JOURNAL OF
EMERGING INVESTIGATORS

Article

Machine Learning Algorithm Using Logistic Regression and an Artificial Neural Network (ANN) for Early Stage Detection of Parkinson's Disease

Shreyas Kar[1], Peter W. Campbell[2]
[1] duPont Manual High School, Louisville, Kentucky, 40208, USA.
[2] W University of Louisville, Department of Anatomical Sciences and Neurobiology, Louisville, Kentucky, 40202,

SUMMARY

Parkinson's disease (PD) is the second most common neurodegenerative disease. Despite the prevalence of PD, diagnosing PD is expensive, requires specialized testing, and is often inaccurate. Moreover, diagnosis is often made late in the disease course when treatments are less effective. Since one of the earliest symptoms of PD are changes in voice patterns, we employed machine learning algorithms to detect these abnormalities. Using existing voice data from patients with PD and healthy controls, we created and trained two different algorithms: one using logistic regression and another employing an artificial neural network (ANN). The inputs for these algorithms were two statistical measures of voice patterns: Pitch Period Entropy (PPE) and Spread1. Both algorithms were successfully able to discriminate between PD patients and healthy controls with F2 scores > 0.93. Moreover, we found that the time from diagnosis had no impact on the performance of our models. Thus, we report the creation of two models that can reliably and accurately identify the voice patterns characteristic of PD. Our findings suggest that it is possible to diagnose PD by analyzing voice patterns, which would enable disease screening that is cheap, accessible, and accurate.

INTRODUCTION

Parkinson's disease (PD) is a neurodegenerative disease that is classically considered a motor disease but can impact the sensory and cognitive systems as well [1–7]. PD occurs when neurons in the brain die either via apoptosis (programmed cell death) or necrosis (unplanned cell death), which causes a decrease in dopamine, a neurotransmitter that is responsible for sending signals to other nerve cells, eventually leading to PD [8,9]. A number of factors contribute to the disease process including genetic predisposition, abnormal protein folding, oxidative stress, inflammation, and immune dysregulation [8,9]. Motor symptoms, especially tremors, usually present unilaterally with the left side involved more frequently than the right, but can spread bilaterally [10]. These symptoms arise because PD occurs due to a progressive degeneration of dopaminergic neurons [1,2,4–6,11]. Today the standard therapy is dopamine replacement therapy by administering the dopamine precursor, levodopa [3–5,12–14]. However, because PD is often detected at later stages when over 80% of dopamine neurons have died, current therapies, including levodopa, are often ineffective [7,14]. Dopamine neurons are located in the substantia nigra, a brain region which is involved in coordinating and planning body movement [1,6]. A reduction in dopamine from these neurons produces several motor symptoms, one of the earliest of which is difficulty in speech [2,11,15–17]. Voice symptoms are attributed to involvement of the vagus nerve and recurrent laryngeal nerve, which occurs either by pathological inclusions or by abnormal excitatory drive from the basal ganglia [18]. The voice of PD patients gets softer, breathy, slurred, and often mumbled. The tone of the voice becomes monotone without any inflection [7,17,19–23]. They also exhibit altered voice quality (dysphonia), a reduced range of articulation (hypokinetic articulation), and an irregular and rapid rate of speech (tachyphemia) [19]. In comparison to changes in brain imaging, PD associated voice changes are easier and cheaper to assess and occur earlier in the disease process. We therefore sought to create a method for detecting PD associated voice changes, which may be used to reliably, cheaply, and accurately diagnose PD. To diagnose PD using voice patterns, we took a machine learning approach that employed binary classification to assign inputs to one of two states: likely to have PD and unlikely to have PD. This classification can be accomplished by different types of algorithms[21,24–28], and we used two in this study: logistic regression and artificial neural networks. Logistic regression is a method of classification that finds the probability of a certain event occurring (in this case the probability of PD) as the output of the function [29]. An artificial neural network (ANN) is an algorithm, inspired by neuronal connectivity in the central nervous system, that takes in an input in the first layer, performs computations on the weighted inputs, applies activation functions in the hidden layer, and outputs a desired result in the final layer [30] . The connections between nodes and layers generates pattern recognition capabilities similar to those generated by neurons and synapses in animal brains.While ANNs are more powerful than logistic regression, they are also more costly in terms of computational demands [31]. Therefore, we used both methods to determine if there was a significant difference between the two approaches in this task. Others have taken similar approaches to use machine learning to assess PD [7,21–23,32,33]. However, these studies have relied on symptoms of PD that appear later in the course of the disease, such as gait disturbances and

tremor. While there have been some studies done on the early detection of PD, most have used algorithms such as support vector machines and data mining, not artificial neural networks [7,21,24–28]. As far as we are aware, this is the first attempt to use an artificial neural network to identify changes in voice characteristics to assist in the diagnosis of PD.

DISCUSSION

Here we report the development of two algorithms that were successfully able to discriminate the voice patterns of PD patients from healthy controls. Both the logistic regression and the artificial neural network (ANN) surpassed our success criteria (F2 > 0.87), and our analysis of the cost function indicates that the logistic regression was more accurate than the ANN. Moreover, the logistic regression had a specificity of 0.70 and a sensitivity of 0.95. Thus, while both algorithms were successful in identifying the voice patterns of PD patients, the ANN was more sensitive and had a higher F2 score.

