[1] V. Navya Sree
[2] S. Srinivasa Rao

# Boosted Autoencoder Decoder Network for Parkinson's Disease Detection with Feature Selection

Journal of Electrical Systems

***Abstract:*** - Parkinson's is a major brain disease that is most common in people who have stressed lives or bustling daily schedules. There are times when this problem ends in death. With 755 characteristics, many experts made a system using machine learning that can tell early on if someone has a sickness and save their life. Like other illnesses, it's hard to tell how the disease will grow or show up because there aren't many signs that are the same for everyone and some symptoms can be different from person to in person. You can tell someone has Parkinson's disease by looking at their motor and non-motor signs. This essay looked at the features that were taken from the recorded sound signals. Image processing methods are expensive ways to look at audio data, but the machine requires to work on recordings in order to work well. In this way, the suggested system looked at the data format's features. The model got rid of the features that weren't needed by combining an improved auto encoder in a gradient boosting method to define the exact hierarchies signs and happenings. Researchers used statistical methods, ML techniques, and more, such as Recursive Feature Elimination and Correlation analysis. The measurements used to judge these methods don't work well because the data has a lot of dimensions. With a 95.18% success rate, the technique uses the suggested method to cut the number of traits down to 94. We change the current autoencoder's design so that it works with the data we have.

***Keywords:*** *Statistical Methods, Feature Selection, High Dimensionality, Auto Encoder, Ensemble, Gradient Boosting*

## I. INTRODUCTION *(HEADING 1)*

This illness starts when nerves in a brain region known as "Nigra" get mutilate. This part controls the contraction of muscles by talking to other parts of the brain [10]. Figure 1 shows some of the problems that people with Parkinson's disease often have.

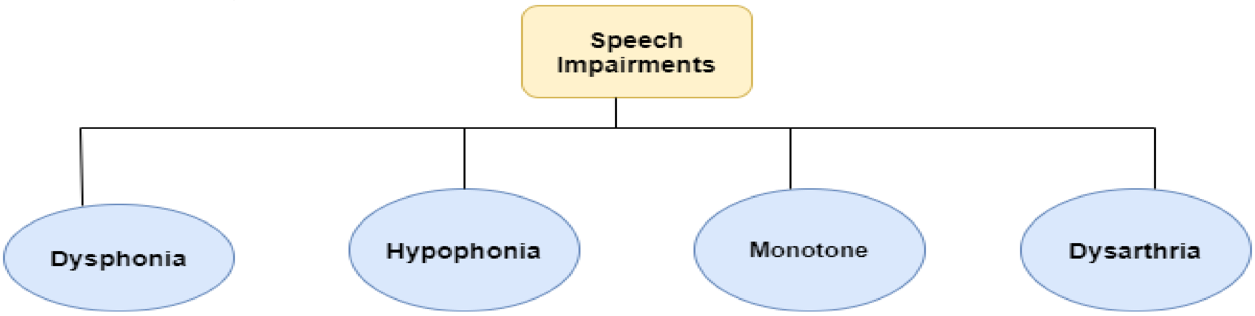


**Figure 1: Different Types of Speech Impairments**

The dataset has six different groups that hold all the measures linked to these disabilities. About seven to eight things make up each group. Along with that, it saves other things like gender, ID, and more. To make the system more accurate [20], the total amount of features needs to be cut down because the dataset possesses more characteristics that were taken from the audio signals. People who work with machine learning usually use machine learning methods to choose which features to emphasize. As demonstrated in Figure 2, the high-dimensionality of the data means that feature extraction takes a while during the deep learning process.

The suggested study used an integrated mechanism with a hierarchy of choices because the information is in csv format and either the traits and the goal name are in numerical form [11]. With 755 characteristics, the collection has a lot of information. The suggested model used a 5-fold cross-validation to get the data ready and entropy to figure out how important each trait was [12]. In equation (1), is able to observe the calculation.

$$\text{Entropy(Attribute)}=-\frac{P_A}{P} * \log_2 \frac{P_A}{P} \text{- (1)}$$

The quality will be more important if the entropy number is lower. After looking at all the possible combos, the model figures out the mistake rate for each option. It doesn't matter which path has more errors. [13] To make the tree, the model uses the entropy measure to choose an attribute as the starting point for each pair and then builds the tree's branches. A unique value helps a pure node rapidly recognize the class label. As a cross-validated approach, it takes an average score from each attribute and uses the one with the most votes to determine which one is more important.

[1] *Corresponding author: Department of Computer Science and Engineering, Koneru Lakshmaiah Education Foundation, Vaddeswaram , AP, India
[2] Department of Computer Science and Engineering, Koneru Lakshmaiah Education Foundation, Vaddeswaram , AP, India

