

## **ABSTRACT**

Parkinson's disease (PD) is a neurodegenerative movement disease where the symptoms gradually develop start with a slight tremor in one hand and a feeling of stiffness in the body and it became worse over time. It affects over 6 million people worldwide. At present there is no conclusive result for this disease by non-specialist clinicians, particularly in the early stage of the disease where identification of the symptoms are very difficult in its earlier stages. The proposed predictive analytics framework is a combination of K-means clustering and Decision Tree which is used to gain insights from patients. By using machine learning techniques, the problem can be solved with minimal error rate. Voice data sets obtained from the UCI Machine learning repository is given as the input for voice data analysis. Also our proposed system provides accurate results by integrating spiral drawing inputs of normal and Parkinson's affected patients. From these drawings Random forest classification algorithm is used which converts these drawings into pixels for classification and the extracted values are been matched with the trained database to extract various features and results are produced with maximum accuracy. Also OpenCV (Open Source Computer Vision Library) a library of programming functions mainly aimed at real-time computer vision was built to provide an infrastructure for computer vision applications and to accelerate the use of machine perception in the real time. Thus, our output will showcase the early detection of the disease and can be able to increase the lifespan of the diseased patient with proper treatments and medications leads to peaceful life.

## TABLE OF CONTENT

<b>CHAPTER NO</b>	<b>TITLE</b>	<b>PAGE NUMBER</b>
<b>1.</b>	<b>INTRODUCTION</b> <b>1.1 OVERVIEW</b> <b>1.2 PROBLEM STATEMENT</b> <b>1.3 EXISTING SYSTEM</b> <b>1.4 PROPOSED SYSTEM</b>	<b>9</b> <b>} 10</b>
<b>2.</b>	<b>LITERATURE SURVEY</b>	<b>11-15</b>
<b>3.</b>	<b>SYSTEM DESIGN</b> <b>3.1 UNIFIED MODELING LANGUAGE</b> <b>3.2 UML FLOW DIAGRAMS</b> <b>3.2.1 Use Case Diagram of Parkinson Disease Detection</b> <b>3.2.2 Activity Diagram of Parkinson Disease Detection</b> <b>3.2.3 Collaboration Diagram of Parkinson Disease Detection</b> <b>3.2.4 Component Diagram of Parkinson Disease Detection</b> <b>3.2.5 Deployment Diagram of Parkinson Disease Detection</b> <b>3.2.6 Package Diagram of Parkinson Disease Detection</b>	<b>16</b>     <b>17-20</b>
<b>4.</b>	<b>SYSTEM ARCHITECTURE</b> <b>4.1 Architectural Design</b> <b>4.2 Architectural Description</b>	<b>23</b>  <b>24-26</b>

<b>5.</b>	<b>SYSTEM IMPLEMENTATION</b> <b>5.1 SYSTEM DESCRIPTION</b> <b>5.2 DATA SET</b> <b>5.3 PREPROCESSING</b>	<b>27-28</b> <b>28-29</b> <b>30</b>
<b>6.</b>	<b>RESULT &amp; CODING</b> <b>6.1 SAMPLE CODES</b> <b>6.2 SCREENSHOTS</b>	<b>31-36</b> <b>37</b>
<b>7.</b>	<b>CONCLUSION AND FUTURE WORKS</b> <b>7.1 CONCLUSION</b> <b>7.2 FUTURE WORK</b>	<b>38</b> <b>39</b>
<b>8.</b>	<b>REFERENCES</b>	<b>40-42</b>

## **CHAPTER 1**

### **INTRODUCTION**

Parkinson Disease is a brain neurological disorder. It leads to shaking of the body, hands and provides stiffness to the body. No proper cure or treatment is available yet at the advanced stage. Treatment is possible only when done at the early or onset of the disease. These will not only reduce the cost of the disease but will also possibly save the life. A person diagnosed with Parkinson's disease can have symptoms include:

- Depression
- Sleeping and memory related issues
- Anxiety
- Loss of sense of smell

#### **1.1 OVERVIEW**

In most countries the detection of crime is the responsibility of the police, though special law enforcement agencies may be responsible for the discovery of particular types of crime (e.g., customs departments may be charged with combating smuggling and related offenses). Crime detection falls into three distinguishable phases: the discovery that a crime has been committed, the identification of a suspect, and the collection of sufficient evidence to indict the suspect before a court. Many crimes are discovered and reported by persons other than the police (e.g., victims or witnesses). Certain crimes—in particular those that involve a subject's assent, such as dealing in illicit drugs or prostitution, or those in which there may be no identifiable victim, such as obscenity—are often not discovered unless the police take active steps to determine whether they have been committed. To detect such crimes, therefore, controversial methods are sometimes required (e.g., electronic eavesdropping, surveillance, interception of communications, and infiltration gangs)

## **1.2 PROBLEM STATEMENT**

Monitoring the patient's condition is the key to the success of the correction of the main clinical manifestations of PD, including the almost inevitable modification of the clinical picture of the disease against the background of prolonged dopaminergic therapy.

## **1.3 EXISTING SYSTEM**

In the existing system, PD is detected at the secondary stage only which leads to medical challenges. Also doctor has to manually examine and suggest medical diagnosis in which the symptoms might vary from person to person so suggesting medicine is also a challenge.

Thus the mental disorders are been poorly characterized and have many health complications. PD is generally diagnosed with the following clinical methods as

- MRI scan
- PET scan
- SPECT scan

The result in a high misdiagnosis rate and many years before diagnosis, people can have the disease. Thus the existing system is not effective in early prediction and accurate medicinal diagnosis to the affected person. .

## **1.4 PROPOSED SYSTEM**

Our proposed approach uses svm methods to detect the Parkinson disease by comparing the data set. The data are read from the dataset and then unwanted rows and columns are removed by data standardization. The data is divided into testing and training data. Confusion matrix, recall, f1 score, accuracy and classification report are determined for training and testing data. From the training data parkinson disease is predicted.

## **CHAPTER 2**

### **LITERATURE SURVEY**

Wu Wang , JUNHO LEE , FOUZI HARROU , (Member, IEEE), AND YING SUN. Parkinson's disease (PD) is becoming an important degenerative disease of the central nervous system, affecting the quality of lives of millions of seniors worldwide [1]. Symptoms of PD can progress differently from one person to another because of the variety of the disease. Till now, there is no way to diagnose Parkinson's disease (PD) [2]. However, there are various symptoms and diagnostic tests used in combination. Several biomarkers have been investigated by scientists to early identify PD to slow down the disease process. This study proposes a deep learning framework for the early detection of PD. PD detection is done into two main stages: training and testing.

Kaveti Kiran Kumar, Panthagani Vijay Babu .A disease that causes due to degeneration in the nerve cells in brain parts with which movements are controlled is a Parkinson's disease. Parkinson's disease increased its occurrence with age and it's becoming the second most common neurodegenerative disorder. Due to this disease, people begin to experience difficulty in speaking, writing, walking. We can diagnose this disease by using voice recordings dataset, handwritten images, and by using several techniques. In this paper, we are using a voice dataset to identify Parkinson's disease which is collected from different people with and without Parkinson's disease.

Merry Saxena, Sachin Ahuja. Prognosis and progression of Parkinson's disease is a critical question among the clinicians since there is a disparity of parameters taken into the diagnostic consideration thereby making the decision process difficult. Different datasets have been independently explored and applied through machine learning to analyze the incidence of occurrence and progression of the disease. The present paper is an updated report of the types of Supervised Machine Learning algorithms which have gained prominence within a span of last 5 years (2015- 2019).

Further it highlights the use of hybrid intelligence models to improve the prediction

accuracy and sensitivity over standalone methods. Conclusively the paper also emphasis on the need of development of multiparametric, big data based holistic predictive system. Predictive analytics is gaining a strength hold in healthcare sector since there is a gaining societal view that “Prevention is better than cure”.

