Early Stage Prediction of Parkinson's Disease using Neural Network

Chaithra B.R[1], Gowthami A.L[1], Harshitha U[1], Keerthana N[1], Asha V.G[2]

[1]Dept.of Computer Science, Jyothy Institute of Technology, Bangalore, India
[2]Assistant Professor, Dept.of Computer Science, Jyothy Institute of Technology, Bangalore, India

---

Abstract - *Parkinson's disease (PD) is a long-term chronic disorder of the central nervous system that mainly affects the motor system. It disturbs the nerve cells in the brain that produce dopamine. The symptoms of Parkinson's disease includes muscle rigidity, tremors and changes in speech.*

The objective of the project is to analyse and predict the Parkinson's disease in early stages. As the symptoms worsen, it may deteriorate the motor and non-motor system of the patients. Thus, it is important to predict the Parkinson's disease at the early stages. The proposed system concentrates on improving the Parkinson's disease diagnosis with impressive experimental results using Artificial Neural Networks (ANN) and Deep Neural Networks (DNN). It mainly concentrates on the prediction of Parkinson's disease using the deep learning algorithms.

***Key Words*: Parkinson's disease, Deep Neural Networks, Artificial Neural Networks, Prediction.**

1. INTRODUCTION

1.1 Parkinson's disease

Parkinson's disease (PD) is a disorder of the nervous system. It is prevalent throughout the world and mainly affects the patients above 60 years. It is a neurodegenerative disorder that affects predominately dopamine producing neurons in a specific area of the brain called substantia nigra. This leads to a reduction in a chemical called dopamine in the brain. Dopamine plays a vital role in the coordination of movement. It acts as a chemical messenger for transmitting signals within the brain. Due to the loss of these cells, patients suffer from movement disorder. [1] People with Parkinson's disease also lose the nerve endings that produce norepinephrine, the main chemical messenger of the sympathetic nervous system, which controls many automatic functions of the body such as heart rate and blood pressure. The four primary symptoms of PD are tremor, or trembling in hands, arms, legs, jaw and face; rigidity or stiffness of the limbs and trunk; slowness of movement; and postural instability, or impaired balance and coordination. PD is both chronic, meaning it persists over a long period of time, and progressive, meaning its symptoms grow worse over time. Although some people become severely disabled, others experience only minor motor disruptions. Tremor is the major symptom for some individuals, while for others tremor is only a minor complaint and other symptoms are more troublesome. [2] It is currently not possible to predict which symptoms will affect an individual, and the intensity of the symptoms also varies from person to person. In addition to these motor-related symptoms, non-motor symptoms such as cognitive impairment, mood and behavioral problems, sleep disorders, and constipation can significantly impair quality of life and require careful symptom-based treatment. Some non-motor symptoms such as hyposmia (reduced ability to detect odors), REM sleepbehavior disorder and constipation typically precede the motor symptoms by several years. Other non-motor symptoms such as cognitive impairment commonly appear after the onset of motor symptoms. Currently available PD medications do offer valuable symptomatic relief, but as PD progresses, their use is often associated with significant and sometimes intolerable side effects. For example, levodopa, one of the most effective treatments for PD can normalize motor function for years but later cause involuntary muscle movements known as dyskinesia and dystonia (sustained muscle contractions). [4] In addition, people in the mid to late stages of PD often experience a wearing-off of the beneficial effects of PD drugs and a re-emergence of motor and non-motor symptoms before their next scheduled dose. In more advanced PD, drug-resistant motor symptoms (e.g., postural instability, freezing of gait, loss of balance, frequent falls), behavioral changes (impulse control disorders, hallucinations, and psychosis), and often dementia are leading causes of impairment. [3]

1.2 Artificial Neural Network

Artificial neural networks are one of the main tools used in machine learning. As the "neural" part of their name suggests, they are brain-inspired systems which are intended to replicate the way that we humans learn. Neural networks consist of input and output layers, as well as (in most cases) a hidden layer consisting of units that transform the input into something that the output layer can use. They are excellent tools for finding patterns which are far too complex or numerous for a human programmer to extract and teach the machine to recognize.

ANNs are composed of multiple nodes, which imitate biological neurons of human brain. The neurons are connected by links and they interact with each other. The nodes can take input data and perform simple operations on the data. The result of these operations is passed to other neurons. The output at each node is called its activation or node value. Each link is associated with weight. ANNs are capable of learning, which takes place by altering weight values.