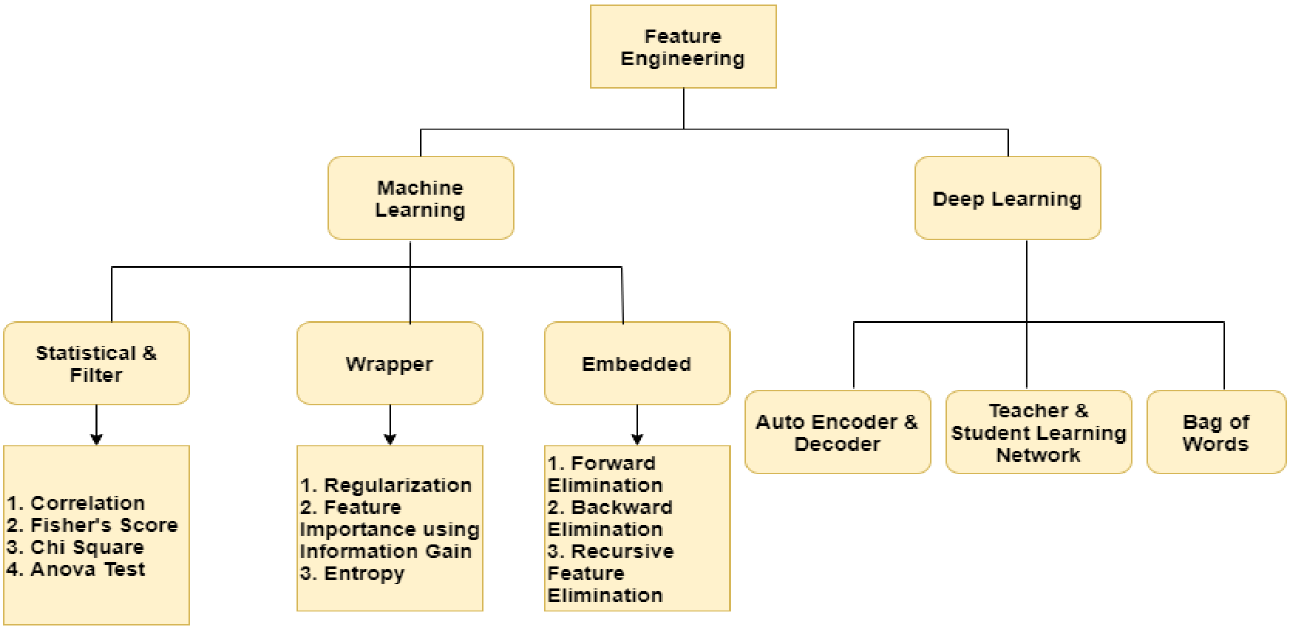


**Figure 2: Classification of Feature Engineering**

## II. LITERATURE SURVEY

Lipsita Sahu et al. [1] has come up with several combined approaches for solving the brain health problems that are very complicated. The process is easy to understand and only has two steps. You can use either digitized or written data to drive the preprocessing of either way. People have chosen two different methods. Through this method, the writer used four types. factors to come up with get the ideal outcomes and finish your job using the suggested method. As of now, the suggested way is what is used to get info from those two steps. At first, there is only one way to preprocess. Now neural is picked because it can join to many layers. This means that the disease is found and the right solutions are given to fix the problem. By comparing various ways, it is clear that the suggested approach does 93.46% superior to the others in DL.

Based on the person's age and gender, Mohamed Shaban et al [2] picked the ANN method to identify them. It is better to use a collection because it has all of the patients' findings and symptoms. Categories for that info are age & gender. This splits the information into two groups, 16 & 15. All are turned into Excel files, which are then sent to the suggested method along with three sets of location [14]. It starts with tests and training info, then goes to the spatial to check the patients' information. The results are known for 80% of the training and 20% of the tests. Only two approaches are voted as one and zero after testing and training. All modes were optimized for efficiency, which was defined as 98%, 97%, or 100%. Since there are so many other spatial steps to learn more about how it is affected, it can give people the best way to look at it.

[3] Mehedi Masud et al. suggested combining both DP and CROW together to understand PD speech problems. There are 754 different kinds of features in this. In total, there are 36 parts and 7 traits. It removes all unnecessary data in a preparation step based on what the author wants. The next traits to be pulled for that ASCA will be utilized as well as the data is taken out to clean it up. This autoencoder is what pulls out the layers. It uses a classification method and a certain number of levels to find the right traits and values. In this case, the secret levels are more important than the other layers. Pre-trained ways get better results, and the suggested method does the best job of speech identification. Combining techniques has made the suggested method 96% more effective.

[4] Sukhpal Kaur et al. utilized deep CNN to find the MRI & figure out what was wrong with the person. MR has information about both impacted and healthy people in the dataset. It goes through picture preparation, gets rid of all the inappropriate images, and finds the right images for each of the three tracks. As the train set learned at the start, GAN is used to change data. Once more, data is fed into Alex-Net and slightly changed. After Alex-Net is fully made, those are transmitted to loss that happened during the transmittion. It reaches on time for the choice, which ends action. After making a choice, CNN sorts things into groups based on how true the decision was. The choice is wrong. On the other hand, it works for training. The step for confirmation goes straight into the step for move Alex-Net and stays the same. Classification is the first step in the test. This means that 89.23% was the highest level of productivity.

[5] Changqin Quan et al. used LSTM and speech to find voices and tell the difference between people who are affected and those who are not. As everyone knows, the dataset has both good and bad information. Based on the data overview, it should be able to tell. This involves finding data from speech, which is hard to do because of the low and elevated frequencies. To get around all of these problems, the system uses intonation to correctly identify the speech and study it. This whole process, which involves getting data and breaking it down into a amount of attributes based on triple phases—the initiating, finishing, & suggested method layers—is known as LSTM. ML work with four different ways to compare and work with the system to rate the process and technique. The suggested methods are the only ones that have worked most effectively in all evaluation techniques. Two validation features work together to give the best efficiency, with a high success rate 84.29%.

[6] Kevin H. Leung et al. came up with triple ways to use deep learning to find the particular illness in the resulting datasets. It uses a collection of pictures and three steps of sorting. The file is made up of scanned pictures that have been changed in one way. There may be high-projected ones on the scan that will go through the brain part and then change back to the first method. Getting groups in method is the second step. The third thing is that all the resulting traits are put together as phase to get appropriate right result. The last step is to join to the network. It uses DP methods for the layers and has some values that separate them into different sections. Then they are forecast and checked to see how well they work. In order to compare the methods, the writer picked three different support techniques that worked 84% of the time.
[7] Shigeki Aoki et al. suggested a DKI-weight, & DL-based methods were utlizied by telling the difference between people who were sick and those who were not. There are the same number of impacted and untouched patient information in the collection. For the study of a patient, the method only looks at the weighted matrix. There are two parts to the extraction: weight and NOS. The person can change CNN to fit their needs in this way. Because not every MRI shows the same thing, it's important to match the picture to the state of the patient before figuring out what the matrix is. The author did this by applying the CAM method to the DL method. The suggested method works 89% of the time. It came from a number of different evaluation methods, but it's hard to make CNN work with a dataset that you don't know much about.
[8] Luis Sigcha et al. used the Multitask CNN method to look at present popular which related to watches & wellness. The info has written words and a tracker in it. Health monitoring on smartwatches uses this method. The main goal of the suggested method is to find the person even when they are not online. Known as a signal, it requires data filtering and extraction. These methods sort and find the right data for training and evaluating them. The discovery uses deep learning methods to figure out the state of a dead person. ml and dl both go through this process. From initialization up to extraction, DL uses the multiple CNN method. The steps are the same once the model has been used for both written language and motion, and the exam has been completed. If you use the suggested method, you will get two different results. The author employed two main ways to rate the techniques and has reached 95% and 97%. To find holes, Table 1 looks at how well the current works do their job.