PROTIMA KHAN, MD. FAZLUL KADER, AISHA B. RAHMAN Brain is the controlling center of our body. With the advent of time, newer and newer brain diseases are being discovered. Thus, because of the variability of brain diseases, existing diagnosis or detection systems are becoming challenging and are still an open problem for research. Detection of brain diseases at an early stage can make a huge difference in attempting to cure them. In recent years, the use of artificial intelligence (AI) is surging through all spheres of science, and no doubt, it is revolutionizing the field of neurology. Application of AI in medical science has made brain disease prediction and detection more accurate and precise. In this study, we present a review on recent machine learning and deep learning approaches in detecting four brain diseases such as Alzheimer’s disease (AD), brain tumor, epilepsy, and Parkinson’s disease.

Waseem Ahmad Mir, Tawseef Ayoub Shaikh. Parkinson's disease (PD) is disabling disease that affects the quality of life. It belimps due to the death of cells that produce dopamine’s in the substantia nigra part of the central nervous system (CNS) which affects the human body. People who have Parkinson’s disease feel difficulty in doing activities like speaking, writing, and walking. PD is a progressive neuropathological degeneration of the CNS in which the motor functions of the human body are affected Among the neurological disorders, after Alzheimer’s disease, the second most frequently found disease is PD [2], and in North America alone the number of PD patients has crossed more than one million people. In proposed the normal subjects from PD subjects. In their work, the dimension of the feature vector was reduced and optimized features were obtained by using a genetic algorithm (GA) and k nearest neighbor (k-NN) was used for classification.

At present, the vast majority of human subjects with neurological disease are still

diagnosed through in-person assessments and qualitative analysis of patient data.

In this paper, we propose to use Topological Data Analysis (TDA) together with machine learning tools to automate the process of Parkinson's disease classification and severity assessment. We proposed a methodology which incorporates TDA into analyzing Parkinson's disease postural shifts data through the representation of persistence images. We propose a method to 1) classify healthy patients from those afflicted by disease and 2) diagnose the severity of disease. We explore the use of the proposed method in an application involving a Parkinson's disease dataset comprised of healthy-elderly, healthy-young and Parkinson's disease patients. For the classification experiment, we performed two tests on the dataset. The first was a binary classification test to assess whether we can discriminate between a subject with Parkinson's disease and one who is healthy. Our second test, a 3-class classification, is more challenging as we seek to discriminate between healthyelderly, healthy-young, and Parkinson's disease subjects. For both tests, we used a linear-SVM from the SVM package in the Python library scikit-learn.

Kazi Amit Hasan; Md. Al Mehedi Hasan

Parkinson's disease (PD) is a growing and chronic neurodegenerative disease with a great amount of motor and non-motor symptoms. In the initial stages, most of the PD patients face difficulties in regular movements. Vocal disorders are one of the common symptoms of them. Vocal disorder centric diagnosis systems are one of the leading areas in recent PD detection studies. In this paper, the dataset was taken from the UCI Machine Learning repository and a feature extraction technique was applied. Multiple machine learning classification methods were applied and compared with other related existing works. Experimental results show that the highest accuracy score of 0.91 was achieved with the Random Forest Classifier method by feeding the selected features.

Robert LeMoyne; Timothy Mastroianni; Donald Whiting; Nestor Tomycz Deep brain stimulation enables highly specified patient-unique therapeutic intervention ameliorating the symptoms of Parkinson's disease. Inherent to the efficacy of deep brain



stimulation is the acquisition of an optimal parameter configuration. Using conventional methods, the optimization process for tuning the deep brain stimulation system parameters can intrinsically induce strain on clinical resources. An advanced means of quantifying Parkinson's hand tremor and distinguishing between parameter settings would be highly beneficial. The conformal wearable and wireless inertial sensor system, such as the BioStamp nPoint, has a volumetric profile on the order of a bandage that readily enables convenient quantification of Parkinson's disease hand tremor. Furthermore, the BioStamp nPoint has been certified by the FDA as a 510(k) medical device for acquisition of medical grade data. Parametric variation of the amplitude parameter for deep brain stimulation can be quantified through the BioStamp nPoint conformal wearable and wireless inertial sensor system mounted to the dorsum of the hand

- Yi-Wen Wang\* , Chien-Hsu Chen, Yang-Cheng Lin The purpose of this study is to develop an AR-based serious game for the rehabilitation of patients with PD and assess improvements in gait and sense of balance after 3 weeks of training. The game was designed to allow patients with PD to ambulate while following virtual visual cues in a real environment, and incorporated levels with increasing difficulty. To motivate the patients to step on the visual cues, the game was designed with a cumulative points system. One point was awarded for stepping on each visual cue, and patients were encouraged to step on as many cues as possible to maximize the number of points obtained within the allotted time. The score at the end of each game was recorded, and training progress was documented by comparing each patient's most recent performance with the scores obtained in previous games. This feedback also motivated patients to improve their performance in subsequent games. As we aimed to train patients to raise their legs during regular walking, the first episode of the game was designed with an obstacle-free route.

Afra Nawar; Farhan Rahman; Narayanan Krishnamurthi; Anirudh Som; Pavan Turaga Parkinson's disease is diagnosed primarily by trained medical professionals using in-person assessments and surveys that gauge Parkinson's disease patients' motor

abilities. A common measure of the disease severity is the Unified Parkinson's Disease Rating Scale (UPDRS) score used by clinicians to track symptom progression .

Because the disease is mainly analyzed by medical personnel, diagnoses are prone to subjectivity. To mitigate this, many have tried to develop machine learning (ML) methods to assess the severity of Parkinson's disease. Some of these automated methods utilize data from different sensor devices. Sensors provide biometric data about a Parkinson's Disease patient which can then be processed to determine discrepancies between patients and healthy subjects. Sensors also offer Parkinson's disease patients a more convenient alternative to conventional assessment methods.

Sachin Shetty, Y. S. Rao In this paper we try to differentiate PD patients from a data set which consists of gait features of PD, healthy controls, Amyotrophic lateral sclerosis(ALS) and Huntingtons disease (HD). HD is another neurodegenerative disease which causes motor and mental decline. One of the most significant symptoms is abnormal writhing movement called chorea which results in abnormal gait. Hausdorff et al. [6] used fluctuation analysis to measure correlation between adjacent stride intervals. The paper concluded that the stride interval fluctuations were more random in the Huntingtons patients than in healthy controls. ALS is a disorder that causes muscle weakness and muscle atrophy due to death of motor neurons. Stephen Hawking is a well-known scientist suffering from ALS. Lack of muscle co-ordination affects gait before the disease completely ceases muscle activity. Hausdorff et al. It also showed that for ALS, average stride time as well as walking speed is slow and measure of magnitude of stride to stride variability are increased, The paper structure is as follows: Section II deals with Related work in this domain. Section III deals with Data selection and processing. Section IV deals with Feature vector selection. Section V is about Machine learning using SVM algorithm. Section VI talks about the results and finally Section VII ends with the conclusions.

## **CHAPTER 3**

### **SYSTEM DESIGN**

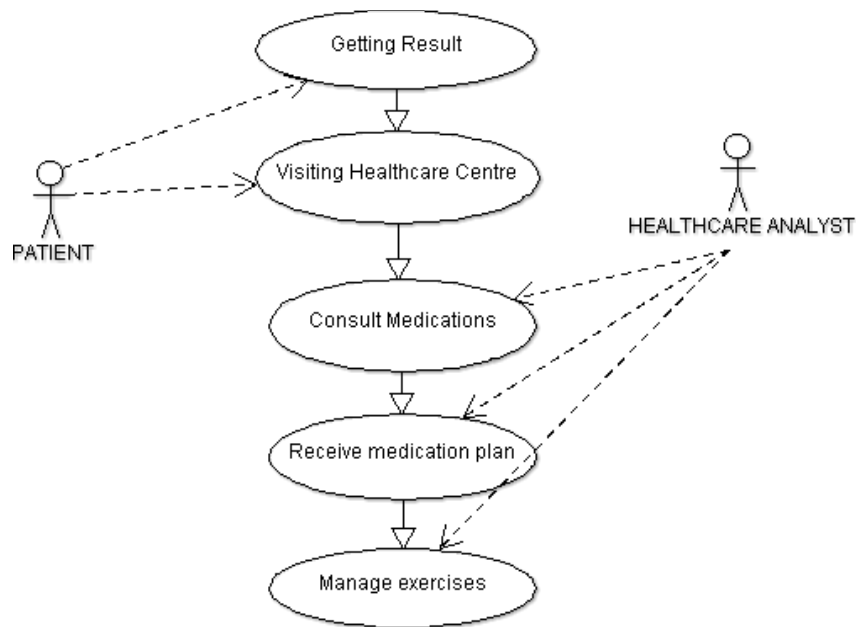
In this chapter, the various UML diagrams for Parkinson Disease detection is represented and the various functionalities are explained.

