International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 07 Issue: 01 | January - 2023 Impact Factor: 7.185 ISSN: 2582-3930

PREDICTION OF PARKINSON DISEASE USING MACHINE LEARNING

Y.Ranga Sai Reddy[1], K.Devi Sri Prasad[2], D.Anurag Ashwin[3], Dr. Subhani Shaik[4]

[1,2,3]Student, [4]Professor

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

[S1,2,3,4]Sreenidhi Institute of Science and Technology, Ghatkesar, Hyderabad, India

ABSTRACT - Parkinson Disease is an increasing problem in the modern period. It might be difficult for a clinician to identify this neurological illness in its early stages. However, early illness detection is critical since therapies have a higher chance of improving patients' quality of life. Since there is no recognized test to identify the symptoms, we propose a statistical technique employing the gait, tremors, and micrographics, which are the most common symptoms of Parkinson disease.

This entails examining the link between the two symptoms and classifying the resulting data using multiple classification algorithms in order to identify the classification algorithm that offers the maximum accuracy in diagnosing the patients.

Early identification may lead to better medical treatment and improved standards of care.

Keywords : Parkinson Disease, Micrographic, Diagnosis.

INTRODUCTION

A gradual deterioration in the structure and/or function of neuronal cells affecting the brain and spinal cord is identified as the Parkinson Disease. Parkinson Disease can be caused by a wide range of undiscovered causes and variables, in addition to having a variety of symptoms. Parkinson Disease takes a very long time to be detected because of its diverse symptoms and rates of development. Because the symptoms get worse as the disease gets worse, more sensitive diagnostic techniques are needed for the diagnosis. In general, the main objective of this endeavor is to find a treatment as quickly as feasible. The possibility that therapies will improve the quality of life for patients and their families rises with early illness identification.

2. LITERATURE VIEW

2.1 PROBLEM STATEMENT

The existing approach mandates that the patient undergo a number of exams, including PET scans, CT scans, and other imaging. The patient would have to wait for the results and their condition might get worse as these tests take a long time to gather results and need a lot of waiting. Therefore, a quick check is required in order to make an earlier diagnosis, which would significantly improve the patient's quality of life. Early Parkinson disease detection is the goal of our investigation.

2.2 EXISITING SYSTEM

The current method of diagnosing Parkinson disease relies on extensive scanning, which takes a lot of time and could make the patient's condition worse.
A few of the current systems are:

- MRI/CT Scan
- PET Scan
- SPECT Scan.

The patient's quality of life would suffer as a result of these tests' high cost, 25% mistake rate, and potential wait times for results.

2.3 PROPOSED SYSTEM :

By using machine learning techniques, the problems can be solved with a low error rate. Parkinson Disease uses gait, tremors, and handwriting samples as their dataset in order to increase accuracy by figuring out the association between these symptoms. The objective of our approach

is to create a model that may be able to accurately predict whether a patient is sick or not by looking at a range of symptoms. When considered separately, each symptom has drawbacks. For instance, in voice recognition, additional steps like noise removal and speech segmentation are required since handwriting is a complex activity in which a variety of circumstances may affect motor movement. It has been established that using breath samples does not consistently yield results that are medically beneficial.

3. COMPONENTS

3.1 SOFTWARE REQUIREMENTS

- Python 3
- Python libraries: Numpy, Pandas.