**Table 1: Pros and cons of using systems that use machine learning to find Parkinson's disease**

| S.No. | Author | Algorithm | Merits | Demerits | Accuracy |
|---|---|---|---|---|---|
| 1 | Lipsita Sahu et al | ANN and RA | Both of them are labourers best way as they can. So, there is a high level of speed. | | 93.46% |
| 2 | Mohamed Shaban | ANN | This applies ANN to three different spatial spaces. Therefore, there are a great number of examples that may be identified to treat PD. | The process of dividing up requires a significant amount of time as well as a great deal of information. | PD – 98%<br>Sensitivity – 97%<br>Specificity – 100% |
| 3 | Mehedi Masud et al | DL and ACSA, Crowd | This method does a great job of recognising speech. | For the solo method, this is mostly what you need. | 96% |
| 4 | Sukhpal Kaur et al | Deep CNN | Best choice for picture editing is it. | Not being able to identify something if the picture quality is bad. | 89.23% |
| 5 | Changqin Quan et al | BI-LSTM | Disease is named and put into a category with the assistance of voice. | It takes longer to find the voice when there is too much filtering. | 84.29% |
| 6 | Kevin H. Leung et al | DL | It functions on both picture & text information. | The method is not worth it for other datasets. | 84% |
| 7 | Shigeki Aoki et al | CNN | The best way to work with pictures is to use identity mapping. | Without knowing enough about the information, it's hard to make CNN fit your needs. | 89% |
| 8 | Lusi Sigcha et al | Multitask CNN | When it comes to technological aspects, no one has done a better job than CNN. | There are two ways to take extraction individually. | Context – 97%<br>Tremor – 95% |

## III. Proposed Methodology

### *A. Dataset Description*

From the speech samples that were taken [15], the model tracks the traits that are linked to time and frequency. It is possible to divide all the factors related to traits into six groups. In order to classify groups, we have the following: The name of all the measurements that deal with jitter is "Frequency." "Amplitude" is the name of the measures that deal with shimmer. Median, standard deviation, and mean pitch values affect pitch parameters [16]. Harmonic factors store numbers that have to do with noise and association measures. "Pulse Parameters" are the numbers that describe the rhythmic bursts. In Figure 3, you can see an example of an image from the information in this study.

| id | gender | PPE | DFA | RPDE | numPulses | numPeriodsPulses | meanPeriodPulses | stdDevPeriodPulses | locPctJitter | locAbsJitter | rapJitter | ppq5Jitter | ddpJitter | locShimmer | locDbShimmer | apq3Shimmer | apq5Shimmer | apq11Shimmer | ddaShimmer |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 0.85247 | 0.71826 | 0.57227 | 240 | 239 | 0.00806353 | 8.68E-05 | 0.00218 | 1.76E-05 | 0.00067 | 0.00129 | 0.002 | 0.05883 | 0.517 | 0.03011 | 0.03496 | 0.04828 | 0.09034 |
| 0 | 1 | 0.76686 | 0.69481 | 0.53966 | 234 | 233 | 0.00825826 | 7.31E-05 | 0.00195 | 1.61E-05 | 0.00052 | 0.00112 | 0.00157 | 0.05516 | 0.502 | 0.0232 | 0.03675 | 0.06195 | 0.06961 |
| 0 | 1 | 0.85083 | 0.67604 | 0.58982 | 232 | 231 | 0.00833959 | 6.04E-05 | 0.00176 | 1.47E-05 | 0.00057 | 0.00111 | 0.00171 | 0.09902 | 0.897 | 0.05094 | 0.06497 | 0.07772 | 0.15282 |
| 1 | 0 | 0.41121 | 0.79672 | 0.59257 | 178 | 177 | 0.01085773 | 0.000182739 | 0.00419 | 4.55E-05 | 0.00149 | 0.00268 | 0.00446 | 0.05451 | 0.527 | 0.02395 | 0.02857 | 0.04462 | 0.07185 |
| 1 | 0 | 0.3279 | 0.79782 | 0.53028 | 236 | 235 | 0.00816157 | 0.002668863 | 0.00535 | 4.37E-05 | 0.00166 | 0.00227 | 0.00499 | 0.0561 | 0.497 | 0.02909 | 0.03327 | 0.05278 | 0.08728 |
| 1 | 0 | 0.5078 | 0.78744 | 0.65451 | 226 | 221 | 0.0076312 | 0.002696381 | 0.00783 | 5.97E-05 | 0.00232 | 0.00312 | 0.00697 | 0.07752 | 0.678 | 0.03805 | 0.04767 | 0.06451 | 0.11415 |
| 2 | 1 | 0.76095 | 0.62145 | 0.54543 | 322 | 321 | 0.00599099 | 0.000107266 | 0.00222 | 1.33E-05 | 0.00036 | 0.00094 | 0.00108 | 0.03203 | 0.28 | 0.0155 | 0.01971 | 0.03274 | 0.0465 |
| 2 | 1 | 0.83671 | 0.62079 | 0.51179 | 318 | 317 | 0.00607386 | 0.000135739 | 0.00282 | 1.71E-05 | 0.00034 | 0.00088 | 0.00103 | 0.063 | 0.539 | 0.02949 | 0.04091 | 0.06445 | 0.08848 |
| 2 | 1 | 0.80826 | 0.61766 | 0.50447 | 318 | 317 | 0.00605719 | 6.93E-05 | 0.00161 | 9.73E-06 | 0.00027 | 0.00068 | 0.00081 | 0.02783 | 0.244 | 0.01376 | 0.0176 | 0.02698 | 0.04129 |
| 3 | 0 | 0.85302 | 0.62247 | 0.54855 | 493 | 492 | 0.00391022 | 3.99E-05 | 0.00075 | 2.93E-06 | 9.00E-05 | 0.00025 | 0.00027 | 0.0567 | 0.512 | 0.02692 | 0.03344 | 0.0563 | 0.08077 |
| 3 | 0 | 0.80657 | 0.67256 | 0.61745 | 488 | 487 | 0.00395611 | 5.38E-05 | 0.00083 | 3.29E-06 | 0.0001 | 0.00026 | 0.00029 | 0.06639 | 0.641 | 0.03747 | 0.03516 | 0.05412 | 0.11241 |
| 3 | 0 | 0.82653 | 0.58326 | 0.44555 | 498 | 497 | 0.00387269 | 3.26E-05 | 0.00069 | 2.68E-06 | 7.00E-05 | 0.00021 | 0.00022 | 0.02531 | 0.218 | 0.01283 | 0.0138 | 0.02256 | 0.03849 |
| 4 | 0 | 0.8726 | 0.78996 | 0.78026 | 492 | 491 | 0.00392415 | 6.72E-05 | 0.0028 | 1.10E-05 | 0.00077 | 0.00184 | 0.0023 | 0.20811 | 1.814 | 0.08936 | 0.14476 | 0.27669 | 0.26808 |
| 4 | 0 | 0.81148 | 0.76831 | 0.70809 | 305 | 304 | 0.00631642 | 0.003245324 | 0.00341 | 2.16E-05 | 0.00093 | 0.00141 | 0.0028 | 0.13878 | 1.326 | 0.0722 | 0.0901 | 0.11271 | 0.21661 |
| 4 | 0 | 0.80978 | 0.77992 | 0.6918 | 291 | 290 | 0.00662419 | 0.002756584 | 0.00457 | 3.03E-05 | 0.00159 | 0.00292 | 0.00477 | 0.13069 | 1.222 | 0.07043 | 0.09023 | 0.10685 | 0.21129 |
| 5 | 1 | 0.81471 | 0.61483 | 0.33216 | 300 | 299 | 0.00643329 | 3.88E-05 | 0.00085 | 5.45E-06 | 0.00017 | 0.00042 | 0.00051 | 0.04046 | 0.354 | 0.01756 | 0.02433 | 0.0469 | 0.05269 |
| 5 | 1 | 0.83269 | 0.62018 | 0.37051 | 286 | 285 | 0.00675426 | 5.17E-05 | 0.00111 | 7.52E-06 | 0.00024 | 0.00059 | 0.00072 | 0.02995 | 0.266 | 0.01211 | 0.01732 | 0.03501 | 0.03632 |
| 5 | 1 | 0.82016 | 0.63124 | 0.37031 | 266 | 265 | 0.00725672 | 4.86E-05 | 0.00086 | 6.28E-06 | 0.0002 | 0.00045 | 0.00059 | 0.02734 | 0.241 | 0.01309 | 0.01662 | 0.02548 | 0.03926 |
| 6 | 1 | 0.78067 | 0.66085 | 0.44583 | 283 | 282 | 0.00682409 | 0.000138247 | 0.00177 | 1.21E-05 | 0.00025 | 0.00061 | 0.00075 | 0.0481 | 0.422 | 0.02602 | 0.02692 | 0.03984 | 0.07806 |
| 6 | 1 | 0.79774 | 0.71199 | 0.36714 | 289 | 288 | 0.00669304 | 6.49E-05 | 0.00122 | 8.19E-06 | 0.0002 | 0.00049 | 0.00061 | 0.08552 | 0.741 | 0.04596 | 0.05921 | 0.06201 | 0.13788 |
| 6 | 1 | 0.82169 | 0.62901 | 0.36176 | 292 | 291 | 0.00662361 | 2.80E-05 | 0.00084 | 5.58E-06 | 0.00018 | 0.00041 | 0.00055 | 0.02324 | 0.205 | 0.01087 | 0.0125 | 0.02123 | 0.03262 |
| 7 | 1 | 0.43551 | 0.81029 | 0.71652 | 267 | 266 | 0.00722688 | 0.000200238 | 0.00653 | 4.72E-05 | 0.00244 | 0.00389 | 0.00731 | 0.22066 | 1.891 | 0.09811 | 0.10565 | 0.06615 | 0.29433 |
| 7 | 1 | 0.7622 | 0.73507 | 0.75672 | 175 | 165 | 0.00967745 | 0.00321833 | 0.01268 | 0.000123 | 0.00494 | 0.00499 | 0.01482 | 0.13048 | 1.246 | 0.0632 | 0.06968 | 0.08788 | 0.18961 |