#### **3.1 UNIFIED MODELLING LANGUAGE**

Unified Modeling language (UML) is a standardized modeling language enabling developers to specify, visualize, construct and document artifacts of a software system. Thus, UML makes these artifacts scalable, secure and robust in execution. It uses graphic notation to create visual models of software systems. UML is designed to enable users to develop an expressive, ready to use visual modeling language. In addition, it supports high-level development concepts such as frameworks, patterns and collaborations. Some of the UML diagrams are discussed.

##### **3.1.1 USE CASE DIAGRAM OF PARKINSON DISEASE DETECTION**

Use case diagrams are considered for high level requirement analysis of a system. So when the requirements of a system are analysed the functionalities are captured in use cases. So it can be said that uses cases are nothing but the system functionalities written in an organized manner. Now the second things which are relevant to the use cases are the actors. Actors can be defined as something that interacts with the system. The actors can be human user, some internal applications or may be some external applications. Use case diagrams are used to gather the requirements of a system including internal and external influences. These requirements are mostly design requirements. Hence, when a system is analyzed to gather its functionalities, use cases are prepared and actors are identified.



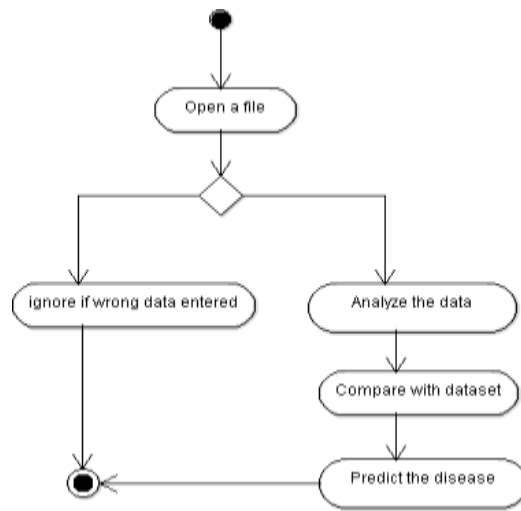
**Figure 3.1 Use case diagram of Parkinson**

### **Disease Detection**

The Functionalities are to be represented as a use case in the representation. Each and every use case is a function in which the user or the server can have the access on it. The names of the use cases are given in such a way that the functionalities are preformed, because the main purpose of the functionalities is to identify the requirements. To add some extra notes that should be clarified to the user, the notes kind of structure is added to the use case diagram. Only the main relationships between the actors and the functionalities are shown because all the representation may collapse the diagram. The use case diagram as shown in Figure 3.1 provides details based on the Parkinson Disease Detection.

### 3.1.2 ACTIVITY DIAGRAM OF PARKINSON DISEASE DETECTION

Activity is a particular operation of the system. Activity diagram is suitable for modeling the activity flow of the system.



**Figure 3.2 Activity diagram of Parkinson Disease Detection**

Activity diagrams are not only used for visualizing dynamic nature of a system but they are also used to construct the executable system by using forward and reverse engineering techniques. The only missing thing in activity diagram is the message part.

An application can have multiple systems. Activity diagram also captures these systems and describes the flow from one system to another. This specific usage is not available in other diagrams. These systems can be database, external queues, or any other system. Activity diagram is suitable for modeling the activity flow of the system

It does not show any message flow from one activity to another. Activity diagram is sometime considered as the flow chart. Although the diagrams looks like a flow chart but it is not. It shows different flow like parallel, branched, concurrent and single. The Figure 3.2 shows the activity diagram of the developed application.

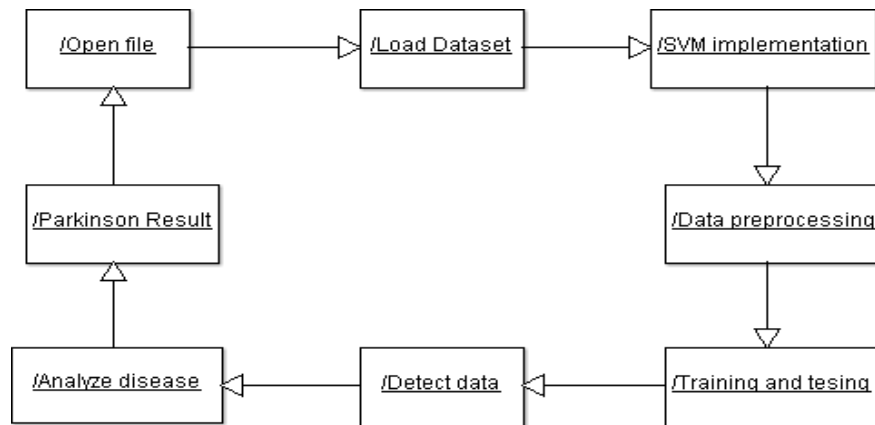
### **3.1.3 COLLABORATION DIAGRAM OF PARKINSON DISEASE DETECTION**

The next interaction diagram is collaboration diagram. It shows the object organization. Here in collaboration diagram the method call sequence is indicated by some numbering technique. The number indicates how the methods are called one after another. The method calls are similar to that of a sequence diagram. But the difference is that the sequence diagram does not describe the object organization whereas the collaboration diagram shows the object organization. The various objects involved and their collaboration is shown in Figure 3.3

Now to choose between these two diagrams the main emphasis is given on the type of requirement. If the time sequence is important then sequence diagram is used and if organization is required then collaboration diagram is used.

There are three primary elements of a collaboration diagram:

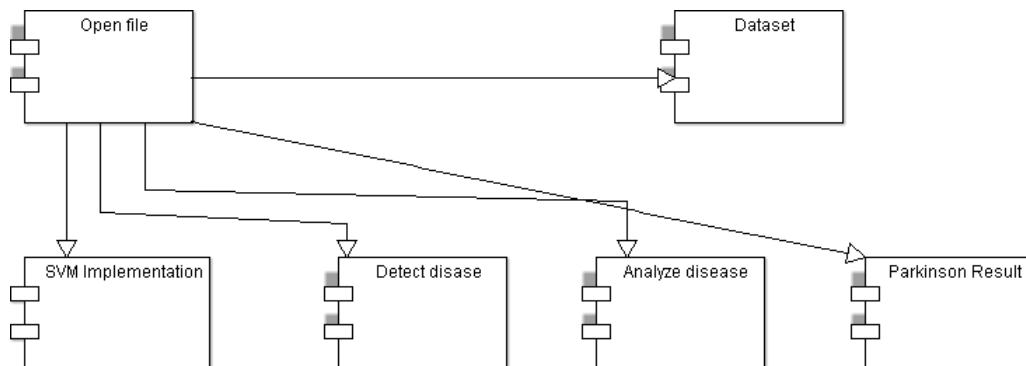
- Objects.
- Links.
- Messages.