1.3 Deep Neural Network

A DNN is a neural network with a certain level of complexity, a neural network with more than two layers. Deep neural networks use sophisticated mathematical modelling to process data in complex ways.

Deep neural networks as networks that have an input layer, an output layer and at least one hidden layer in between. Each layer performs specific types of sorting and ordering in a process that some refer to as "feature hierarchy." One of the key uses of these sophisticated neural networks is dealing with unlabeled or unstructured data. The phrase "deep learning" is also used to describe these deep neural networks, as deep learning represents a specific form of machine learning where technologies using aspects of artificial intelligence seek to classify and order information in ways that go beyond simple input/output protocols.

The DNN finds the correct mathematical manipulation to turn the input into the output, whether it be as linear relationship or a non-linear relationship. The network moves through the layers calculating the probability of each output. DNN architectures generate compositional models where the object is expressed as a layered composition of primitives. The extra layers enable composition of features from lower layers, potentially modelling complex data with fewer units than a similarly performing shallow network.

2. LITERATURE SURVEY

Different researches have used various features and data in order to predict the Parkinson's disease. The authors Zahari Abu Bakar, Nooritawati Md Tahir and Ihsan M Yassin present the Classification of Parkinson's disease based on Multilayer Perceptron Neural Network. In this paper the algorithms used were Levenberg Marquardt and Scaled Conjugate Gradient. The advantage of this model is while the LM algorithm showed accuracy of 97% for training data and 92% accuracy for test data, SCG algorithm showed accuracy of 79% and 78% respectively. The disadvantage is that the above-mentioned accuracy was obtained only for certain number of hidden units that is, in LM algorithm when the number of hidden layers were 25 the aforementioned accuracy was obtained and for SCG the count was hidden layers was 10 for the accuracy obtained.[1]

The authors Sonu S.R, Ravi Ranjan, Vivek Prakash, Saritha.K, in their paper Prediction of Parkinson's Disease using Data Mining, have used Decision tree and Logistic Regression for data of voice recordings of the patients. This model gives a 100% accuracy in decision tree and a 90% accuracy in logistic regression when no feature was selected. However when features such as voice jitter, voice frequency etc. were selected, the prediction only provided a varying accuracy in the range of 84% to 92% for decision tree. [5]

The authors Aarushi Agarwal, Spriha Chandrayan and Sitanshu S Sahu present their paper Prediction of Parkinson's Disease using Speech Signal with Extreme Learning Machine. In this paper the analysis is performed on the voice sample of the patients. The dataset used is taken from the UCI repository and the model yielded 81.55% and this system proved to be more accurate than neural network and support vector machine. [2]

The authors Oana Geman and Luliana Chiuchisan, present the Deep Brain Stimulation efficiency and Parkinson's disease stage prediction using Markov's Models. This model has helped in the stage prediction of Parkinson's and using Deep Brain Simulation procedure that part of the brain which is responsible for the tremors is electrically simulated which relieves it. But, this method is minimally invasive and expensive therefore not suitable for all population groups and these symptoms come back with time as there is no permanent solution. [3]

The authors Mrugali Bhat, Sharvari Inamdar, Devyani Kulkarni, Gauri Kulkarni and Revati Shriram, present the paper Parkinson's disease Prediction based on Hand Tremor Analysis. In this paper the analysis is performed using accelerometer as a sensor for measuring the frequency of tremors caused. The proposed system is non-invasive and reliable for the prediction of Parkinson's disease. But, the prognosis was only based on the tremors that were caused as a symptom of Parkinson's. So, these predictions can be inaccurate for those patients who experience these tremors in later stages of the disease. [4]

3. REQUIREMENT SPECIFICATION

The purpose of the Software Requirement Specification is to reduce the communication gap between the clients and the developers. Software Requirement Specification is the medium though which the client and user needs are accurately specified. It forms the basis of software development. A good SRS should satisfy all the parties involved in the system.

3.1 Hardware Requirements

Processor: Any Processor above 500 MHz

Ram: 4 GB

Hard Disk: 4 GB

Input device: Standard Keyboard and Mouse.

Output device: VGA and High Resolution Monitor.

3.2 Software Requirements

Operating System: Windows 7 or higher

Programming language: Python and related libraries. Software: Anaconda Version 3.6

Dataset is obtained from UCI Repository. It contains 197 instances and 23 attributes. We considered multiple symptoms of patients such as speech and key stroke data for Parkinson's disease prediction.