**Figure 3: Information taken from the people's audio recordings**

### *B. Working*

In systems that use standard machine learning methods, the number of factors doesn't go down by a lot. Since audio transmissions include high-dimensional characteristics [17]. For picking out the most important traits, the suggested system made an improved autoencoder and decoder. Figure 4 shows the suggested model's design, and this part talks about how it works.

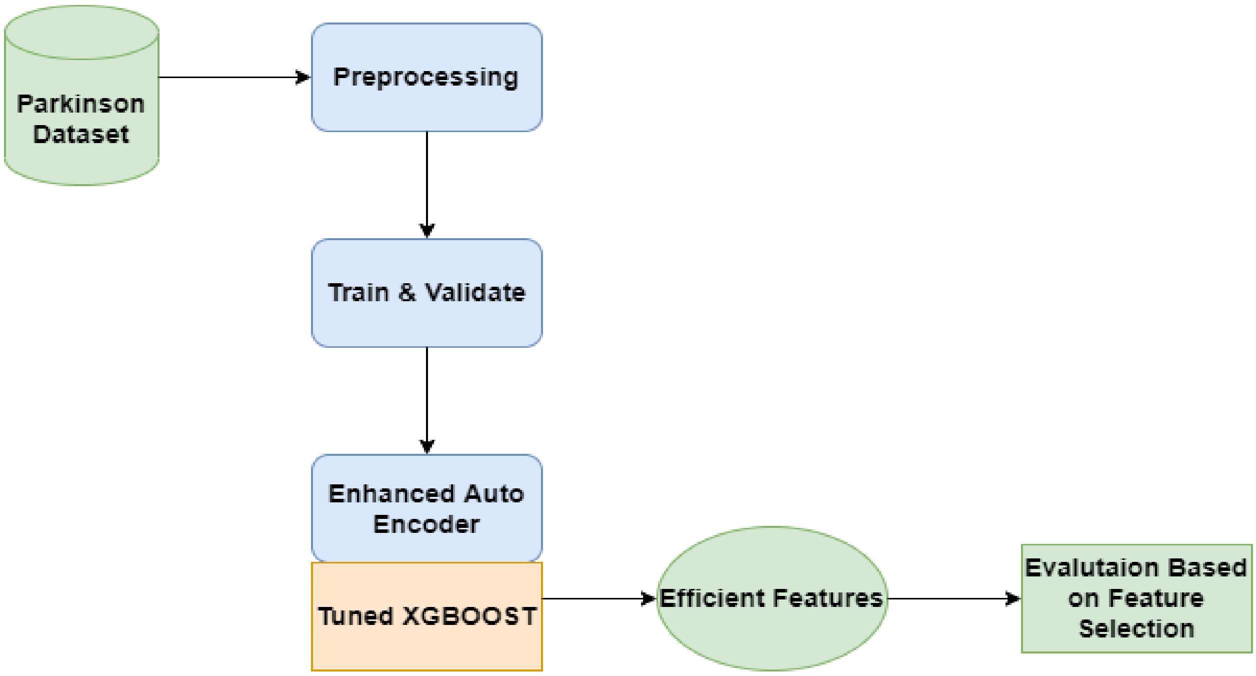


**Figure 4: Selecting Features with an Improved Auto Encoder**

**3.1. Cleaning and Transforming Data:** The whole dataset's empty merit have been filled with the mean value, which is based on classification label analysis. Different people with Parkinson's disease have different problems with hearing [18]. To make the model more consistent, it is very important to collect data in different number forms. The suggested method uses linear scaling since the disabilities have numbers with different ranges. Equation (2) shows how to do the math.

