionally our proposed device presents correct results with the aid of integrating spiral drawing inputs of regular and Parkinson's affected sufferers. We endorse a hybrid and accurate results reading affected person both voice and spiral drawing data's. for that reason combining each the effects, the medical doctor can finish normality or abnormality and prescribe the medicine based totally at the affected stage.

E. Overview of The Project

Parkinson's disease is a revolutionary nervous system sickness that impacts movement. signs and symptoms begin progressively, now and again starting with slightly substantive tremor in only one hand. Tremors are not unusual, however he disorder additionally generally causes stiffness or slowing of movement. within the early degrees of Parkinson's ailment, your face may display very little expression. Your palms may not swing while you stroll. Your speech may additionally turn out to be gentle or slurred. Parkinson's sickness signs worsen as your condition progresses through the years. Although Parkinson's disorder can't be cured, medicines would possibly appreciably enhance your signs and symptoms. occasionally, your doctor may additionally recommend surgery to regulate certain regions of your brain and enhance your signs and symptoms.

F. Problem Definition

Parkinson's disorder (PD) is a progressive sickness with a presymptomatic interval; this is, there is a length all through which the pathologic technique has started, but motor symptoms required for the clinical prognosis are absent. in spite of fascinating leads, direct testing of preclinical markers has been constrained, in particular because there is no reliable manner to identify preclinical disorder. Idiopathic RBD is characterized via loss of regular atonia with REM sleep. approximately 50% of affected individuals will increase PD or dementia inside 10 years.

II. LITERATURE REVIEW

Dr. Anupam Bhatia and Raunak Sulekh (2019), "Predictive Model for Parkinson'sDisease through Naive Bayes Classification" In this study, Naive Bayes was appliedto predict the performance of the dataset. Rapid miner 7.6.001 is a tool, which wasused to explore, statistically analyze, and mine the data. The Naive Bayes model performs with 98.5 % accuracy, and 99.75% of precision.

Carlo Ricciardi, et al (2019), "the use of gait analysis' parameters to categorise Parkinsonism: A information mining method" in this gadget, Random woodland is used for class in conjunction with comparing it with Gradient Boosted trees. these consequences are being categorized into three special categories specifically PSP, De Novo Parkinson's disorder and stable Parkinson's disease with their accuracy being as high as 86.4% in comparison to Gradient Boosted trees which have been correct to a meagre 70%. additionally the precision price of Random forest changed into most of ninety % against Gradient Boosted trees which were around most of 85%.

Researches have been carried out on PD detection strategies the usage of medical and computational biology strategies. Sadek et al (2019) designed and carried out artificial Neural network (ANN) and lower back Propagation (BP) techniques for predicting PD signs and symptoms. ANN and BP techniques have been used for figuring out continuous patient actions. The paintings used ANN primarily based pattern matching techniques. The patterns of various affected person movements have been registered and trained for most reliable disorder evaluation. The paintings introduced ideal PD analysis results on homogeneous scientific functions on the identical time, this paintings were given limitations in multi-variation characteristic evaluation.

©IJRASET: All Rights are Reserved

Gao et al. (2019) proposed a specific prediction and classification technique for Parkinson data analysis. This work was implemented with the help of data preprocessing techniques, cross fold validations, and ML approaches. The neuro data features and the tremor data features were analyzed to predict the symptoms. The methods used in this work provided valuable results but lacked in PD detection and sensitivity rate. Many research works were inherited with different perspectivesfor predicting PD.

In this manner, Mostafa et al. (2020) determined the answer the use of multi-agent facts analysis device. The multi-agent device turned into designed for evaluating vocal issues. affected person's vocal statistics were diagnosed as primary functions for disease detection. The vocal variations were analyzed with the assist of Reinforcement gaining knowledge of (RL), choice Tree (DT), Naïve Bayes category and Random forest (RF) techniques. This paintings gathered the medical facts from Tel Aviv Sourasky medical middle however, the paintings became lagging with real-time problems. Biswajt et al. (2020) proposed the system to locate D symptoms the usage of speech attributes. on this work, SVM and Naïve Byes strategies have been mentioned the use of voice information features. the radical paintings built ML primarily based voice evaluation models for locating the PD symptoms. The work tried to task extra correct outcomes. however, this work lacked with the constrained dataset features.

Rastegari et al (2021) evolved data gain evaluation model for detecting PD capabilities from the given dataset. This technique used diverse ML and data gain approached in mixed manner. This method labored well in PD findings. however they produced insignificant effects as compared to DL primarily based PD evaluation models. Seppi et al. (2021) analyzed and carried out Parkinson treatment evaluation for various non-motor signs. This work cautioned that the remedy analysis allows to improve and update the future subsequent level treatments. on this regard, the paintings gathered the evidences of numerous treatments and produced valuable suggestions. Many systems created clinical review reviews on PD and remedies however those structures were no longer equipped with own specific techniques.

Espay et al. (2022) proposed new strategies for comparing Synuclein protein disorders and aggregation price for the detection of Parkinson and Alzheimer signs. specifically, the protein aggregation techniques used for sporadic Parkinson and Alzheimer. At that point, the gadget did not take DL and ML help to educate the protein capabilities. locating one of a kind styles of PD and predicting the signs and symptoms were virtually complicated tasks.

Bot et al. (2022) used cell datasets and mobile primarily based statistics collections for PD evaluation. on this work, the techniques have been educated to supply less overhead statistics functions on the detection of Parkinson. The techniques were correctly operated but restricted with scalability issues.