**Figure 3.3 Collaboration diagram of Parkinson Disease Detection**

### 3.1.4 COMPONENT DIAGRAM OF PARKINSON DISEASE DETECTION

A component diagram displays the structural relationship of components of a software system. These are mostly used when working with complex systems that have many components such as sensor nodes, cluster head and base station. It does not describe the functionality of the system but it describes the components used to make those functionalities. Components communicate with each other using interfaces. The interfaces are linked using connectors. The Figure 3.4 shows a component,diagram.



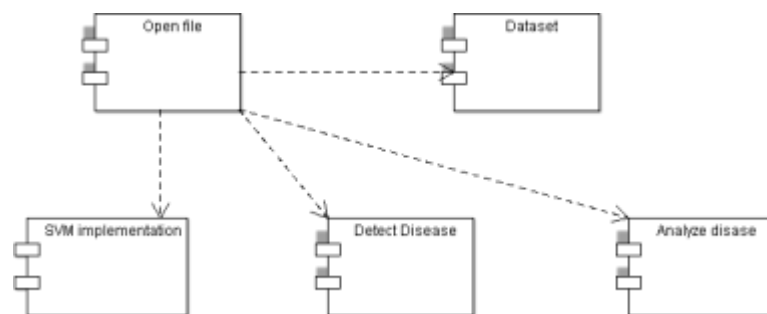
**Figure 3.4 Component diagram of Emotion Detection**

### 3.1.5 DEPLOYMENT DIAGRAM OF PARKINSON DISEASE DETECTION

A deployment diagrams shows the hardware of your system and the software in those hardware. Deployment diagrams are useful when your software solution is deployed across multiple machines such as sensor nodes, cluster head and base station with each having a unique configuration. The Figure 3.5 represents deployment diagram for the developed application.

A Deployment diagram is a diagram that shows the configuration of run time processing nodes and the components that live on them. Deployment diagrams is a kind of structure diagram used in modeling the physical aspects of an object-oriented system. They are often be used to model the static deployment view of a system (topology of the hardware).

Deployment Diagram in the figure 3.5 shows how the modules gets deployed in the system.



**Figure 3.5 Deployment diagram of Parkinson Disease Detection**

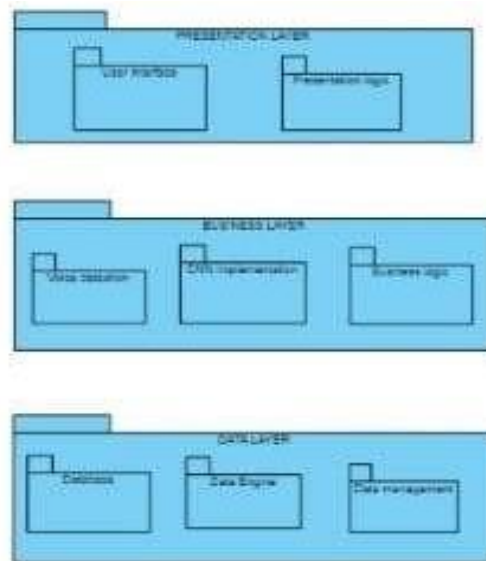


### 3.1.6 PACKAGE DIAGRAM OF PARKINSON DISEASE DETECTION

Package diagrams are used to reflect the organization of packages and their elements. When used to represent class elements, package diagrams provide a visualization of the namespaces. Package diagrams are used to structure high level system elements.

Package diagrams can be used to simplify complex class diagrams, it can group classes into packages. A package is a collection of logically related UML elements.

Packages are depicted as file folders and can be used on any of the UML diagrams. The Figure 3.6 represents package diagram for the developed application which represents how the elements are logically related.



**Figure 3.6 Package diagram of Parkinson Disease Detection**

Package is a namespace used to group together elements that are semantically related and might change together. It is a general- purpose mechanism to organize elements into groups to provide better structure for system model.

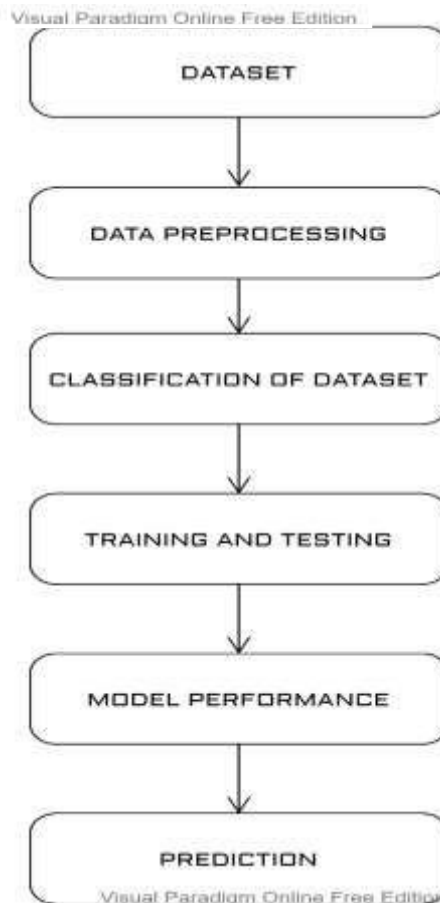
## CHAPTER 4

### SYSTEM ARCHITECTURE

In this chapter, the System Architecture for Parkinson Disease Detection is represented and the modules are explained.

#### 4.1 ARCHITECTURE DESIGN

In system architecture the detailed description about the system modules and the working of each module is discussed as shown in figure



**Figure 4.1 System Architecture of Parkinson Disease Detection**

## **4.2 ARCHITECTURE DESCRIPTION**

### **DATASET**

This dataset is composed of a range of biomedical voice measurements from 31 people, 23 with Parkinson's disease (PD). Each column in the table is a particular voice measure, and each row corresponds one of 195 voice recording from these individuals ("name" column). The main aim of the data is to discriminate healthy people from those with PD, according to "status" column which is set to 0 for healthy and 1 for PD.

The data is in ASCII CSV format. The rows of the CSV file contain an instance corresponding to one voice recording. There are around six recordings per patient, the name of the patient is identified in the first column.

### **DATA PREPROCESSING**

Data preprocessing is a process of preparing the raw data and making it suitable for a machine learning model. It is the first and crucial step while creating a machine learning model. When creating a machine learning project, it is not always a case that we come across the clean and formatted data. And while doing any operation with data, it is mandatory to clean it and put in a formatted way. So for this, we use data preprocessing task.

A real-world data generally contains noises, missing values, and maybe in an unusable format which cannot be directly used for machine learning models. Data preprocessing is required tasks for cleaning the data and making it suitable for a machine learning model which also increases the accuracy and efficiency of a machine learning model.

## CLASSIFICATION OF DATASET

In machine learning data preprocessing, we divide our dataset into a training set and test set. This is one of the crucial steps of data preprocessing as by doing this, we can enhance the performance of our machine learning model. Suppose, if we have given training to our machine learning model by a dataset and we test it by a completely different dataset. Then, it will create difficulties for our model to understand the correlations between the models. If we train our model very well and its training accuracy is also very high, but we provide a new dataset to it, then it will decrease the performance. So we always try to make a machine learning model which performs well with the training set and also with the test dataset. Here, we can define these datasets as:

**Training Data:** A subset of dataset to train the machine learning model, and we already know the output.

**Test Data:** A subset of dataset to test the machine learning model, and by using the test set, model predicts the output.

## TRAINING AND TESTING

This type of data builds up the machine learning algorithm. The data scientist feeds the algorithm input data, which corresponds to an expected output. The model evaluates the data repeatedly to learn more about the data's behavior and then adjusts itself to serve its intended purpose.

After the model is built, testing data once again validates that it can make accurate predictions. If training and validation data include labels to monitor performance metrics of the model, the testing data should be unlabeled. Test data provides a final, real-world check of an unseen dataset to confirm that the ML algorithm was trained effectively.

## **MODEL PERFORMANCE**

We have used the Support Vector Machine (SVM) as the machine learning algorithm. The SVM is a state-of-the-art classification method that uses a learning algorithm based on structural risk minimization. The classifier can be used in many disciplines because of its high accuracy, ability to deal with high dimensions, and flexibility in modeling diverse sources of data.