$$Transform\_value(Attribute) = \frac{Old\_Attribute\_value - maximum(Attribute)}{maximum(Attribute) - minimum(Attribute)} \quad - (2)$$

**3.2. Selection of Features by Utilizing EADNN:** Make use of auto encoder & decoder technologies in order to model certain characteristics. The model does a reconstruction of the input by constructing an encoder & a decoder. An encoder maps the inputted info for lower dimensionality by making the features smaller. [19] The bottleneck layer stores aspects in their internal implementations, such as "latent dimensions." The model that was suggested looks at the regularized process where the layers are smaller than the amount of features that were sent in. During the training phase, this model learns from data in lower dimensions. It then codes the most important traits while showing data in higher dimensions. Although noise happened during the transfer from one layer to the next, it is simply ignored. It is vital to note that the following components make up this NN:

i. The encoder receives input from each of the training and testing data, compresses it into a space of latent information model at the bottleneck layer. The suggested model used normalization, two rounds of dense layers, and then two rounds of leaky ReLu layers instead of the typical pooling layers after an input layer. The primary objective of the proposed technique is to retain all data in the input and output phases. Consequently, the dense components are used to establish connections between each neuron and the neurons in the layer below it [9]. Figure 5 shows the computer calculations for the thick layer.

$$\underset{3*3}{\begin{pmatrix} f11 & f12 & f13 \\ f21 & f22 & f23 \\ f31 & f32 & f33 \end{pmatrix}} * \underset{3*1}{\begin{pmatrix} c11 \\ c21 \\ c31 \end{pmatrix}} \xrightarrow{\text{Vector Multiplication}} \underset{3*1}{\begin{pmatrix} f11*c11+f12*c21+f13*c31 \\ f21*c11+f22*c21+f23*c31 \\ f31*c11+f32*c21+f33*c31 \end{pmatrix}}$$

**Figure 5: Vector Multiplication in Dense Layers**

The network can now access more parts than it could before. So the model could lose its ability to stay stable and work well. You can fix this by giving the layers the normalization method. Equation (3) shows how to do the standardization math.

$$Normalized\_output = \gamma * \frac{\sum f_i - \mu_f}{\sqrt[2]{\frac{1}{n} * \sum_{i=0}^{n} (f_i - \mu_f)^2}} \quad - (3)$$

Because there are more factors, the system may have a disappearing gradient problem. This happens when negative values are replaced with zero values, which means the gradient goes away. This issue is solved by using Leaky ReLu, which specifies negative values as tiny slopes as a layer rather than an activation function. It is possible to write the activation function in the form of mathematics in equation (4).

$$f(input\_value) = \max(0, input\_value), input\_value \geq 0$$
$$= \frac{1}{\alpha} * \text{input\_value, input\_value} < 0 \quad - (4)$$

In Figure 6, you can see the layer-by-layer design for the part that records.

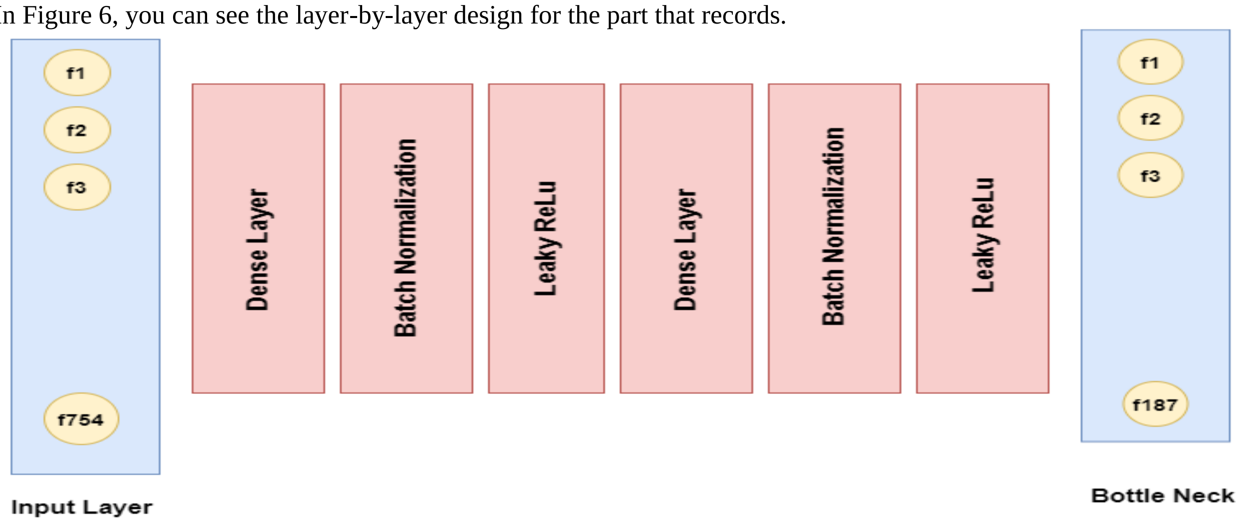


**Figure 6: A Neural Network Containing a Layered Architecture of the Encoder**

ii. The bottleneck moves the most important data from the encoder to the decoder but saves all the data that was taken from the important parts. This part sets up a link between the original picture and the rebuilt image and stops the overfitting problem from happening.

iii. The decoder compresses the bottleneck data and reassembles the signals by upsampling the info gleaned through conv phases. The suggested model normalises the info with those help of the ReLu activation function. Figure 7 displays the general structure of the improved encoder.