Bouwmans et al. (2022) passed through with Idiopathic PD (IPD) symptoms for assorted feature class Ramzi M. Sadek et al (2022), "Parkinson's disease Prediction the use of artificial Neural network" on this machine, 195 samples in the dataset had been divided into one hundred seventy training samples and 25 validating samples. Then importing the dataset inside the just Neural network (JNN) environment, we skilled, established the synthetic Neural community model. The maximum crucial attributes contributing to the ANN model were made known of. The ANN version changed into 90% correct

Satish Srinivasan, Michael Martin & Abhishek Tripathi (2020), "ANN based totally information Mining analysis of Parkinson's disorder" on this study, it was supposed to understand how the exceptional types of preprocessing steps should affect the prediction accuracy of the classifier. in the method of classifying the Parkinson's disorder dataset the use of the ANN based MLP classifier a appreciably high prediction accuracy changed into discovered while the dataset turned into pre-processed the usage of each the Discretization and Resample approach, both inside the case of 10-fold go validation and 80:20 split.

III. SOFTWARE REQUIRMENTS SPECIFICATION

A. Python

1) Python 3

B. Libraries

1) Numpy
2) pandas
3) joblib
4) sklearn
5) argparse
6) pickle
7) matplotlib, etc.

C. *Operating System*

1) Windows or Ubuntu

D. *Hardware Requirements Specification*

1) Laptop with basic hardware
2) Webcam

E. *Inputs*

1) pd_speech_features.csv

IV. SYSTEM ANALYSIS

A. *System Review*

This survey is executed to recognize the need and prerequisite of the general population who has PD, and to do as such, we went through exceptional sites and programs and searched for the essential statistics. based on these information, we made an audit that helped us get new thoughts and make distinctive arrangements for our challenge. We reached the choice that there's a need of such utility and felt that there's a respectable extent of development on this field too.

B. *Technology Used*

1) *Python:* Python is an interpreted, high-level, general-purpose programming language. Python's design philosophy emphasizes code readability with its notable use of significant whitespace. Its language constructs and object- oriented approach aim to help programmers write clear, logical code for small andlarge-scale projects. Python is dynamically typed AND supports multiple programming paradigms, including procedural, object-oriented, and functional programming.
2) *Pycharm:* Pycharm makes it easier for programmers to write various webapplications in python supporting widely used web technologies like HTML, CSS, JavaScript and CoffeeScript. It even simplifies isomorphic web application development by supporting both AngularJS and NodeJS
3) *machine learning:* Machine learning is the scientific study ofalgorithms and statistical models that computer systems use in order to perform a specific task effectively without using explicit instructions, relying on patterns andinference instead. It is seen as a subset of artificial intelligence.
4) *Machine Learning Datasets:* Machine learning has been developed for decades, and therefore there are some datasets of historical significance. One of the most well-known repositories for these datasets is the UCIMachine Learning Repository. Most of the datasets over there are small in size because the technology at the time was not advanced enough to handle larger size data. Some famous datasets located in this repository are the iris flower dataset

V. REQUIREMENT ANALYSIS

A. *Python*

Python is the basis of the program that we wrote. It utilizes many of thepython libraries.

B. *Libraries*

1) *NumPy:* NumPy, which stands for Numerical python, is a library consistingof multidimensional array objects and a collection of routines for processing those arrays. Using NumPy mathematical and logical operations on arrays can be performed. NumPy is a python Pre-requisite for Dlib.
2) *Pandas:* Pandas is an open source Python package that is most widely used for data science/data analysis and machine learning tasks. It is built on top of another package named Numpy, which provides support for multi-dimensional arrays.
3) *Matplotlib:* Matplotlib is a cross-platform, data visualization and graphical plotting library for Python and its numerical extension NumPy. As such, it offers aviable open source alternative to MATLAB.
4) *Sklearn:* Scikit-learn is a free machine learning library for Python. It features various algorithms like support vector machine, random forests, and k-neighbours,and it also supports Python numerical and scientific libraries like NumPy and SciPy.
5) *Argparse:* The argparse module in Python helps create a program in a command-line-environment not only easy to code but also improves interaction. The argparse module also automatically generates help and usage messages and issues errors when users give the program invalid arguments.
6) *Joblib:* Joblib is a set of tools to provide lightweight pipelining in Python. Inparticular: transparent disk-caching of functions and lazy re-evaluation (memorize pattern) easy simple parallel computing.