In this module, we evaluate the performance of trained machine learning model using performance evaluation criteria such as F1 score, accuracy, recall and classification error. In case the model performs poorly, we optimize the machine learning algorithms to improve the performance. Performance Evaluation is defined as a formal and productive procedure to measure an employee's work and results based on their job responsibilities. It is used to gauge the amount of value added by an employee in terms of increased business revenue, in comparison to industry standards and overall employee return on investment (ROI).

## **PREDICTION**

Prediction" refers to the output of an algorithm after it has been trained on a historical dataset and applied to new data when forecasting the likelihood of a particular outcome. In this module we use trained and optimized machine learning model to predict whether the patient has parkinsons or not.

## CHAPTER 5

### SYSTEM IMPLEMENTATION

In this chapter, the System Implementation for the Parkinson disease detection System.

#### 5.1 SYSTEM DESCRIPTION

Machine learning (ML) is an application that provides a system with the ability to learn and improve automatically from past experiences without being explicitly programmed. After viewing the data, an exact pattern or information cannot always be determined. In such cases, ML is applied to interpret the exact pattern and information. ML pushes forward the idea that, by providing a machine with access to the right data, the machine can learn and solve both complex mathematical problems and some specific problems

The main focus of this study was to investigate machine-learning-based techniques with the best accuracy in predicting Parkinson disease and explore its applicability with particular importance to the dataset. Supervised ML techniques were used to analyze the dataset to carry out data validation, data cleaning, and data visualization on the given dataset. The results of the different supervised ML algorithms were compared to predict the results. The proposed system consists of data collection, data preprocessing, construction of a predictive model, dataset training, dataset testing. The aim of this study is to prove the effectiveness and accuracy of a ML algorithm for predicting violent crimes.

Random forest is a **Supervised Machine Learning Algorithm** that is **used widely in Classification and Regression problems**. It builds decision trees on different samples and takes their majority vote for classification and average in case of regression.

One of the most important features of the Support Vector Machine Algorithm is that it can handle the data set containing **continuous variables** as in the case of regression and **categorical variables** as in the case of classification. It performs better results for classification problems.

## 5.2 PREPROCESSING

Data pre-processing is one of the most important steps in machine learning. It is the most important step that helps in building machine learning models more accurately. In machine learning, there is an 80/20 rule. Every data scientist should spend 80% time for data pre-processing and 20% time to actually perform the analysis.

### What is data pre-processing?

Data pre-processing is a process of cleaning the raw data i.e. the data is collected in the real world and is converted to a clean data set. In other words, whenever the data is gathered from different sources it is collected in a raw format and this data isn't feasible for the analysis. Therefore, certain steps are executed to convert the data into a small clean data set, this part of the process is called as data pre-processing.

### Why do we need it?

As we know that data pre-processing is a process of cleaning the raw data into clean data, so that can be used to train the model. So, we definitely need data pre-processing to achieve good results from the applied model in machine learning and deep learning projects.

Most of the real-world data is messy, some of these types of data are:

1. **Missing data:** Missing data can be found when it is not continuously created or due to technical issues in the application (IOT system).
2. **Noisy data:** This type of data is also called outliers, this can occur due to human errors (human manually gathering the data) or some technical problem of the device at the time of collection of data.

3. **Inconsistent data:** This type of data might be collected due to human errors (mistakes with the name or values) or duplication of data.

### Three Types of Data

1. Numeric e.g. income, age
2. Categorical e.g. gender, nationality
3. Ordinal e.g. low/medium/high

How can data pre-processing be performed?

These are some of the basic pre — processing techniques that can be used to convert raw data.

1. **Conversion of data:** As we know that Machine Learning models can only handle numeric features, hence categorical and ordinal data must be somehow converted into numeric features.
2. **Ignoring the missing values:** Whenever we encounter missing data in the data set then we can remove the row or column of data depending on our need. This method is known to be efficient but it shouldn't be performed if there are a lot of missing values in the dataset.
3. **Filling the missing values:** Whenever we encounter missing data in the data set then we can fill the missing data manually, most commonly the mean, median or highest frequency value is used.
4. **Machine learning:** If we have some missing data then we can predict what data shall be present at the empty position by using the existing data.
5. **Outliers detection:** There are some error data that might be present in our data set that deviates drastically from other observations in a data set. [Example: human weight = 800 Kg; due to mistyping of extra 0]



## 5.3 TESTING

Once the data is divided into the 3 given segments we can start the training process.

In a data set, a training set is implemented to build up a model, while a test (or validation) set is to validate the model built. Data points in the training set are excluded from the test (validation) set. Usually, a data set is divided into a training set, a validation set (some people use 'test set' instead) in each iteration, or divided into a training set, a validation set and a test set in each iteration.

- The model uses any one of the models that we had chosen in step 3/ point 3. Once the model is trained we can use the same trained model to predict using the testing data i.e. the unseen data. Once this is done we can develop a confusion matrix, this tells us how well our model is trained. A confusion matrix has 4 parameters, which are '**True positives**', '**True Negatives**', '**False Positives**' and '**False Negative**'. We prefer that we get more values in the True negatives and true positives to get a more accurate model. The size of the Confusion matrix completely depends upon the number of classes.
- **True positives** : These are cases in which we predicted TRUE and our predicted output is correct.
- **True negatives** : We predicted FALSE and our predicted output is correct.
- **False positives** : We predicted TRUE, but the actual predicted output is FALSE.
- **False negatives** : We predicted FALSE, but the actual predicted output is TRUE.

We can also find out the accuracy of the model using the confusion matrix.

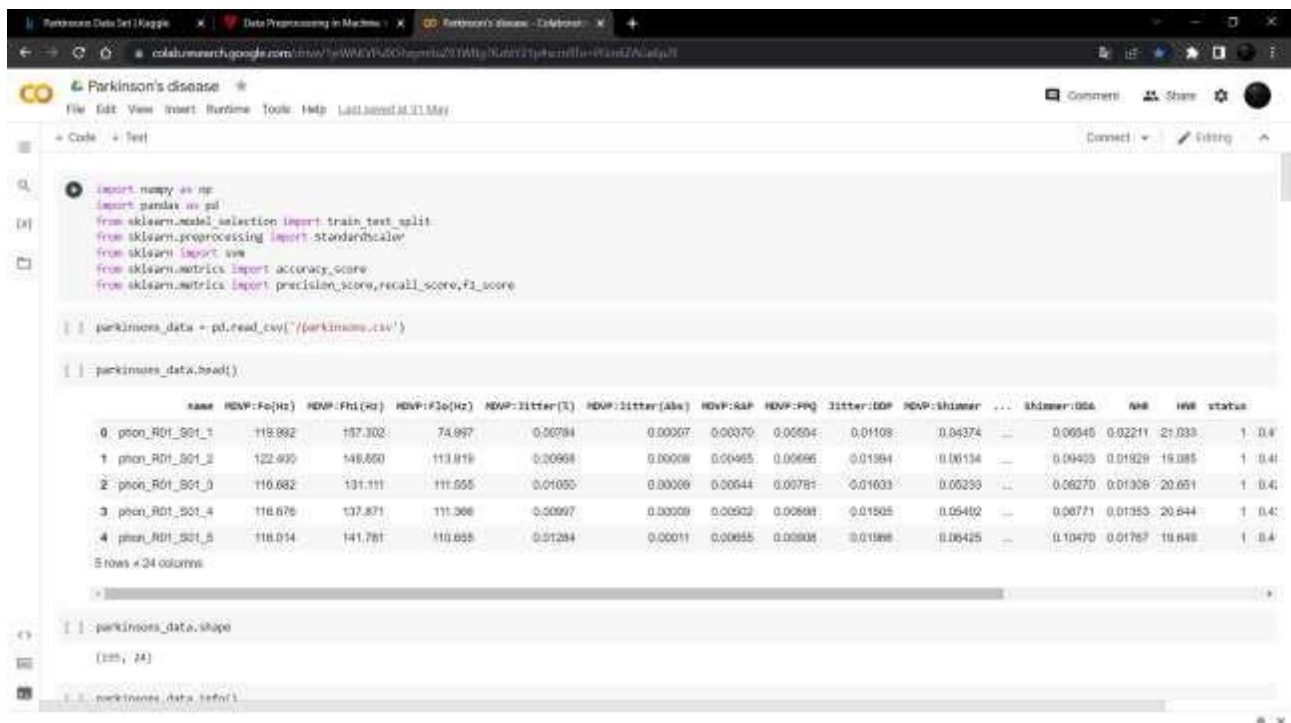
$$\text{Accuracy} = (\text{True Positives} + \text{True Negatives}) / (\text{Total number of classes})$$