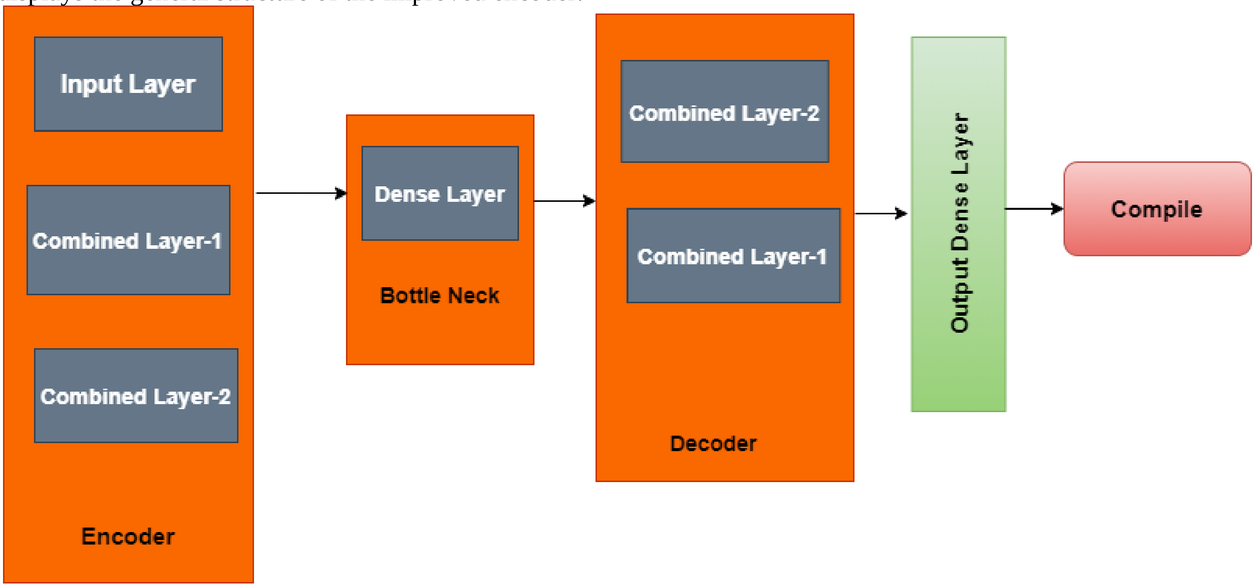


**Figure 7: Enhanced Auto Encoder Architecture**

It takes the decoded features as input and uses them to do more than just classify and regression. It also figures out the rankings for the characteristics. The information is first split into testing and training parts so that the model can be trained and tested. By giving random records to different models and thinking that all traits have the same weight, boosting algorithms work. However, some models don't work as well because they make mistakes. The boosting algorithm takes smaller methodology and builds not a single powerful form through lowering how much weight each the right categories & raising weights to records that were wrongly classified. The model has a better chance of getting the records of wrong classification after the update; it is these documents that then sent for the subsequent methodology. The procedure keeps going until the number of mistakes is as low as possible. Equation (5) shows how to figure out the mistake rate.

$$Error_{Rate} = -(\sum_{i=1}^{n} classlabel_i * \log(predicted)) + (1 - predicted)\log(1 - predcited) \text{ - (5)}$$

## IV. Results

With the use of a DT, the RFE algorithm was able to identify the characteristics that appear in Figure 8. There are now only 678 characteristics, which is still called high-dimensional data.

```
The optimal number of features: 678
[False False False False False False False False False False False False
 False False False False False False False False False False False False
 False False False False False False False False False False False False
 False False False False False False False False False False False False
 False False False False False False False False False False False False
 False False False  True  True  True  True  True  True  True  True  True
  True  True  True  True  True  True  True  True  True  True  True  True
  True  True  True  True  True  True  True  True  True  True  True  True
  True  True  True  True  True  True  True  True  True  True  True  True
  True  True  True  True  True  True  True  True  True  True  True  True
  True  True  True  True  True  True  True  True  True  True  True  True
  True  True  True  True  True  True  True  True  True  True  True  True
```

**Figure 8: Choosing features with the RFE Integrated Decision Tree**

In addition to the block's design, Fig 9 illustrates the first epochs of time during which to and from the encoder function. Many epochs are set up at the start of the model, which makes it less accurate when testing and training. As time goes on, the accuracy gets better.

```
Epoch 1/500
5/5 - 2s - loss: 0.0842 - accuracy: 0.0199 - val_loss: 0.0604 - val_accuracy: 0.0000e+00 - 2s/epoch - 350ms/step
Epoch 2/500
5/5 - 1s - loss: 0.0190 - accuracy: 0.0331 - val_loss: 0.0391 - val_accuracy: 0.0000e+00 - 608ms/epoch - 122ms/step
Epoch 3/500
5/5 - 1s - loss: 0.0149 - accuracy: 0.0298 - val_loss: 0.0376 - val_accuracy: 0.0000e+00 - 608ms/epoch - 122ms/step
Epoch 4/500
5/5 - 1s - loss: 0.0120 - accuracy: 0.0050 - val_loss: 0.0378 - val_accuracy: 0.0000e+00 - 622ms/epoch - 124ms/step
Epoch 5/500
5/5 - 1s - loss: 0.0099 - accuracy: 0.0083 - val_loss: 0.0384 - val_accuracy: 0.0000e+00 - 610ms/epoch - 122ms/step
Epoch 6/500
5/5 - 1s - loss: 0.0083 - accuracy: 0.0033 - val_loss: 0.0380 - val_accuracy: 0.0000e+00 - 613ms/epoch - 123ms/step
Epoch 7/500
5/5 - 1s - loss: 0.0073 - accuracy: 0.0033 - val_loss: 0.0374 - val_accuracy: 0.0000e+00 - 606ms/epoch - 121ms/step
Epoch 8/500
5/5 - 1s - loss: 0.0065 - accuracy: 0.0099 - val_loss: 0.0366 - val_accuracy: 0.0000e+00 - 606ms/epoch - 121ms/step
Epoch 9/500
5/5 - 1s - loss: 0.0057 - accuracy: 0.0116 - val_loss: 0.0363 - val_accuracy: 0.0000e+00 - 611ms/epoch - 122ms/step
Epoch 10/500
5/5 - 1s - loss: 0.0052 - accuracy: 0.0116 - val_loss: 0.0359 - val_accuracy: 0.0000e+00 - 588ms/epoch - 118ms/step
Epoch 11/500
5/5 - 1s - loss: 0.0049 - accuracy: 0.0116 - val_loss: 0.0360 - val_accuracy: 0.0000e+00 - 605ms/epoch - 121ms/step
Epoch 12/500
5/5 - 1s - loss: 0.0049 - accuracy: 0.0182 - val_loss: 0.0363 - val_accuracy: 0.0000e+00 - 609ms/epoch - 122ms/step
Epoch 13/500
5/5 - 1s - loss: 0.0045 - accuracy: 0.0182 - val_loss: 0.0391 - val_accuracy: 0.0066 - 595ms/epoch - 119ms/step
Epoch 14/500
5/5 - 1s - loss: 0.0044 - accuracy: 0.0166 - val_loss: 0.0419 - val_accuracy: 0.0066 - 606ms/epoch - 121ms/step
```