©IJRASET: All Rights are Reserved

C. OS

Program is tested on Windows 10 build 1903 and PopOS 19.04

D. Classification Techniques

1) *Logistics Regression:* Logistic Regression changed into commonly used in the organic studies and packages in the early twentieth century. Logistic Regression (LR) is one of the maximum used device learning algorithms that is used wherein the goal variable is categorical. lately, LR is a famous method for binary classification troubles. moreover, it presents a discrete binary product between zero and 1. Logistic Regression computes the relationship among the feature variables by means of assessing possibilities (p) the use of underlying logistic function.
2) *Support Vector Machine (SVM):* guide vector system has been first added via Vladimir Vapnik and Alexey Chervonenkis. SVM is a way of device studying that can clear up both linear and nonlinear issues. It presents exact overall performance to remedy each regression and classification hassle. The SVM classification technique inspects for the highest quality separable hyperplane if you want to classify the dataset between two instructions. eventually, the model can estimate noisy information troubles for brand new instances
3) *Ada Boost:* AdaBoost combines the predictions from brief one-degree decision timber, referred to as selection stumps, despite the fact that other algorithms also can be used. selection stump algorithms are used as the AdaBoost set of rules seeks to apply many susceptible fashions and correct their predictions with the aid of including additional weak models. The education algorithm includes beginning with one choice tree, locating those examples within the training dataset that have been misclassified, and including more weight to the ones examples. another tree is educated at the same records, despite the fact that now weighted with the aid of the misclassification mistakes. This technique is repeated until a preferred variety of trees are introduced
4) *Gradient Boost:* Unlike, Adaboosting set of rules, the base estimator within the gradient boosting algorithm can not be cited through us. the bottom estimator for the Gradient raise algorithm is fixed and i.e. decision Stump. Like, AdaBoost, we will music the n_estimator of the gradient boosting set of rules. but, if we do not point out the fee of n_estimator, the default price of n_estimator for this algorithm is 100.Gradient boosting set of rules can be used for predicting not simplest non-stop goal variable (as a Regressor) but also specific target variable (as a Classifier). while it's far used as a regressor, the value characteristic is suggest square errors (MSE) and while it's far used as a classifier then the cost function is Log loss
5) *Random Forest:* Random woodland is a Supervised device mastering algorithm this is used widely in classification and Regression troubles. It builds choice bushes on extraordinary samples and takes their majority vote for type and average in case of regression. one of the most vital functions of the Random woodland set of rules is that it can manage the information set containing continuous variables as within the case of regression and specific variables as within the case of category. It plays higher outcomes for category issues
6) *Naive Bayes:* Naive Bayes algorithm is a supervised learning algorithm, which is based on Bayes theorem and used for solving classification problems. It ismainly used in text classification that includes a high-dimensional training dataset. Naïve Bayes Classifier is one of the simple and most effective Classification algorithms which helps in building the fast machine learning models that can makequick predictions. It is a probabilistic classifier, which means it predicts on the basisof the probability of an object. Some popular examples of Naïve Bayes Algorithm are spam filtration, Sentimental analysis, and classifying articles.
7) *Neural Network:* Neural networks, also referred to as artificial neural networks (ANNs) or simulated neural networks (SNNs), are a subset of gadget gaining knowledge of and are on the heart of deep learning algorithms. Their call and structure are stimulated via the human brain, mimicking the way that organic neurons sign to one another. synthetic neural networks (ANNs) are constructed from a node layers, containing an input layer, one or extra hidden layers, and an output layer. every node, or synthetic neuron, connects to another and has an associated weight and threshold. If the output of any individual node is above the required threshold value, that node is activated, sending facts to the next layer of the community. in any other case, no records is exceeded alongside to the subsequent layer of the community
8) *XGBoost:* XGBoost, which stands for severe Gradient Boosting, is a scalable, disbursed gradient-boosted choice tree (GBDT) device mastering library. It gives parallel tree boosting and is the leading machine mastering library for regression, class, and ranking issues. It’s important to an understanding of XGBoost to first draw close the system getting to know standards and algorithms that XGBoost builds upon: supervised device mastering, choice bushes, ensemble getting to know, and grading raise. Supervised machine learning uses algorithms to educate a version to find patterns in a dataset with labels and functions

and then uses the educated model to are expecting the labels on a brand new dataset's functions.

9) *Decision Tree:* The decision of making strategic splits heavily impacts a tree's accuracy. The choice criteria are exceptional for type and regression timber. decision timber use a couple of algorithms to decide to break up a node into or greater sub-nodes. The introduction of sub-nodes will increase the homogeneity of resultant sub- nodes. In other phrases, we will say that the purity of the node increases with admire to the target variable. The decision tree splits the nodes on all available variables after which selects the break up which results in most homogeneous sub-nodes.

E. Laptop

Used to run our code.

VI. SYSTEM DESIGN

A. Architectural Diagram

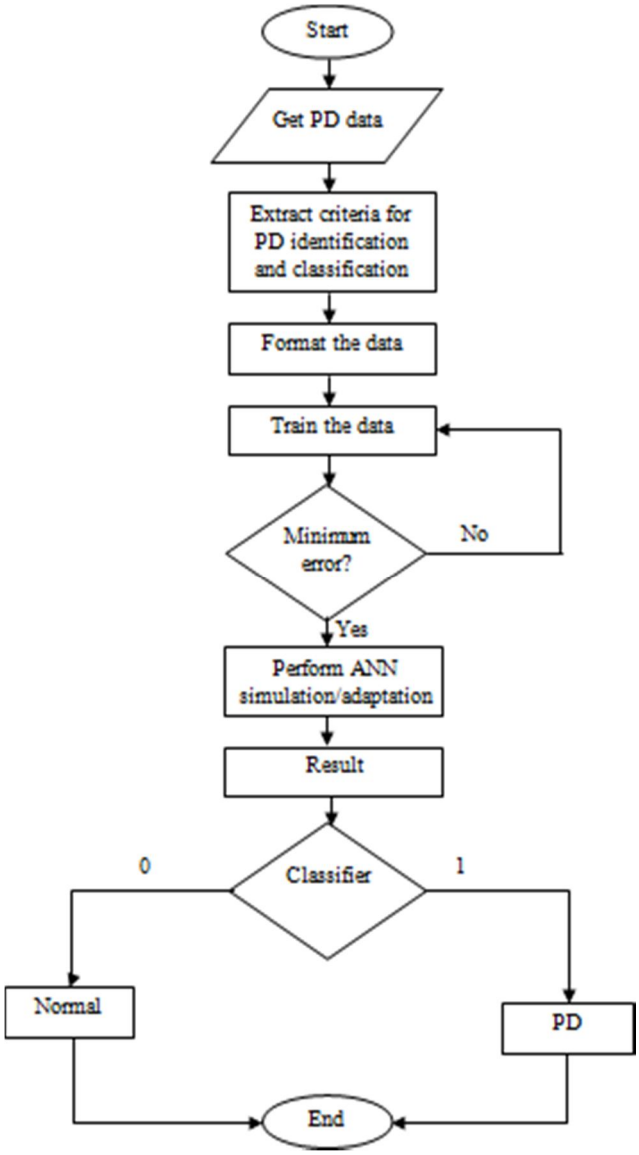


FIG 6.1 Architectural Diagram

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 10 Issue VI June 2022- Available at www.ijraset.com