The attribute columns considered are as shown below in the table 5.1.

Table 5.1: List of attributes

Attribute Name	Attribute description
Name	ASCII subject name and recording number
MDVP:Fo(Hz)	Average vocal fundamental frequency
MDVP:Fhi(Hz)	Maximum vocal fundamental frequency
MDVP:Flo(Hz)	Minimum vocal fundamental frequency
MDVP:Jitter(%), MDVP:Jitter(Abs), MDVP:RAP, MDVP:PPQ, Jitter:DDP	Several measures of variation in fundamental frequency
MDVP:Shimmer, MDVP:Shimmer(dB), Shimmer:APQ3, Shimmer:APQ5, MDVP:APQ, Shimmer:DDA	Several measures of variation in * amplitude
NHR,HNR	Two measures of ratio of noise to tonal components in the voice
Status	Health status of the subject (one) - Parkinson's, (zero) – healthy
RPDE,D2	Two nonlinear dynamical complexity measures
DFA	Signal fractal scaling exponent
spread1,spread2,PPE	Three nonlinear measures of fundamental frequency variation

4. IMPLEMENTATION

4.1 Data Pre-Processing

We have taken Parkinson symptoms in our case study, in which the patient's dataset with speech is considered. Pre-processing of dataset is done for converting the string attributes to numerals and missing data records are dropped.

4.2 Artificial Neural Network

The Artificial Neural network is implemented using Keras, which is a powerful easy-to-use Python library for developing and evaluating deep learning models. The following sub modules are applicable in prediction.

Load Data
Define Model
Compile Model
Fit Model
Evaluate Model

Load data - We have to load the file directly using the Pandas function pd.read_csv(). There are twenty two input variables and one output variable (the last column). Once loaded we can split the dataset into input variables (X) and the output class variable (Y).

Define model - Models in Keras are defined as a sequence of layers. The first thing to get right is to ensure the input layer has the right number of inputs. This can be specified when creating the first layer with the input dim argument and setting it to 8 for the 8 input variables.

Fully connected layers are defined using the Dense class. We can specify the number of neurons in the layer as the first argument, the initialization method as the second argument as init and specify the activation function using the activation argument.

In this case, we initialize the network weights to a small random number generated from a uniform distribution ('uniform'), in this case between 0 and 0.05 because that is the default uniform weight initialization in Keras. Another traditional alternative would be 'normal' for small random numbers generated from a Gaussian distribution. We will use the rectifier ('relu') activation function on the first two layers and the sigmoid function in the output layer. It used to be the case that sigmoid and tanh activation functions were preferred for all layers. These days, better performance is achieved using the rectifier activation function. We use a sigmoid on the output layer to ensure our network output is between 0 and 1 and easy to map to either a probability of class 1 or snap to a hard classification of either class with a default threshold of 0.5.

The first layer has 16 neurons and expects 22 input variables. The second hidden layer has 8 neurons and finally, the output layer has 1 neuron to predict the class (Parkinson disease is there or not).

Compile model - Compiling the model uses the efficient numerical libraries under the covers such as Theano or TensorFlow. When compiling, we must specify some additional properties required when training the network. We must specify the loss function to use to evaluate a set of weights, the optimizer used to search through different weights for the network and any optional metrics we would like to collect and report during training. In this case, we will use logarithmic loss, which for a binary classification problem is defined in Keras as "binary_crossentropy". Finally, because it is a classification problem, we will collect and report the classification accuracy as the metric.

Fit model - We can train or fit our model on our loaded data by calling the fit() function on the model. The training process will run for a fixed number of iterations through the dataset called epochs, that we must specify using the nepochs argument. For this problem, we will run for a small number of iterations (4000).

Evaluate model - Before evaluation we predict the value. Once the prediction is done the evaluation of the value and model takes place.

4.3 Deep Neural Network

Deep Neural network implemented using TFLearn and TensorFlow Python library for developing and evaluating deep learning models. The following sub modules are applicable in prediction.

Load Data

Deep neural model

Training

Prediction

Load data - We have to load the file directly using the Pandas function pd.read_csv(). There are twenty two input variables and one output variable (the last column). Once loaded we can split the dataset into input variables (X) and the output class variable (Y).