**Figure 9: Learning a NN with 500 Iterations**

Following the completion of each CNN layer, Figure 10 shows an overview of the neural network. As a "class name" in classification, each of the 754 characteristics in the input layer is used to start the process. It talks about how many trainable and non-trainable factors there are in the neural network's summary report. When you work with deep learning methods instead of normal methods for image processing, you may be able to train fewer parameters.

```
Model: "model_3"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 input_7 (InputLayer)        [(None, 754)]             0

 dense_15 (Dense)            (None, 47)                35485

 dense_16 (Dense)            (None, 754)               36192

 batch_normalization_10 (Bat  (None, 754)              3016
 chNormalization)

 leaky_re_lu_10 (LeakyReLU)  (None, 754)               0

 dense_17 (Dense)            (None, 3016)              2277080

 batch_normalization_11 (Bat  (None, 3016)             12064
 chNormalization)

 leaky_re_lu_11 (LeakyReLU)  (None, 3016)              0

 dense_18 (Dense)            (None, 754)               2274818

=================================================================
Total params: 4,638,655
Trainable params: 4,631,115
Non-trainable params: 7,540
_________________________________________________________________
```

**Figure 10: Report on the Customised Neural Network in Brief**

Figure 11 shows how accurate both the train information and the test data were. People have noticed that as the amount of epochs goes up, so does the model's accuracy. This picture shows the accuracy curve for all 100 epochs, making it easy to see. On the graph, displayed on the X-axis is the overall amount of periods of time, while the Y-axis displays readings with accuracy values.

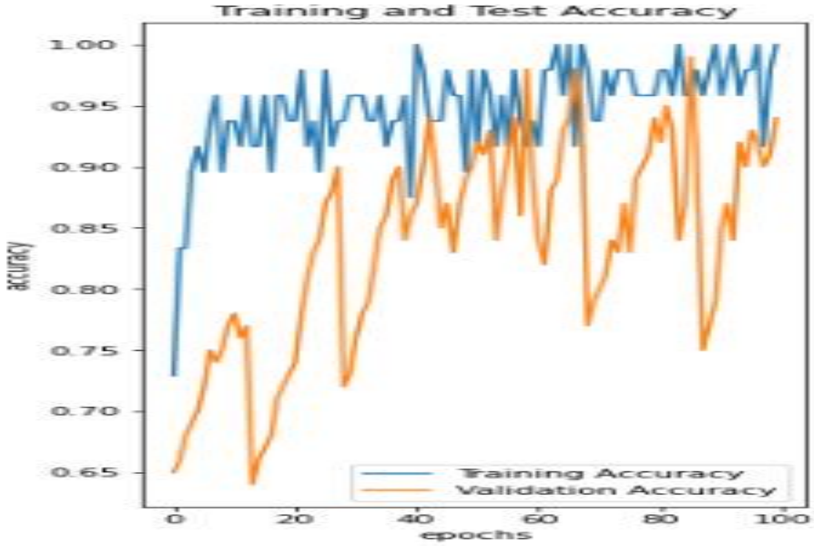


**Figure 11: Accuracy Training & Testing for 100 Epochs**

Fig 12 shows the suggested model along alongside the different rating measures for models that already exist. The suggested method is more accurate by almost 1.1%, more precise by 2%, and more recalling by 2%.

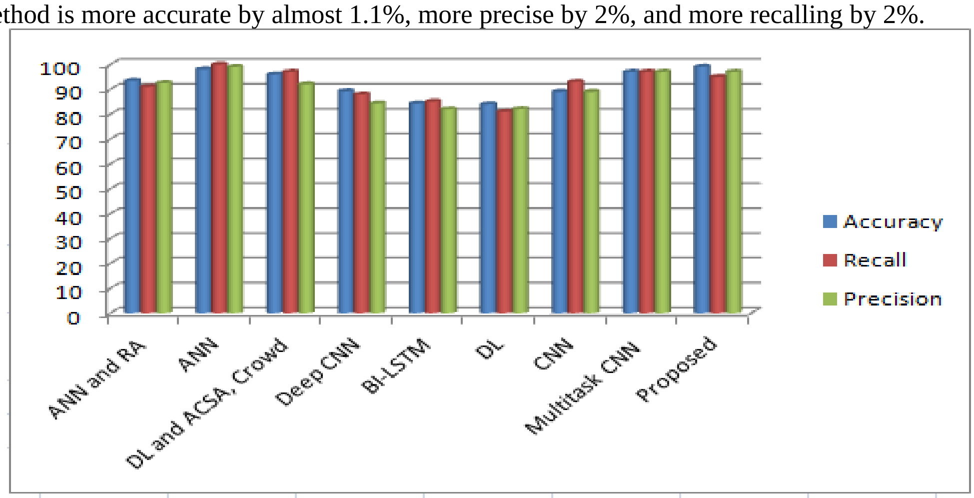


**Figure 12: Proposed System Efficiency**

## V. CONCLUSION

People with Parkinson's disease have different signs because the disease affects nerve cells. Therefore, using statistics or machine learning methods to find important traits is a time-consuming process. The model ranks parameters based on record interconnections using neural networks and boosting approaches. With the improved encoder, the number of features drops from 755 to 187, but it's still expensive to explore with the smaller number. The Extreme gradient boosting method is added to the suggested system after the network is put together so that features can be ranked by how often they make mistakes. In order to build a strong model, it changes the mistake rates as needed. Autoencoders & decoders enhance the model's ensembles by reducing their weakness. This is for auto encoders and decoders are effective in the reconstruction process. By using a good optimisation process and stacking vectors in future work, the model will be able to find the best ML methodology to act as the grouper for the last part of the network.