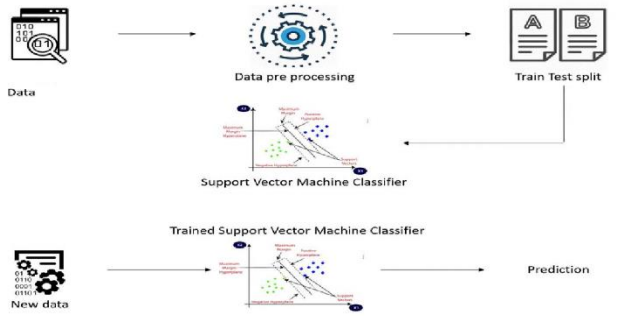
3.2 HARDWARE COMPONENTS

- Processor – intel i5 8th gen
- Memory – 8GB RAM
- Hardisk – 10GB

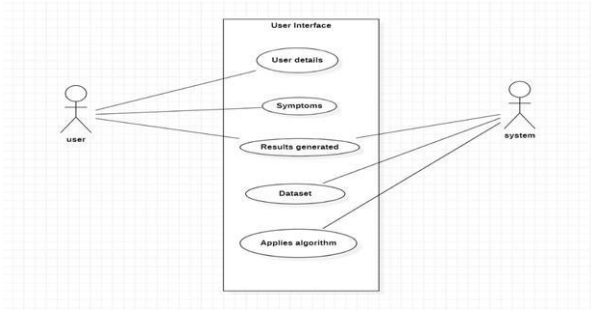
4. SYSTEM DESIGN

Systems design is the process of defining a system's architecture, components, modules, interfaces, and data to satisfy preset requirements. It might be viewed as a systems theory application to the process of product development. The methods that emphasize object-oriented analysis and design are becoming the most widely used for creating computer systems.