VII. IMPLEMENTATION

A. *Parkinson's Disease Prediction*

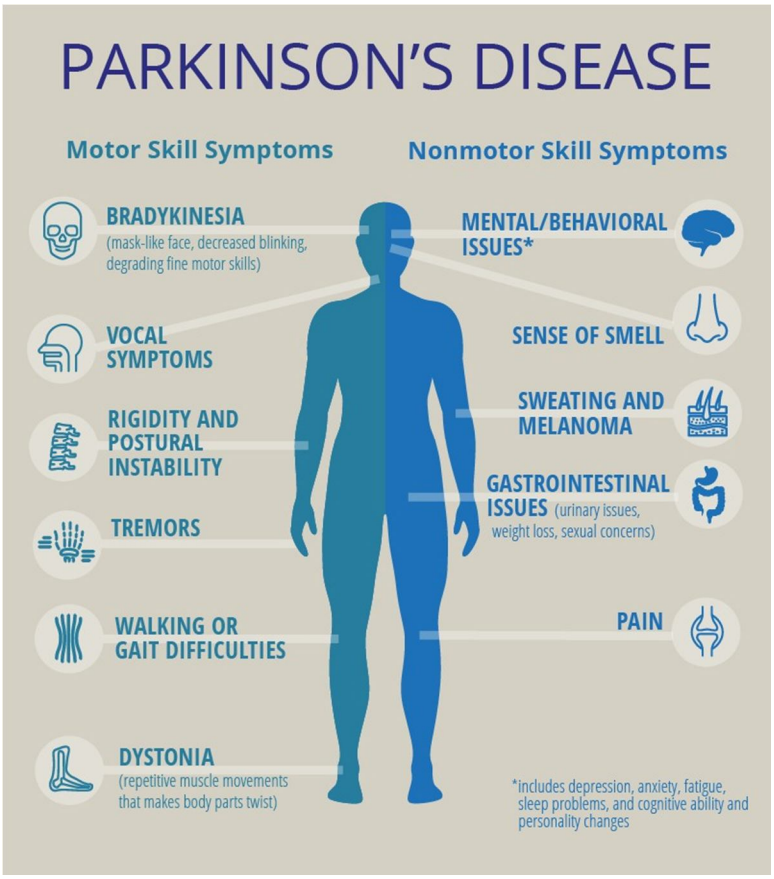


FIG 7.1 Parkinson Disease

B. *Running Training Models*

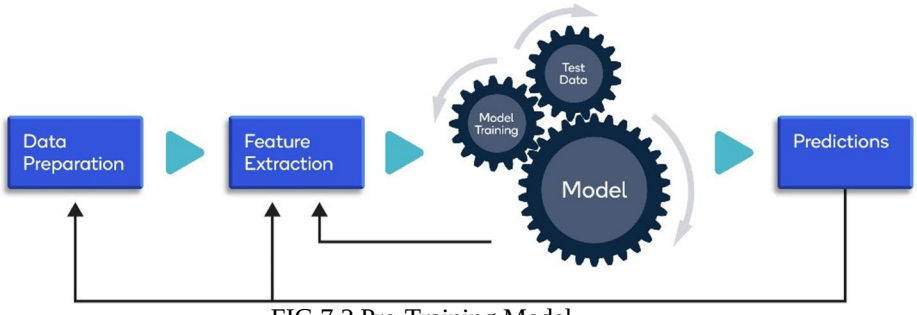


FIG 7.2 Pre-Training Model

C. *See run_model.py*

1) Use python run_model.py --help to see the complete list of trained models.

2) Choose a model with the python run_model.py --model {model_number} to use thepretrained model on the parkinsons's disease dataset.

TO RUN

- EDA.py to see the data analysis visual plots.
- generate_weights.py to save weights for models with best hyperparameters.
- run_model.py to see results on the saved weights.

©IJRASET: All Rights are Reserved

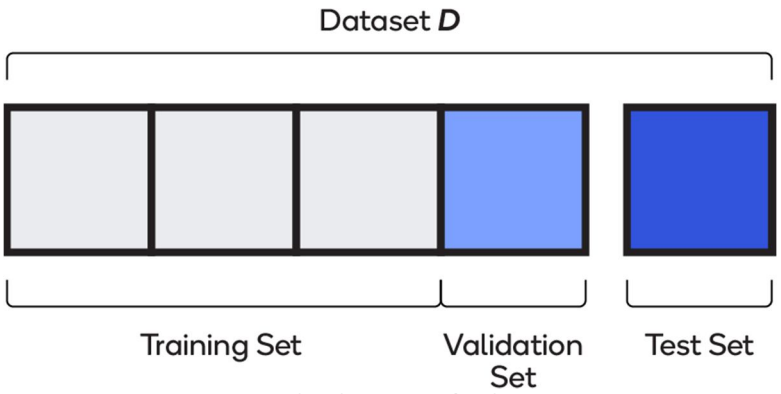


FIG 7.3 Traning ANG Testing

D. Generating Weights

See generate_weights.py

1) Use python generate_weights.py --help to see the complete list of trained models.

2) Choose a model with the python generate_weights.py --model {model_number} togenerate fresh set of weights from best_params.py.