## CHAPTER 6

### CODING & SCREENSHOTS

The following code has been implemented in python using GOOGLE COLAB

#### 6.1 SAMPLE CODING FOR PARKINSON DISEASE DETECTION



```
import numpy as np
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn import svm
from sklearn.metrics import accuracy_score
from sklearn.metrics import precision_score, recall_score, f1_score

parkinsons_data = pd.read_csv('parkinsons.csv')

parkinsons_data.head()
```

	name	MDVP:F0(Hz)	MDVP:F1(Hz)	MDVP:F2(Hz)	MDVP:Dttrr(L)	MDVP:Dttrr(Abs)	MDVP:RAP	MDVP:RPQ	Dttrr:DDP	MDVP:Shimmer	...	Shimmer:DDA	AGE	HW	status	
0	phon_R01_S01_1	113.892	137.302	74.997	0.06784	0.00007	0.00070	0.00504	0.01108	0.04374	...	0.06940	0.02211	21.033	1	0.4
1	phon_R01_S01_2	122.490	148.660	113.819	0.00998	0.00008	0.00465	0.00696	0.01994	0.00134	...	0.09403	0.01929	19.085	1	0.4
2	phon_R01_S01_3	110.682	131.111	111.555	0.01050	0.00009	0.00544	0.00781	0.01633	0.00293	...	0.06270	0.01306	20.651	1	0.4
3	phon_R01_S01_4	118.676	137.871	111.368	0.00997	0.00009	0.00932	0.00608	0.01505	0.05402	...	0.06771	0.01355	20.644	1	0.4
4	phon_R01_S01_5	118.014	141.781	110.655	0.01284	0.00011	0.00655	0.00808	0.01986	0.06425	...	0.10470	0.01767	19.848	1	0.4

```
parkinsons_data.shape
(195, 24)

parkinsons_data.info()
```

Figure 6.1.1 Importing Dataset

```

parkinsons_data.shape

(195, 24)

parkinsons_data.info()
Out[1]:
<class 'pandas.core.frame.DataFrame'>
Int64Index: 195 entries, 0 to 194
Data columns (total 24 columns):
 #   Column              Non-Null Count  Dtype  
---  -
 0   name                195 non-null    object  
 1   PDVP: Fo(Hz)        195 non-null    float64  
 2   PDVP: Fb1(Hz)       195 non-null    float64  
 3   PDVP: Fb2(Hz)       195 non-null    float64  
 4   PDVP: Jitter(X)     195 non-null    float64  
 5   PDVP: Jitter(Abs)   195 non-null    float64  
 6   PDVP: RAP           195 non-null    float64  
 7   PDVP: PPQ           195 non-null    float64  
 8   Jitter: DDP         195 non-null    float64  
 9   PDVP: Shimmer       195 non-null    float64  
10  PDVP: Shimmer(dB)   195 non-null    float64  
11  Shimmer: APQ2       195 non-null    float64  
12  Shimmer: APQ5       195 non-null    float64  
13  PDVP: APQ           195 non-null    float64  
14  Shimmer: DDA        195 non-null    float64  
15  RWR                 195 non-null    float64  
16  RWR                 195 non-null    float64  
17  status              195 non-null    int64  
18  RPSE                195 non-null    float64  
19  DFA                 195 non-null    float64  
20  spread1             195 non-null    float64  
21  spread2             195 non-null    float64  
22  D2                  195 non-null    float64  
23  PPE                 195 non-null    float64  
dtypes: float64(22), int64(1), object(1)

```

Figure 6.1.2 Data Collection

```

parkinsons_data.isnull().sum()
Out[1]:
name                0
PDVP: Fo(Hz)        0
PDVP: Fb1(Hz)       0
PDVP: Fb2(Hz)       0
PDVP: Jitter(X)     0
PDVP: Jitter(Abs)   0
PDVP: RAP           0
PDVP: PPQ           0
Jitter: DDP         0
PDVP: Shimmer       0
PDVP: Shimmer(dB)   0
Shimmer: APQ2       0
Shimmer: APQ5       0
PDVP: APQ           0
Shimmer: DDA        0
RWR                 0
RWR                 0
status              0
RPSE                0
DFA                 0
spread1             0
spread2             0
D2                  0
PPE                 0
dtypes: int64

parkinsons_data.describe()
Out[1]:
      PDVP: Fo(Hz)  PDVP: Fb1(Hz)  PDVP: Fb2(Hz)  PDVP: Jitter(X)  PDVP: Jitter(Abs)  PDVP: RAP  PDVP: PPQ  Jitter: DDP  PDVP: Shimmer  PDVP: Shimmer(dB)  Shimmer: APQ2  Shimmer: APQ5  RWR  RPSE  DFA  spread1  spread2  D2  PPE
count  195.000000  195.000000    195.000000    195.000000    195.000000    195.000000  195.000000  195.000000  195.000000  195.000000  195.000000  195.000000  195.000000  195.000000  195.000000  195.000000  195.000000  195.000000  195.000000