Build a Deep Neural model - We are building a 3layers neural network using TFLearn. We need to specify the shape of our input data. In our case, each sample has a total of 22 features and we will process samples per batch to save memory, so our data input shape is [None, NUMBER_OF_FEATURES] ('None' stands for an unknown dimension, so we can change the total number of samples that are processed in a batch).

Training - TFLearn provides a model wrapper 'DNN' that can automatically performs a neural network classifier tasks, such as training, prediction, save/restore, etc.

Prediction - The trained deep neural network can be used for prediction. In our case, we are passing X_test value to the model.predict() to get out as binary class value.

5. RESULT AND DISCUSSION

The proposed work is implemented in Python 3.6.4 with libraries scikit-learn, pandas, matplotlib Tensorflow, Keras and other mandatory libraries. The training dataset of Parkinson disease patients contains 195 samples. Deep learning algorithm is applied such as artificial neural networks and deep neural networks. We used these learning algorithm for Parkinson disease prediction. The result shows that Parkinson detection is efficient using artificial neural networks algorithm. Artificial Neural network achieves 93.2% accuracy while Deep neural network achieves around 73.84% accuracy. The following table shows the accuracy arrived in our experimental study.

Table 5.1: Experimental Results of proposed system

Algorithm	**Accuracy (%)**
Neural networks	93.2
Deep Neural Networks	73.84

A large amount of information represented in graphic form is easier to understand and analyze. Some companies specify that a data analyst must know how to create slides, diagrams, charts, and templates. In our approach, the detection rates of intrusion is shown as data visualization part.

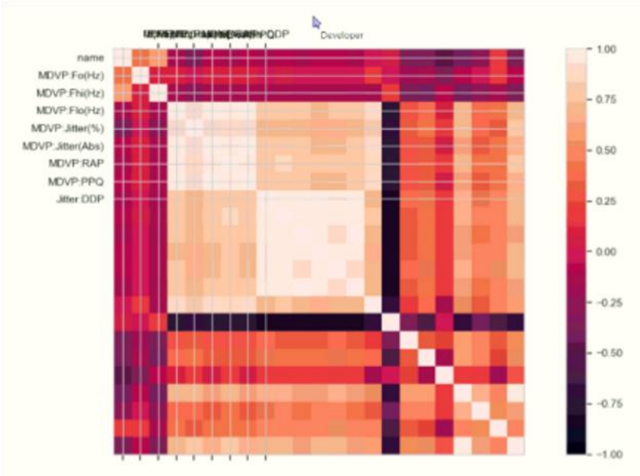

Figure 5.1: Data Visualization of Correlation matrix.

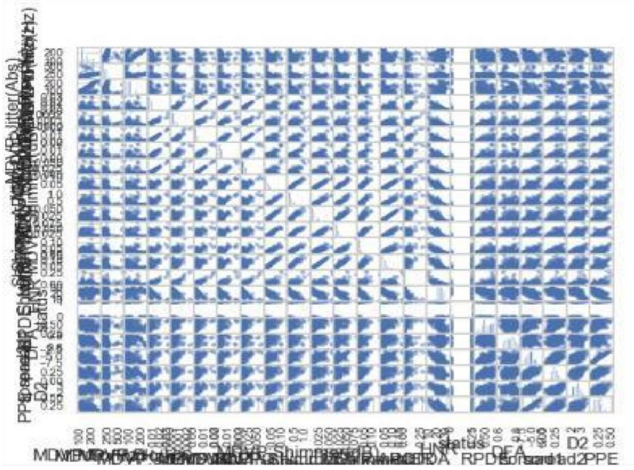

Figure 5.2: Data Visualization of Scatter plot

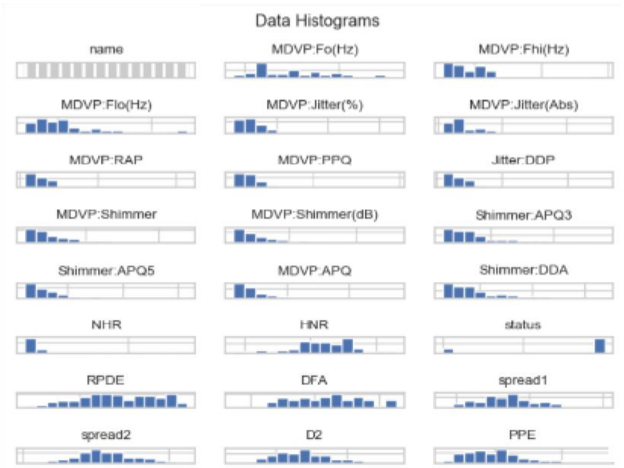

Figure 5.3: Data Visualization of Histogram plot

The below figure shows the evaluation metric of neural network algorithm (Figure 4) and Deep neural network algorithm (Figure 5).