Name	Description
run_model.py	Run Model from pre-trainedweights
best_params.py	Get Models for besthyperparameters
generate_weights.py	Save Weights for best models
pre_processing.py	PCA, Feature Selection, train-testsplit
EDA.py	Exploratory Data Analysis ondataset
weights_original	Weights from the Original Project

VIII. PROGRAM AND SCREENSHOTS OF PROJECT

A. best_params.py

```
from sklearn.naive_bayes import BernoulliNB
from sklearn.linear_model import LogisticRegression
from sklearn.ensemble import RandomForestClassifier, GradientBoostingClassifier
from sklearn.tree import DecisionTreeClassifier
from xgboost import XGBClassifier
from sklearn.svm import SVC
from sklearn.ensemble import AdaBoostClassifier
from sklearn.model_selection import GridSearchCV
from pre_processing import X_train, X_test, y_test, y_train
from sklearn.neural_network import MLPClassifier
```

```
# best_params file returns the best_models tuned to best performing parameters

best_models = {1: "naive_bayes", 2: "logistic_regression", 3: "decision_tree",
                        4: "svm", 5: "random_forest", 6: "ada_boost", 7: "gradient_boost", 8: "xg_boost",
9: "neural_net"}


def best_model(model_num):
    # Naive Bayes
    if model_num == 1:
        clf = BernoulliNB()
        clf.fit(X_train, y_train)
        return clf


    # Logistic Regression
    if model_num == 2:
        param_grid = {'penalty': ['l1', 'l2'],
                      'C': [0.001, 0.01, 0.1, 1, 10, 100, 1000]}
        grid = GridSearchCV(LogisticRegression(random_state=0),
                            param_grid, refit=True)
        grid.fit(X_train, y_train)
        return grid

    # decision_tree

    if model_num == 3:
        param_grid = {'criterion': ['gini', "entropy"],
                      'splitter': ["best", "random"],
                      'max_depth': range(1, 20),
                      'min_samples_split': range(2, 10, 1)}
        grid = GridSearchCV(DecisionTreeClassifier(random_state=0),
                            param_grid, refit=True)
        grid.fit(X_train, y_train)
        return grid


    # SVM
    if model_num == 4:
        param_grid = {'C': [0.1, 1, 10, 100, 1000],
                      'gamma': [1, 0.1, 0.01, 0.001],
                      'kernel': ['rbf', 'poly', 'sigmoid']
                      }
        grid = GridSearchCV(SVC(random_state=0), param_grid, refit=True)
        grid.fit(X_train, y_train)
        return grid
```

```
# Random Forest
if model_num == 5:
    param_grid = {'n_estimators': range(10, 100, 5),
                  'criterion': ['gini', 'entropy'],
                  'max_depth': range(1, 20),
                  'min_samples_split': range(2, 10, 1),
                  'max_features': ['auto', 'sqrt', 'log2']
                  }
    grid = GridSearchCV(RandomForestClassifier(random_state=0, n_jobs=10),
                                               param_grid, refit=True, n_jobs=10)
    grid.fit(X_train, y_train)
    return grid

# ADA Boost
if model_num == 6:
    param_grid = {'n_estimators': range(10, 100, 5),
                  'algorithm': ['SAMME', 'SAMME.R'],
                               'learning_rate': [0.001, 0.01, 0.1, 1.0, 10.0]
                  }
    grid = GridSearchCV(AdaBoostClassifier(random_state=0),
                        param_grid, refit=True)
    grid.fit(X_train, y_train)
    return grid

# Gradient Boost
if model_num == 7:
    param_grid = {'n_estimators': [100, 150, 200, 250, 300, 350, 400, 450, 500],
                  'criterion': ['friedman_mse', 'mse'],
                  'max_features': ['sqrt', 'log2'],
                               'learning_rate': [0.001, 0.01, 0.1, 1.0, 2, 4],
                  'loss': ['deviance', 'exponential'],
                  }
    grid = GridSearchCV(GradientBoostingClassifier(
        random_state=0), param_grid, refit=True)
    grid.fit(X_train, y_train)
    return grid
# XG Boost
if model_num == 8:
    param_grid = {'n_estimators': range(0, 100, 5),
                  'learning_rate': [0.01, 0.1],
                  'max_depth': range(2, 6)
                  }
```

```
            grid = GridSearchCV(XGBClassifier(random_state=0),
                                param_grid, refit=True)
            grid.fit(X_train, y_train)
            return grid


        # Neural Net
        if model_num == 9:
            param_grid = {'alpha': [x * 0.0001 for x in range(1, 1000, 2)],
                          'learning_rate_init': [0.01, 0.001, 0.0001],
                          'max_iter': [100, 300, 1000],
                          'hidden_layer_sizes': [(128, 64, 32), (128, 32, 8), (256, 128, 64, 32), (64,
32, 16), (64, 32), (64, 16)],
                          }
            grid = GridSearchCV(MLPClassifier(random_state=42),
                                param_grid, refit=True, n_jobs=10)
            grid.fit(X_train, y_train)
            return grid
```

B. generate_weights.py

```
import argparse
import pickle
import sys
from best_params import best_model


if__name__== "_main_":
    # Adding Arguments to make user select one model to run results on
    parser = argparse.ArgumentParser(
        formatter_class=argparse.RawTextHelpFormatter)
    model_dict = {1: "naive_bayes", 2: "logistic_regression", 3: "decision_tree",
                  4: "svm", 5: "random_forest", 6: "ada_boost", 7: "gradient_boost", 8: "xg_boost",
9: "neural_net", 10: "all"}
    parser.add_argument(
        "-m", "--model", type=int, choices=list(range(1, len(model_dict)+1)), help="\
            \n1. Naive Bayes \
            \n2. Logistic Regression \
            \n3. Decision Tree\
            \n4. SVM\
            \n5. Random Forest\
            \n6. AdaBoost\
            \n7. Gradient Boost\
            \n8. XGBoost\
            \n9. Neural Network\
            \n10. All Models\
            ")

    args = parser.parse_args()
```

```
    if not args.model:
        print("Model not provided. See run_model.py --help")
        sys.exit()

    def save_weights(model_num):
        model = best_model(model_num)
        filename = f"./weights/{model_dict[model_num]}.sav"
        print(f"\nSaving {filename} ...")
        pickle.dump(model, open(filename, 'wb'))

    # save the model to disk
    if(model_dict[args.model] != "all"):
        save_weights(args.model)
    else:
        # Run for all models
        for model_num in range(1, len(model_dict)):
            save_weights(model_num)
else:
    print("Run the script run_model.py to obtain results.")
```