```

Figure 6.1.3 Analyzing data

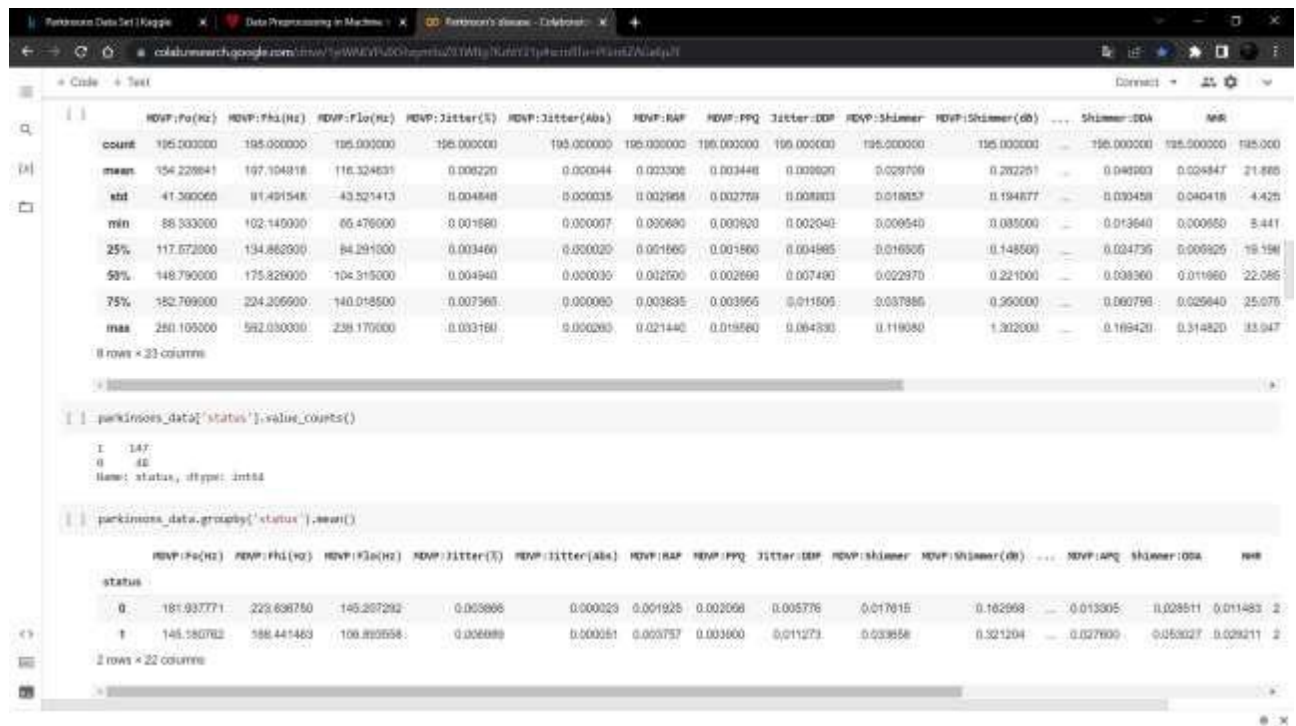


Figure 6.1.4 Describing Data

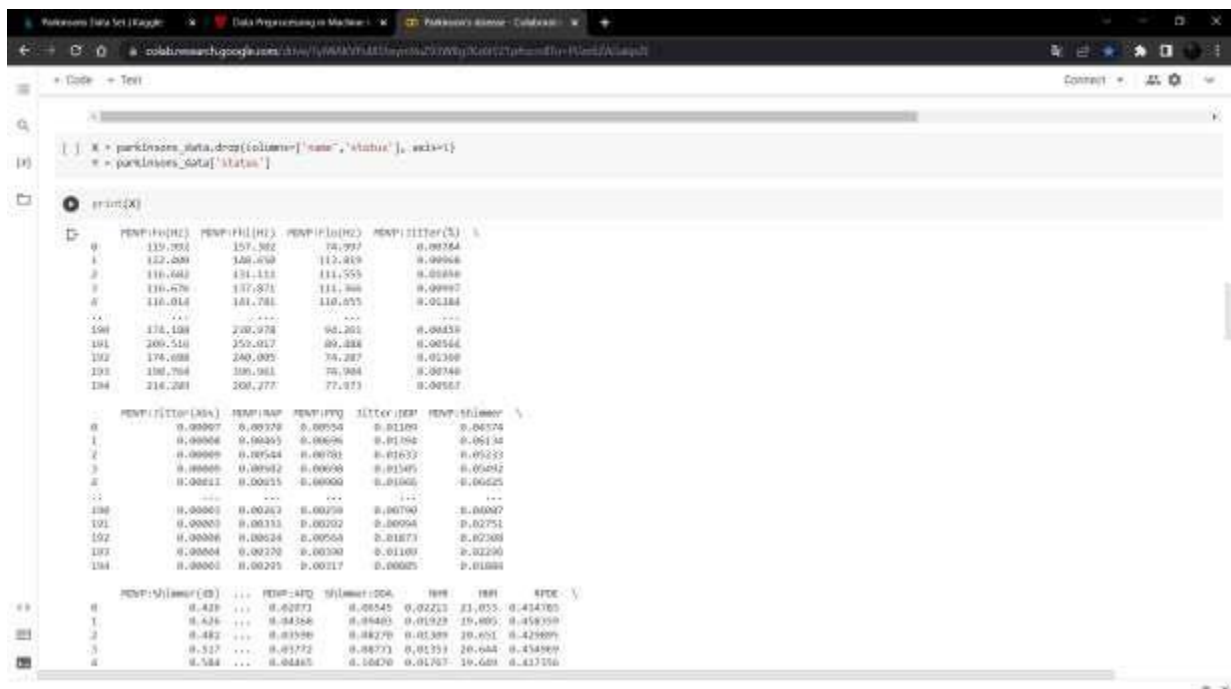


Figure 6.1.5 Data Preprocessing

```

Rankinson Data Set | Kaggle | Data Processing in Machine | Rankinson's disease - ColabNotebook
colab.research.google.com/drive/7pWwVp50Zgmru2YTAh3Kutv2tp4mndfUu-PlmZNAuq0t

Code | Text
[ ] 2 0.482 ... 0.07500 0.08270 0.01300 38.653 0.429895
[ ] 3 0.537 ... 0.07772 0.08771 0.01353 20.644 0.438669
[ ] 4 0.584 ... 0.04465 0.10430 0.01767 19.649 0.417356
[ ] ...
[ ] 190 0.265 ... 0.02345 0.07000 0.02764 19.527 0.440439
[ ] 191 0.263 ... 0.01879 0.04812 0.01810 19.147 0.431674
[ ] 192 0.250 ... 0.01067 0.03804 0.10735 17.883 0.407569
[ ] 193 0.281 ... 0.01180 0.03794 0.07211 19.020 0.351221
[ ] 194 0.190 ... 0.01173 0.03050 0.03310 11.200 0.662801

DFA spread1 spread2 B3 PPE
0 0.819285 -4.811011 0.366402 2.301042 0.200014
1 0.819521 -4.075192 0.325500 2.486055 0.360074
2 0.825288 -4.643179 0.311173 2.342299 0.132614
3 0.819235 -4.117501 0.334147 2.405534 0.360975
4 0.825484 -3.907707 0.234513 2.332180 0.410335
...
190 0.857890 -4.598180 0.121052 2.857476 0.133050
191 0.885344 -4.193125 0.128361 2.784312 0.168316
192 0.850883 -4.707107 0.150453 2.678772 0.131736
193 0.843856 -4.704577 0.203454 2.136886 0.123306
194 0.668357 -5.724056 0.190067 2.555477 0.148588

[195 rows x 22 columns]

[ ] print(Y)
0 1
1 1
2 1
3 1
4 1
...
190 0
191 0
192 0
193 0
194 0
Name: status, length: 195, dtype: int8

```

```

Rankinson Data Set | Kaggle | Data Processing in Machine | Rankinson's disease - ColabNotebook
colab.research.google.com/drive/7pWwVp50Zgmru2YTAh3Kutv2tp4mndfUu-PlmZNAuq0t

Code | Text
[ ] X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.2, random_state=3)
[ ] print(X.shape, X_train.shape, X_test.shape)
(195, 22) (156, 22) (39, 22)

[ ] scaler = StandardScaler()
[ ] scaler.fit(X_train)
StandardScaler()

[ ] X_train = scaler.transform(X_train)
X_test = scaler.transform(X_test)

[ ] print(X_train)
[[ 0.62239611 -0.02733083 -0.87945849 ... -0.07586547 -0.50160318
  0.07760404]
 [-1.05512729 -0.83337043 -0.0264770 ... 0.3081098 -0.81824073
  0.20201782]
 [ 0.02896167 -0.29531868 -1.12211397 ... -0.43937044 -0.62849605
  -0.50846486]
 ...
 [-0.4906735 -0.6637382 -0.340038 ... 1.22001077 -0.47806029
  -0.23504022]
 [-0.35577689 0.19731822 -0.70003679 ... -0.17056019 -0.47271835
  0.28381321]
 [ 1.03067008 0.19922313 -0.81944072 ... -0.710232 1.23832006
  -0.04629346]]

SUPPORT VECTOR MACHINE

```

Figure 6.1.6 Splitting data (training and testing data)

The screenshot shows a Jupyter Notebook interface with a browser window at the top displaying the URL `colab.research.google.com`. The notebook has three tabs: "Parkinson's Data Set | Kaggle", "Data Preprocessing in Machine", and "Parkinson's disease - ColabNotebook". The active tab is "Parkinson's disease - ColabNotebook". The notebook contains the following code and output:

```
SUPPORT VECTOR MACHINE

[ ] model = svm.SVC(kernel='linear')

[ ] model.fit(X_train, y_train)

SVM(kernel='linear')

ACCURACY

[ ] X_train_prediction = model.predict(X_train)
training_data_accuracy = accuracy_score(y_train, X_train_prediction)

[ ] print('Accuracy score of training data : ', training_data_accuracy)

Accuracy score of training data : 0.882035846253846

[ ] X_test_prediction = model.predict(X_test)
test_data_accuracy = accuracy_score(y_test, X_test_prediction)

[ ] print('Accuracy score of test data : ', test_data_accuracy)

Accuracy score of test data : 0.837946717048718

CONFUSION MATRIX

[ ] from sklearn.metrics import confusion_matrix
confusion_matrix(y_train, X_train_prediction)

array([[ 77, 13],
       [ 3, 111]])
```