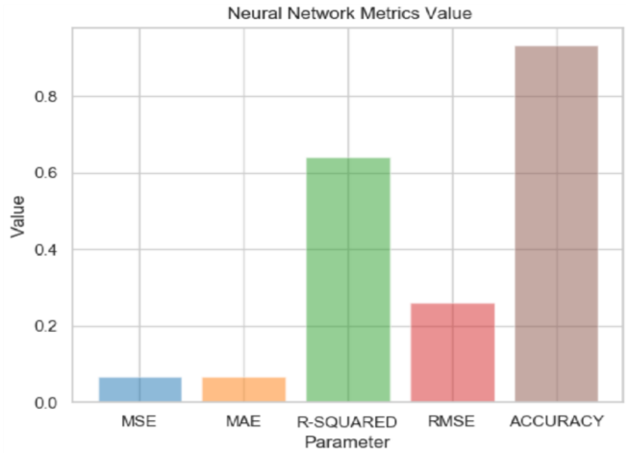

Figure 5.4: Evaluation metric of Artificial Neural network Algorithm.

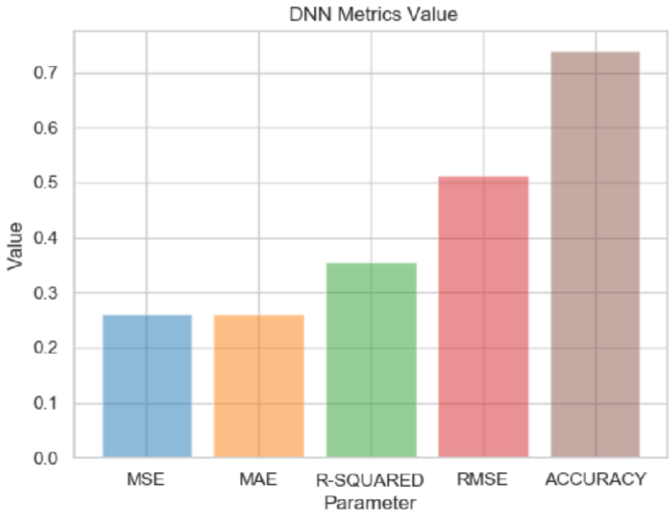

Figure 5.5: Evaluation metric of Deep Neural network Algorithm

6. CONCLUSION

The use of instance learning for detecting Parkinson disease symptoms is studied. Proposed work addressed the formulation of PD symptom detection from weakly labeled data as a semi-supervised multiple instance learning problem. The features were carefully chosen to address the subject and symptom specific nature of the problem. We show promising preliminary results by monitoring performed with the PD subjects.

7. REFERENCES

[1] Zahari Abu Bakar, Nooritawati Md Tahir and Ihsan MYassin, "Classification of Parkinson's Disease Based on Multilayer Perceptron Neural Network", 2010 6thInternational Colloquium on Signal Processing & ItsApplications (CSPA).

[2] Aarushi Agarwal, Spriha Chandrayan and Sitanshu S Sahu," Prediction of Parkinson's Disease using Speech Signal with Extreme Learning Machine", International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT)-2016.

[3] Oana Geman and Luliana Chiuchisan," Deep Brain Efficiency and Parkinson's disease Stage Prediction Using Markov Models", The 5th IEEE International Conference on E-Health and Bioengineering-EHB 2015.

[4] Mrugali Bhat, Sharvari Inamdar, Devyani Kulkarni, Gauri Kulkarni and Revati Shriram," Parkinson's Disease Prediction based on Hand Tremor Analysis", International Conference on Communication and Signal Processing.

[5] Sonu S.R, Ravi Ranjan, Vivek Prakash Saritha K," Prediction of Parkinson's Disease using Data Mining", International Conference on Energy, Communication, Data Analytics and Soft Computing (ICECDS-2017).

[6] Parkinson's disease clinical features and diagnosis by Annett Blochberge and Shelley Jones, Vol 3, December 2011.