C. pre_processing.py

```
import pandas as pd
import numpy as np
from sklearn.feature_selection import f_classif, SelectKBest
from sklearn.preprocessing import StandardScaler
from sklearn.decomposition import PCA


df = pd.read_csv('./dataset/pd_speech_features.csv')
id = df['id']
label = df['class']
n = len(id)
id_label = {}
for i in range(n):
    id_label[id[i]] = label[i]


train_healthy_id = []
test_healthy_id = []
train_pd_id = []
test_pd_id = []
train_n_healthy = 58
test_n_healthy = 6
train_n_pd = 170
test_n_pd = 18


train_count_healthy = 0
```

```
test_count_healthy = 0
train_count_pd = 0
test_count_pd = 0
for key, val in id_label.items():
    if val == 0:
                    if train_count_healthy < train_n_healthy:
            train_healthy_id.append(key)
            train_count_healthy += 1
        else:
            test_healthy_id.append(key)
            train_count_healthy += 1
    else:
        if train_count_pd < train_n_pd:
            train_pd_id.append(key)
            train_count_pd += 1
        else:
            test_pd_id.append(key)
            test_count_pd += 1


train_healthy_id.extend(train_pd_id)
test_healthy_id.extend(test_pd_id)
train_id = train_healthy_id
test_id = test_healthy_id
train_id = np.array(train_id)
test_id = np.array(test_id)


trainX = None
trainY = None
for i in range(train_id.shape[0]):
    vals = df[df['id'] == train_id[i]].values
    if i == 0:
        trainX = vals[:, 1:-1]
        trainY = vals[:, -1].reshape((-1, 1))
    else:
        trainX = np.append(trainX, vals[:, 1:-1], axis=0)
        trainY = np.append(trainY, vals[:, -1].reshape((-1, 1)), axis=0)


testX = None
testY = None
for i in range(test_id.shape[0]):
    vals = df[df['id'] == test_id[i]].values
    if i == 0:
        testX = vals[:, 1:-1]
```

```
            testY = vals[:, -1].reshape((-1, 1))
        else:
            testX = np.append(testX, vals[:, 1:-1], axis=0)
            testY = np.append(testY, vals[:, -1].reshape((-1, 1)), axis=0)
    trainY = trainY.reshape((-1,))


    trainX_copy = trainX
    trainY_copy = trainY
    testX_copy = testX
    testY_copy = testY


    sel = SelectKBest(f_classif, k=490)
    trainX_copy = sel.fit_transform(trainX_copy, trainY_copy)
    testX_copy = sel.transform(testX_copy)


    sc = StandardScaler()
    trainX_copy = sc.fit_transform(trainX_copy)
    testX_copy = sc.transform(testX_copy)


    pca = PCA(n_components=0.99)
    trainX_copy = pca.fit_transform(trainX_copy)
    testX_copy = pca.transform(testX_copy)


    if__name__== "_main_":
        print("Data Pre-processed Succesfully. Run the script run_model.py")
    else:
        X_train, X_test, y_test, y_train = trainX_copy, testX_copy, testY_copy, trainY_copy
```

D. run_model.py

```
    import argparse
    import pickle
    import Joblib
    import sys
    # Loading the dataset
    from pre_processing import X_train, X_test, y_test, y_train


    # Adding Arguments to make user select one model to run results on
    parser = argparse.ArgumentParser(formatter_class=argparse.RawTextHelpFormatter)
    model_dict = {1: "naive_bayes", 2: "logistic_regression", 3: "decision_tree",
                  4: "svm", 5: "random_forest", 6: "ada_boost", 7: "gradient_boost", 8: "xg_boost", 9:
    "neural_net", 10: "all"}
```

```
parser.add_argument(
    "-m", "--model", type=int, choices=list(range(1, len(model_dict)+1)), help="\
        \n1. Naive Bayes \
        \n2. Logistic Regression \
        \n3. Decision Tree\
        \n4. SVM\
        \n5. Random Forest\
        \n6. AdaBoost\
        \n7. Gradient Boost\
        \n8. XGBoost\
        \n9. Neural Network\
        \n10. All Models\
        ")

args = parser.parse_args()
if not args.model:
    print("Model not provided. See run_model.py --help")
    sys.exit()


# Loading and Running the Model
def load_and_predict(model_num):
    filename = f"./weights/{model_dict[model_num]}.sav"
    print(f"\nLoading {filename} ...")

    loaded_model = pickle.load(open(filename, 'rb'))
    result = loaded_model.score(X_test, y_test)

    print(f"({model_dict[model_num]}) Model Accuracy: {result}%")


if(model_dict[args.model] != "all"):
    load_and_predict(args.model)
else:
    # Run for all models
    for model_num in range(1, len(model_dict)):
        load_and_predict(model_num)
```