Figure 6.1.7 Accuracy

The screenshot shows a Jupyter Notebook interface with a browser window at the top displaying the URL `colab.research.google.com`. The notebook has three tabs: "Parkinson's Data Set | Kaggle", "Data Preprocessing in Machine", and "Parkinson's disease - ColabNotebook". The active tab is "Parkinson's disease - ColabNotebook". The notebook contains the following code and output:

```
array([[ 77, 13],
       [ 3, 111]])

[ ] confusion_matrix(y_test, X_test_prediction)

array([[ 3, 3],
       [ 2, 29]])

CLASSIFICATION REPORT

[ ] from sklearn.metrics import classification_report
print(classification_report(y_train, X_train_prediction))

      precision    recall  f1-score   support

     0       0.84      0.68      0.75        88
     1       0.59      0.90      0.73       118

 accuracy: 0.87
 macro avg: 0.82
 weighted avg: 0.88

[ ] print(classification_report(y_test, X_test_prediction))

      precision    recall  f1-score   support

     0       0.73      0.67      0.70         3
     1       0.93      0.94      0.93        31

 accuracy: 0.81
 macro avg: 0.79
 weighted avg: 0.87
```

Figure 6.1.8 Classification Report

The screenshot shows a Google Colab notebook with the following code and output:

```
PRECISION SCORE

[ ] precision_score(y_train, x_train_prediction)
0.9951612903225000

[ ] precision_score(y_test, x_test_prediction)
0.99025

RECALL SCORE

[ ] recall_score(y_train, x_train_prediction)
0.99990031741371

[ ] recall_score(y_test, x_test_prediction)
0.975483670677419

F1 SCORE

[ ] f1_score(y_train, x_train_prediction)
0.925

[ ] f1_score(y_test, x_test_prediction)
0.9206349226330506
```

Below the code, there is a section labeled "PREDICTION" which is currently empty.

**Figure 6.1.9 Precision score,Recall F1 score**



[illegible]

## Sample Output



## **CHAPTER 7**

### **CONCLUSION & FUTURE WORK**

#### **7.1 CONCLUSION**

Disease diagnosis and prediction is possible through automated machine learning architectures using only non-invasive voice biomarkers as features. Our analysis provides a comparison of the effectiveness of various machine learning classifiers in disease diagnosis with noisy and high dimensional data. After thorough feature selection, clinical level accuracy is possible. These results are promising because they may introduce novel means to assess patient health and neurological diseases using voice data. Disease diagnosis and prediction is possible through automated machine learning architectures using only non-invasive voice biomarkers as features. Our analysis provides a comparison of the effectiveness of various machine learning classifiers in disease diagnosis with noisy and high dimensional data. Our analysis provides a comparison of the effectiveness of various machine learning classifiers in disease diagnosis with noisy and high dimensional data. After thorough feature selection, clinical level accuracy is possible. These results are promising because they may introduce novel means to assess patient health and neurological diseases using voice data. Currently, the neurodegenerative disease research area is of much significance and its diagnosis at its earlier stage can make the patient's life better. The support vector machine model can be efficiently used to monitor and diagnose the Parkinson disease at the early stage.

## **7.2 FUTURE WORK**

In future work, we can focus on different techniques to predict the Parkinson disease using different datasets. In this research, we using binary attribute of the (1- diseased patients, 0-non-diseased patients) for patient's classification. In the future we will use different types of attributes for the classification of patients and also identify the different stages of Parkinson's disease.

## REFERENCES

- [1] GBD 2013 Mortality and Causes of Death, Collaborators (17 December 2014). "Global, regional, and national age-sex specific all-cause and cause specific mortality for 240 causes of death, 1990-2013: a systematic analysis for the Global Burden of Disease Study 2013."
- [2] MS Bryant, PT, DH Rintala, JG Hou, EC Lai, EJ Protas, "Effects of levodopa on forward and backward gait patterns in persons with parkinson's disease", *Neuro Rehabilitation*, pp.247-252, 2011.
- [3] Moreno Izco F, Poza Aldea JJ, Mart Mass JF, Lpez de Munin, "Gait analysis in Parkinson's disease and response to dopaminergic treatment", *Medicina Clinica*, Jan 2005.
- [4] Roberta de Melo Roiz, Enio Walker Azevedo Cacho, Manoela Macedo Pazinato, Julia Guimares Reis, Alberto Cliquet Jr, Elizabeth M.A. Barasnevicius-Quagliato. "Gait analysis comparing Parkinsons disease with healthy elderly subjects", *Arq Neuropsiquiatr*, pp.81-86, 2010.
- [5] Silvi Frenkel-Toledo, Nir Giladi, Chava Peretz, Talia Herman Leor Gruendlinger<sup>1</sup> and Jeffrey M Hausdorff. "Effect of gait speed on gait rhythmicity in Parkinson's disease: variability of stride time and swing time respond differently", *Journal of NeuroEngineering and Rehabilitation*, 2005.
- [6] Hausdorff JM, Mitchell SL, Firtion R, Peng CK, Cudkowicz ME, Wei JY, Goldberger AL. "Altered fractal dynamics of gait: reduced stride-interval correlations with aging and Huntington's disease", *J Applied Physiology*, pp.262-269, 1997.

- [7] Hausdorff JM, Lertratanakul A, Cudkowicz ME, Peterson AL, Kaliton D, Goldberger AL. “Dynamic markers of altered gait rhythm in amyotrophic lateral sclerosis”, *J Applied Physiology*, pp.2045-2053, 2000. [8] Rustempasic, Indira, Can, M. . “Diagnosis of Parkinsons Disease using Fuzzy C-Means Clustering and Pattern Recognition”, *South East Europe Journal of Soft Computing*, 2013.
- [8] Armaanzas, Ruben, Bielza, C., Chaudhuri, K. R., Martinez-Martin, P., Larraaga, P. “Unveiling relevant non-motor Parkinson’s disease severity symptoms using a machine learning approach”, *Artificial intelligence in medicine*, pp.195-202, 2013.
- [9] Tsanas, A., Little, M. A., McSharry, P. E., Spielman, J., Ramig, L. O. , “Novel speech signal processing algorithms for highaccuracy classification of Parkinson’s disease”, *Biomedical Engineering, IEEE Transactions on*, pp.1264-1271, 2012.
- [10] Cho, C. W., Chao, W. H., Lin, S. H., Chen, Y. Y. “A vision-based analysis system for gait recognition in patients with Parkinsons disease”, *Expert Systems with applications*, pp.7033- 7039, 2009.
- [11] Yadav, G., Kumar, Y., Sahoo, et-al “Predication of Parkinson’s disease using data mining methods: A comparative analysis of tree, statistical, and support vector machine classifiers”, *Indian journal of medical sciences*, pp.231, 2011.
- [12] Huiru Zheng , Mingjing Yang, and Sally McClean. “Machine Learning and Statistical Approaches to Support the Discrimination of Neurodegenerative Diseases Based on Gait Analysis”, *Intelligent Patient Management Volume 189 of the series Studies in Computational Intelligence*

[13] Saara Rissanen, Markku Kankaanp, Mika P. Tarvainen<sup>2</sup>, Juho Nuutinen, Ina M. Tarkka, Olavi Airaksinen and Pasi Karjalainen “Analysis of surface EMG signal morphology in Parkinsons disease”, US National Institute of health, 2008.