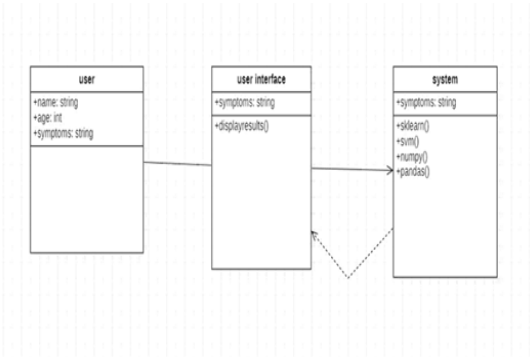
4.1 SYSTEM ARCHITECTURE



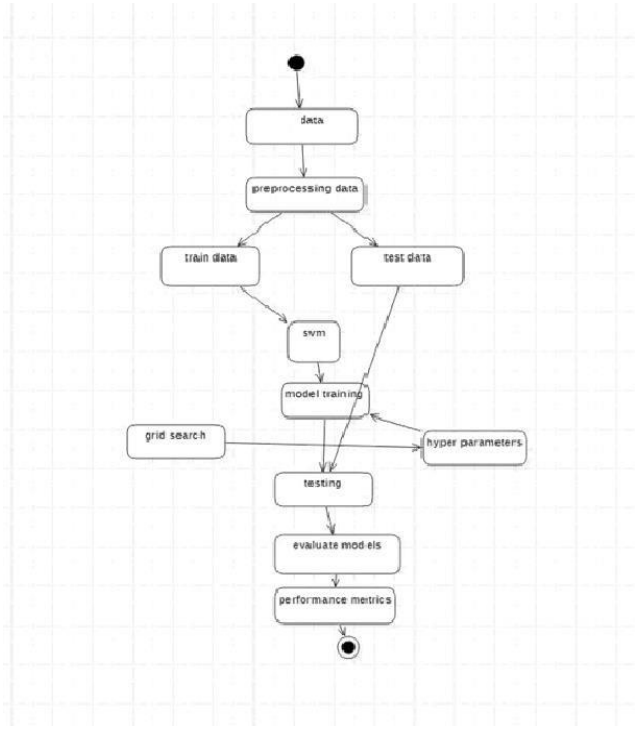
4.2 ULM DIAGRAMS



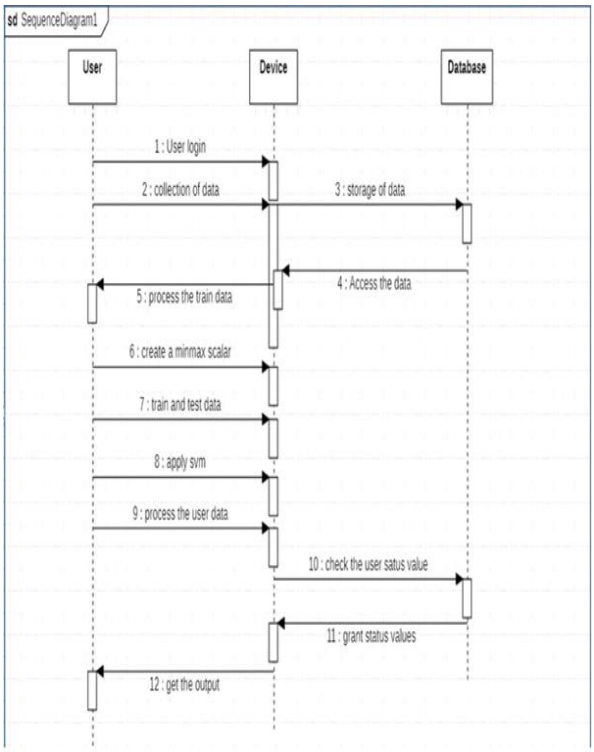
Use-Case Diagram



Class Diagram



Activity Diagram



Sequence Diagram

5. IMPLEMENTATION

5.1 LANGUAGE OR TECHNOLOGY USED

Python : Python is the language used for developing the prediction system. Python supports many libraries which have been used for this ML project. The libraries used for this project are:

- Numpy
- Pandas
- Scikit-Learn

5.2 ALGORITHM

Support Vector Machine (SVM):

A supervised machine learning model called a support vector machine (SVM) employs classification techniques to solve two-group classification problems. An SVM model can classify new text after being given sets of labelled training data for each category. They offer two key advantages over more recent algorithms like neural networks: greater speed and improved performance with fewer samples (in the thousands). As a result, the approach is excellent for text classification issues, where it's typical to only have access to a dataset with a few thousand tags on each sample.

5.3 CODES

```
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
```

Libraries

```
# loading the data from csv file to a Pandas DataFrame
data = pd.read_csv('data.csv')

# printing the first 5 rows of the dataframe
data.head()
```

	name	MDVP:Fo(Hz)	MDVP:Fhi(Hz)	MDVP:Flo(Hz)	MDVP:Jitter(%)	MDVP:Jitter(Abs)	MDVP:RAP	MDVP:PPQ	Jitter:D
0	phon_R01_S01_1	119.992	157.302	74.997	0.00784	0.00007	0.00370	0.00554	0.01
1	phon_R01_S01_2	122.400	148.650	113.819	0.00968	0.00008	0.00465	0.00696	0.01
2	phon_R01_S01_3	116.682	131.111	111.555	0.01050	0.00009	0.00544	0.00781	0.01
3	phon_R01_S01_4	116.676	137.871	111.366	0.00997	0.00009	0.00502	0.00698	0.01
4	phon_R01_S01_5	116.014	141.781	110.655	0.01284	0.00011	0.00655	0.00908	0.01

5 rows × 24 columns

Loading Dataset

```
X = data.drop(columns=['name','status'], axis=1)
Y = data['status']

print(X)
```

```
     MDVP:Fo(Hz)  MDVP:Fhi(Hz)  MDVP:Flo(Hz)  MDVP:Jitter(%)  \
0        119.992       157.302        74.997         0.00784
1        122.400       148.650       113.819         0.00968
2        116.682       131.111       111.555         0.01050
3        116.676       137.871       111.366         0.00997
4        116.014       141.781       110.655         0.01284
..           ...           ...           ...             ...
190      174.188       230.978        94.261         0.00459
191      209.516       253.017        89.488         0.00564
192      174.688       240.005        74.287         0.01360
193      198.764       396.961        74.904         0.00740
194      214.289       260.277        77.973         0.00567

     MDVP:Jitter(Abs)  MDVP:RAP  MDVP:PPQ  Jitter:DDP  MDVP:Shimmer  \
0             0.00007   0.00370   0.00554     0.01109       0.04374
1             0.00008   0.00465   0.00696     0.01394       0.06134
2             0.00009   0.00544   0.00781     0.01633       0.05233
3             0.00009   0.00502   0.00698     0.01505       0.05492
4             0.00011   0.00655   0.00908     0.01966       0.06425
..                ...       ...       ...         ...           ...
190           0.00003   0.00263   0.00259     0.00790       0.04087
191           0.00003   0.00331   0.00292     0.00994       0.02751
192           0.00008   0.00624   0.00564     0.01873       0.02308
193           0.00004   0.00370   0.00390     0.01109       0.02296
194           0.00003   0.00295   0.00317     0.00885       0.01884
```