E. EDA.py

```
from pre_processing import X_train, X_test, y_test, y_train
import pandas as pd
import numpy as np
```

```
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.manifold import TSNE


# Perform EDA
df = pd.read_csv('./dataset/pd_speech_features.csv')

print('Dataset Shape:')
print(df.shape)
print()

print('Class Distribution:')
print(df['class'].value_counts())
print()


positives = df[df['class'] == 1]
negatives = df[df['class'] == 0]

print('Positive Class (Patients with PD) Gender Distribution')
print(positives['gender'].value_counts())
print()

print('Negative Class (Patients without PD) Gender Distribution')
print(negatives['gender'].value_counts())
print()

# Number of patients counts


def print_patient_dist(df):
    ids = np.unique(df['id'].values)
    fem = 0
    male = 0
    for i in ids:
        if df[df['id'] == i]['gender'].iloc[0] == 0:
            fem += 1
        else:
            male += 1
    print('Unique Males: ', male)
```

```
    print('Unique Females: ', fem)


print('Positive Class Unique Patients Distribution')
print_patient_dist(positives)
print()
print('Negative Class Unique Patients Distribution')
print_patient_dist(negatives)
print()

print('Database Info')
print(df.info())
print()


# Plot distribution of 3 important features
df_selected = df[['DFA', 'mean_MFCC_2nd_coef', 'tqwt_entropy_log_dec_12']]

print(df_selected.describe())

plt.gcf().set_size_inches(12, 5)
ax = sns.histplot(df['DFA'], color='#3EADA7', alpha=1, edgecolor='white')
ax.set_title('DCA')
plt.show()

plt.gcf().set_size_inches(12, 5)
ax = sns.histplot(df['mean_MFCC_2nd_coef'],
                  color='#3EADA7', alpha=1, edgecolor='white')
ax.set_title('mean_MFCC_2nd_coef')
plt.show()

plt.gcf().set_size_inches(12, 5)
ax = sns.histplot(df['tqwt_entropy_log_dec_12'],
                  color='#3EADA7', alpha=1, edgecolor='white')
ax.set_title('tqwt_entropy_log_dec_12')
plt.show()


# Peform tsne visualization on the preprocessed data
tsne_results = TSNE(n_components=2, random_state=1).fit_transform(X_train)

DF = pd.DataFrame({'y': y_train.reshape((-1,))})
```

```
DF['tsne1'] = tsne_results[:, 0]
DF['tsne2'] = tsne_results[:, 1]


plt.figure(figsize=(5, 5))
ax = sns.scatterplot(
    x="tsne1", y="tsne2",
    hue="y",
    palette=sns.color_palette("hls", 2),
    data=DF,
    legend=False,
    alpha=1
)

ax.set_title('TSNE Visualization')
plt.show()
```