## REFERENCES

[1] Sahu, L., Sharma, R., Sahu, I., Das, M., Sahu, B., & Kumar, R. (2021). Efficient detection of Parkinson’s disease using deep learning techniques over medical data. Expert Systems. doi:10.1111/exsy.12787

[2] Shaban, M. (2021). Automated Screening of Parkinson’s Disease Using Deep Learning Based Electroencephalography. 2021 10th International IEEE/EMBS Conference on Neural Engineering (NER). doi:10.1109/ner49283.2021.9441065

[3] Masud, M., Singh, P., Gaba, G. S., Kaur, A., Alghamdi, R. A., Alrashoud, M., & Alqahtani, S. A. (2021). CROWD: Crow Search and Deep Learning based Feature Extractor for Classification of Parkinson's Disease. ACM Transactions on Internet Technology, 21(3), 1–18. doi:10.1145/3418500

[4] Kaur, S., Aggarwal, H., & Rani, R. (2020). Diagnosis of Parkinson's disease using deep CNN with transfer learning and data augmentation. Multimedia Tools and Applications, 80(7), 10113–10139. doi:10.1007/s11042-020-10114-1

[5] Quan, C., Ren, K., & Luo, Z. (2021). A Deep Learning Based Method for Parkinson's Disease Detection Using Dynamic Features of Speech. IEEE Access, 9, 10239–10252. doi:10.1109/access.2021.3051432

[6] Leung, K. H., Rowe, S. P., Pomper, M. G., & Du, Y. (2021). A three-stage, deep learning, ensemble approach for prognosis in patients with Parkinson's disease. EJNMMI Research, 11(1). doi:10.1186/s13550-021-00795-6

[7] Yasaka, K., Kamagata, K., Ogawa, T., Hatano, T., Takeshige-Amano, H., Ogaki, K., … Abe, O. (2021). Parkinson's disease: deep learning with a parameter-weighted structural connectome matrix for diagnosis and neural circuit disorder investigation. Neuroradiology, 63(9), 1451–1462. doi:10.1007/s00234-021-02648-4

[8] Sigcha L, Pavón I, Costa N, Costa S, Gago M, Arezes P, López JM, De Arcas G. Automatic Resting Tremor Assessment in Parkinson's Disease Using Smartwatches and Multitask Convolutional Neural Networks. Sensors. 2021; 21(1):291. https://doi.org/10.3390/s21010291

[9] Sudeepthi Govathoti, A Mallikarjuna Reddy, Deepthi Kamidi and G BalaKrishna, "Data Augmentation Techniques on Chilly Plants to Classify Healthy and Bacterial Blight Disease Leaves" International Journal of Advanced Computer Science and Applications(IJACSA), 13(6), 2022. http://dx.doi.org/10.14569/IJACSA.2022.0130618

[10] Grandhe, Padmaja. (2020). A Novel Method for Content based 3D Medical Image Retrieval System Using Dual Tree M-Band Wavelets Transform and Multiclass Support Vector Machine. Journal of Advanced Research in Dynamical and Control Systems. 12. 279-286. 10.5373/JARDCS/V12I3/20201192.

[11] Sivaranjini, S., Sujatha, C.M. Deep learning based diagnosis of Parkinson's disease using convolutional neural network. Multimed Tools Appl 79, 15467–15479 (2020). https://doi.org/10.1007/s11042-019-7469-8

[12] Oh, S.L., Hagiwara, Y., Raghavendra, U. et al. A deep learning approach for Parkinson's disease diagnosis from EEG signals. Neural Comput & Applic 32, 10927–10933 (2020). https://doi.org/10.1007/s00521-018-3689-5

[13] Masud, M., Singh, P., Gaba, G. S., Kaur, A., Alghamdi, R. A., Alrashoud, M., & Alqahtani, S. A. (2021). CROWD: Crow Search and Deep Learning based Feature Extractor for Classification of Parkinson's Disease. In ACM Transactions on Internet Technology (Vol. 21, Issue 3, pp. 1–18). Association for Computing Machinery (ACM). https://doi.org/10.1145/3418500

[14] Kaur, S., Aggarwal, H. & Rani, R. Hyper-parameter optimization of deep learning model for prediction of Parkinson's disease. Machine Vision and Applications 31, 32 (2020). https://doi.org/10.1007/s00138-020-01078-1

[15] Wingate, J., Kollia, I., Bidaut, L., & Kollias, S. (2020). Unified deep learning approach for prediction of Parkinson's disease. In IET Image Processing (Vol. 14, Issue 10, pp. 1980–1989). Institution of Engineering and Technology (IET). https://doi.org/10.1049/iet-ipr.2019.1526

[16] Magesh, P. R., Myloth, R. D., & Tom, R. J. (2020). An Explainable Machine Learning Model for Early Detection of Parkinson's Disease using LIME on DaTSCAN Imagery. In Computers in Biology and Medicine (Vol. 126, p. 104041). Elsevier BV. https://doi.org/10.1016/j.compbiomed.2020.104041

[17] P. Khan *et al.*, "Machine Learning and Deep Learning Approaches for Brain Disease Diagnosis: Principles and Recent Advances," in *IEEE Access*, vol. 9, pp. 37622-37655, 2021, doi: 10.1109/ACCESS.2021.3062484.

[18] Karapinar Senturk, Z. (2020). Early diagnosis of Parkinson's disease using machine learning algorithms. In Medical Hypotheses (Vol. 138, p. 109603). Elsevier BV. https://doi.org/10.1016/j.mehy.2020.109603

[19] Jain, D., Mishra, A. K., & Das, S. K. (2020). Machine Learning Based Automatic Prediction of Parkinson's Disease Using Speech Features. In Proceedings of International Conference on Artificial Intelligence and Applications (pp. 351–362). Springer Singapore. https://doi.org/10.1007/978-981-15-4992-2_33

[20] Ul Haq, A., Li, J., Memon, M. H., khan, J., Ali, Z., Abbas, S. Z., & Nazir, S. (2020). Recognition of the parkinson's disease using a hybrid feature selection approach. In Journal of Intelligent & Fuzzy Systems (Vol. 39, Issue 1, pp. 1319–1339). IOS Press. https://doi.org/10.3233/jifs-200075