Preparing the dataset

```
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.2, random_state=2)

print(X.shape, X_train.shape, X_test.shape)
```

```
(195, 22) (156, 22) (39, 22)
```

Separating the train and test data

```
scaler = StandardScaler()

scaler.fit(X_train)
```

```
StandardScaler()
```

```
X_train = scaler.transform(X_train)

X_test = scaler.transform(X_test)

print(X_train)
```

```
[[ 0.63239631 -0.02731081 -0.87985049 ... -0.97586547 -0.55160318
   0.07769494]
 [-1.05512719 -0.83337041 -0.9284778  ...  0.3981808  -0.61014073
   0.39291782]
 [ 0.02996187 -0.29531068 -1.12211107 ... -0.43937044 -0.62849605
  -0.50948408]
 ...
 [-0.9096785  -0.6637302  -0.160638   ...  1.22001022 -0.47404629
  -0.2159482 ]
 [-0.35977689  0.19731822 -0.79063679 ... -0.17896029 -0.47272835
   0.28181221]
 [ 1.01957066  0.19922317 -0.61914972 ... -0.716232    1.23632066
  -0.05829386]]
```

Data Standardization

```
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 :  0.8846153846153846
```

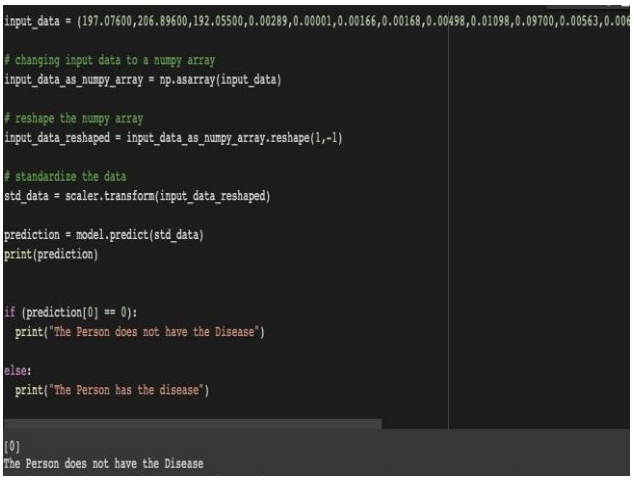
```
# accuracy score on training data
X_test_prediction = model.predict(X_test)
test_data_accuracy = accuracy_score(Y_test, X_test_prediction)

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

```
Accuracy score of test data :  0.8717948717948718
```

Calculating the Accuracy

6.RESULTS



Prediction

7.CONCLUSION AND FUTURE SSCOPE

CONCLUSION

Parkinson Disease involves the central nervous system (CNS) of the brain and is currently untreatable unless caught early. Lack of treatment and life loss result from late discovery. Therefore, it is important to diagnose it early. We used Support Vector Machine for early illness identification. We discovered that SVM is the algorithm that predicts the commencement of this disease with the highest accuracy when compared to other algorithms, enabling early treatment and perhaps saving lives.

FUTURE SCOPE

Future research can concentrate on various methods for predicting the disease utilizing various datasets. In this study, we classify patients using a binary attribute (1- diseased patients, 0- non-diseased patients). In the future, we'll classify people according to the stages of the disease.

8.REFERENCES

- M. Chen, Y. Hao, K. Hwang, L. Wang, and L. Wang, Disease prediction by machine learning over big data from healthcare communities" IEEE Access, vol. 5, no.1, pp.
- X. Bi, S. Li, B. Xiao, Y. Li, G. Wang, and X. Ma, ''Computer aided Alzheimer's disease diagnosis by an unsupervised deep learning technology,'' Neurocomputing, vol. 392, pp. 296–304,Jun. 2020.
- X. Bi, X. Zhao, H. Huang, D. Chen, and Y. Ma, ''Functional brain network classification for Alzheimer's disease detection with deep features and extreme learning machine, Cognit. Compute, vol. 12, no. 3, pp. 513–527, May 2020.
- S. Sharma, R. K. Dudeja , G. S. Aujla, R. S. Bali, and N. Kumar, ''Detras: Deep learning-based healthcare framework for IoT-based assistance of alzheimer patients,'' Neural Computing. Appl., pp. 1–13,Sep. 2020.