F. Screenshots

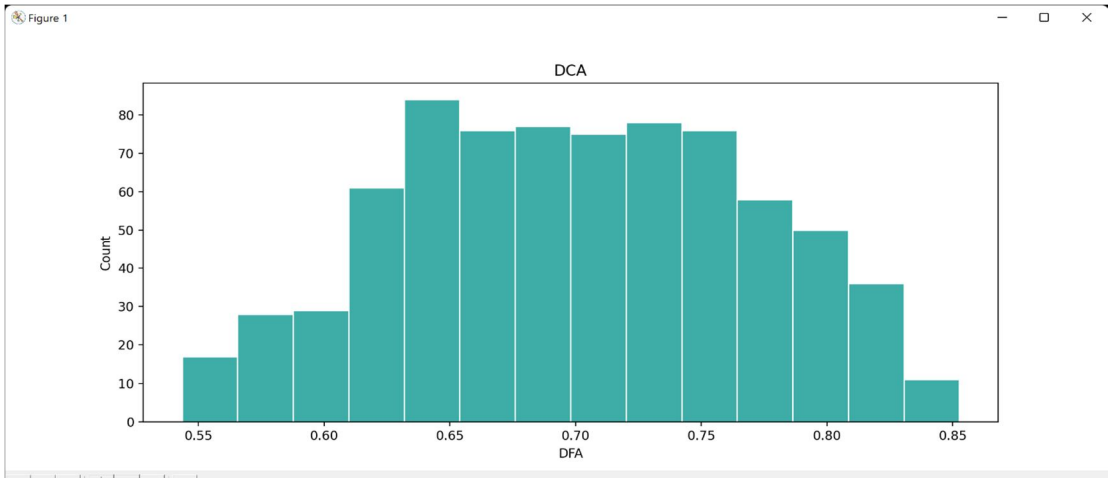


Fig 8.1 DCA

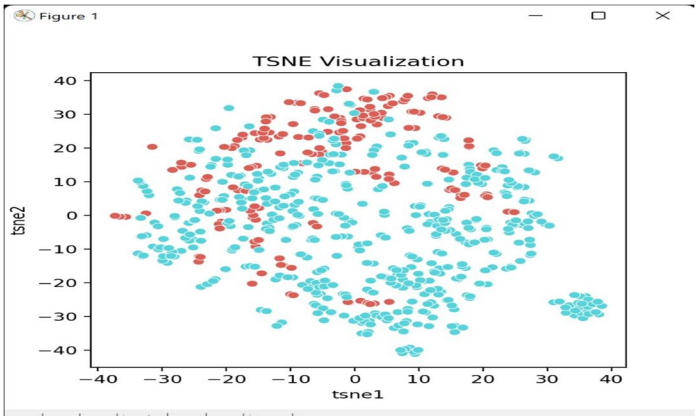


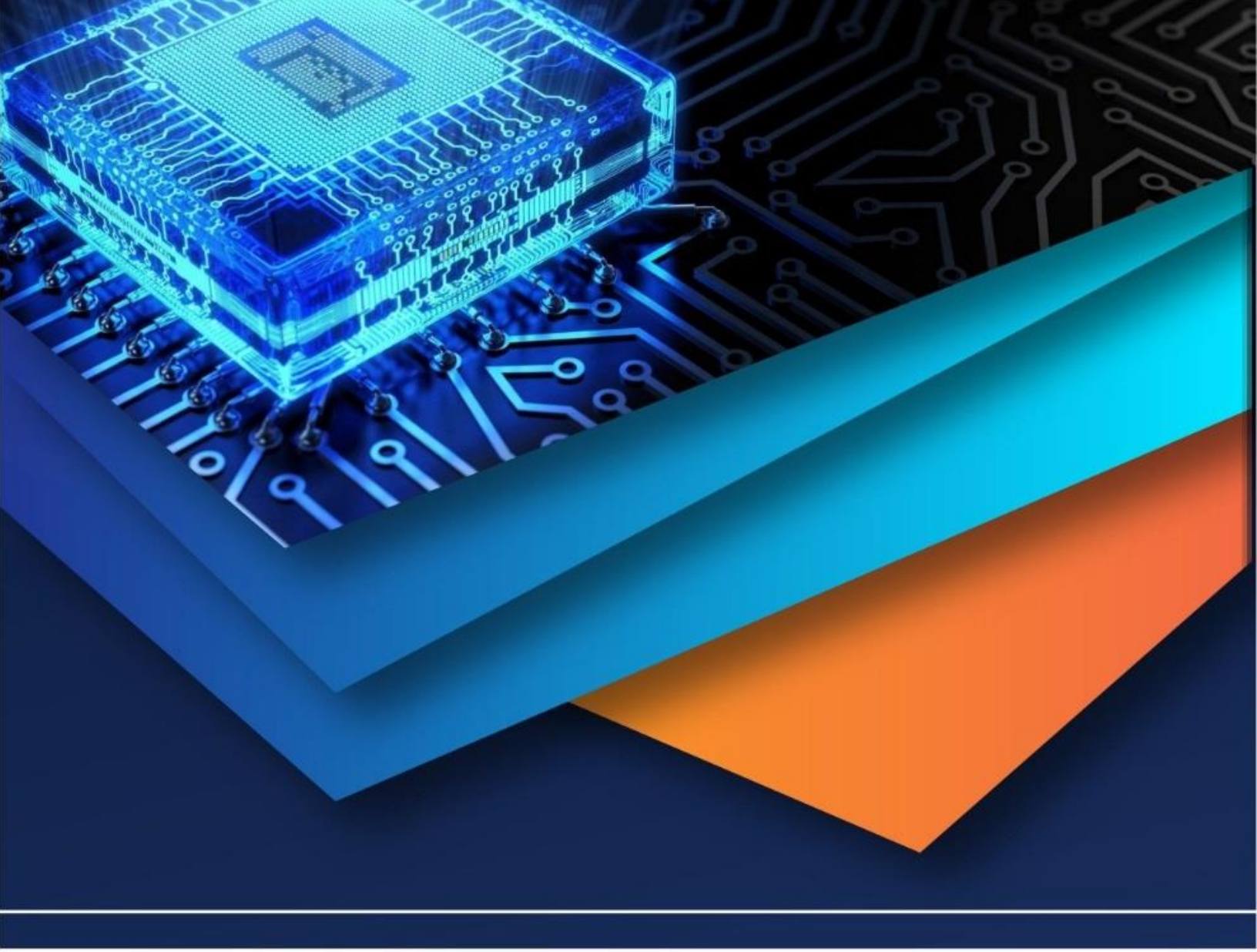
Fig 8.2 TSNE Visualization

IX. CONCLUSION

Parkinson's Disease is a totally grave disease and has no cure till date. since it impacts the actions of the parts of the body, the speech additionally stands affected. here, the gadget tries to offer a way of detecting Parkinson's ailment so one can bring about a quick action to reduce or even put off it from affecting the whole body. This gadget aims to make this method of expertise a case of Parkinson's on the earliest via each, the affected person as well as scientific experts. hence, the goal is to apply numerous machine getting to know strategies like SVM, choice Tree, for buying the maximum accurate result. Here using Decision Tree and building a classifier results in an accuracy of 94.7774555%.

REFERENCES

[1] Adrien Payan, Giovanni Montana, Predicting Alzheimer's disease: a neuroimaging study with 3D convolutional neural networks.
[2] Alemami, Y. and Almazaydeh, L. (2020) Detecting of ParkinsonDisease through Voice Signal Features. Journal of American Science,
[3] Fayao Liu, Chunhua Shen, Learning Deep Convolutional Features for MRI Based Alzheimer's Disease Classification.
[4] Hadjahamadi, A.H. and Askari, T.J. (2020) A Detection Support Systemfor Parkinson's Disease Diagnosis Using Classification and Regression Tree. Journal of Mathematics and Computer Science , 4, 257-263.
[5] Little, M.A., McSharry, P.E., Hunter, E.J. and Ramig, L.O. (2018), Suitability of Dysphonia Measurements for Telemonitoring of Parkinson'sdisease. IEEE Transactions on Biomed ical Engineering, 56, 1015-1022.
[6] Muhlenbach, F. and Rakotomalala, R. (2019) Discretization of Continuous Attributes. In: Wang, J., Ed., Encyclopedia of Data Warehousing and Mining, Idea Group Reference, 397-402.
[7] A. J. Espay, ''Revisiting protein aggregation as pathogenic in sporadicparkinson and alzheimer diseases,'' Neurology, vol. 92, no. 7, pp. 329–337,Feb. 2019.
[8] B. M. Bot, C. Suver, E. C. Neto, M. Kellen, A. Klein, C. Bare, M. Doerr, A. Pratap, J. Wilbanks, E. R. Dorsey, S. H. Friend, and A. D. Trister,''The mPower study, parkinson disease mobile data collected using ResearchKit,'' Sci. Data, vol. 3, no. 1, Dec. 2019, Art. no. 160011.
[9] A. E. P. Bouwmans, A. M. M. Vlaar, W. H. Mess, A. Kessels, and W.
[10] E. J. Weber, ''Specificity and sensitivity of transcranial sonography of the substantia nigra in the diagnosis of Parkinson's disease: Prospective cohortstudy in 196 patients,'' BMJ Open, vol. 3, no. 4, 2020, Art. no. e002613.
[11] Pickle, N. T., Shearin, S. M., and Fey, N. P. Dynamic neural network approach to targeted balance assessment of individuals with and without neurological disease during non-steady-state locomotion. Journal of Neuroengineering and Rehabilitation 16 (JUL 12 2020), 88. PT: J; NR: 31;TC: 0; J9: J NEUROENG REHABIL; PG: 9; GA: IJ0RY; UT: WOS:000475608600003.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)