We also investigated whether the years since diagnosis of PD impacted the function of our logistic regression. We found that the number of false positives and false negatives were similar to our original analysis yet the model poorly discriminated between early and late PD. The model performed similarly in discriminating normal from either early or late PD. This suggests that disease progression did not impact the performance of either model. However, due to how the data were obtained, we could not control for how long patients were in treatment or whether their treatments impacted our model [20]. Since our algorithms are intended to diagnose PD and not necessarily to be used to follow disease progression, the impact of levodopa, other pharmacotherapies, deep brain stimulation, or other therapies are beyond the scope of this study.

Our results support reports from others that machine learning is a promising approach to diagnose PD [7,21–28]. However, as far as we are aware, this is the first use of an ANN to examine voice patterns to diagnose PD. Some researchers have used advanced algorithms to analyze non-vocal data [31,32,39], and our success rate is similar. Others have examined vocal patterns using more traditional analyses [15,22,23,41], but logistic regressions and ANNs are generally preferable because of their ability for automated diagnosis, saving patients time and resources in early diagnosis. Thus, our data extend the field by employing machine learning algorithms to detect changes in vocal patterns.

This study demonstrates a relatively fast and inexpensive method for early stage detection of PD. Such a method can benefit patients as it may enable accurate diagnosis of PD at an earlier stage without expensive imaging [3,4,13,14,16,40]. Earlier diagnosis of PD may allow reevaluation of and improve the efficacy of therapies that have previously failed clinical trials due to use in late stage PD patients [11]. Furthermore, this method of detection can be administered remotely by using voice samples making it suitable for anybody around the world. This can be used as part of regular health checkup for populations susceptible to PD as this is both affordable and non-invasive. Once detected, a patient can then go through detailed clinical and physical diagnosis and start early treatment to reduce loss of valuable dopaminergic neurons.

MATERIALS AND METHODS

Data acquisition

Two voice parameters, Pitch Period Entropy and Spread 1, were used as inputs to the algorithms in this study. These data were obtained using the Machine Learning Repository of University of California Irvine [20]. This dataset is composed of voice samples of 195 voice recordings from 31 people. The voice recordings are comprised of 19 male voices and 12 female voices. The age varies from 46 to 81 with an average age of 66. Out of 31 people, 23 have PD. The number of years since diagnosis with PD varies from 0 to 28 among the PD patients.

Machine Learning Algorithms

Two major machine learning algorithms are used in this study: Logistic Regression and Artificial Neural network (ANN). A programing language called octave, similar to MATLAB, was used to implement these algorithms. Both algorithms were trained and tested using the acquired dataset. 60% of the data was randomly selected for training while the remaining 40% was reserved for testing. Spread 1 and PPE data were loaded into a matrix, termed X, and the status data was loaded into a column vector called Y.

Logistic Regression

The logistic regression was trained on 60% of the data using the flow chart illustrated in Figure 1. The cost function was implemented by using a vectorized version of the cost function for Logistic Regression. Next, in order to use an optimization algorithm, the partial derivatives with respect to each of the features was computed. The partial derivatives were stored in a column vector. After implementing the cost function and the gradients, the *fminunc* algorithm (finding minimum of unconstrained multivariable function) was trained over 100 epochs to find the optimal coefficient values. This algorithm numerically optimizes a multivariable function , in this case, the cost function of logistic regression. Optimal coefficient values (stored in a matrix called θ) are listed in the result section. The data were then normalized using two different techniques: z-score normalization and simple feature scaling normalization. After normalization of the training data, the algorithm was trained again. After logistic regression was implemented, the model was used to classify the test dataset. Prediction was performed by rounding the probability values. The decision boundary that separates the two classes is termed the hypothesis, which is determined by multivariate linear regression $y = \sum_{i=1}^{n} b_i x_i$ also written as $y = b^T x$).

Because probability values are always between 0 and 1, the output given by the hypothesis in multivariate linear regression needs to be constrained to output a value between 0 and 1. For this, the equation for multivariate linear regression is plugged into the logistic/sigmoid function, which is represented as:

$$g(z) = \frac{1}{1+e^{-z}}$$

If the value of $g(b^Tx)$ is greater than or equal 0.5 then the binary classification will output a value of 1, representing a high likelihood to get PD. However, if the value of the binary classification is less than 0.5 then the output will be 0, representing a low likelihood to get PD. The coefficients of the hypothesis are the optimal values such that the cost function is minimized. The cost function for logistic regression [34] is

$$J(\theta) = -\frac{1}{m}\sum_{i=1}^{m}[y^{(i)}\ln(h_\theta(x^{(i)})) + (1-y^{(i)})\ln(1-h_\theta(x^{(i)}))]$$

Where m represents the amount of training examples, y, in this case, represents the status of the Patient (0 for not having PD, and 1 for having PD). $h_\theta(x)$ represents the sigmoid function evaluated at a particular x value.

Artificial Neural Network (ANN)

Figure 1: Logistic Regression Approach

The cost function for the ANN was implemented using a *for* loop (representing the double summation) over all training examples classes (having PD and not having PD).

$$J(\theta) = -\frac{1}{m}\sum_{i=1}^{m}\sum_{k=1}^{K}[y_k^{(i)}\ln((h_\theta(x^{(i)}))_k) + (1-y_k^{(i)})\ln(1-(h_\theta(x^{(i)}))_k) + \frac{\lambda}{2m}\sum_{l=1}^{L-1}\sum_{i=1}^{s_l}\sum_{j=1}^{s_l+1}(\theta_{j,i}^{(l)})^2$$

Taking this sum and plugging it into the sigmoid function produced the value of each activation in the network. $z^{(n)}$ represented the activation values of the n^{th} layer before plugging into the sigmoid function, while $a^{(n)}$ represented the actual (post-sigmoid) activations of the n^{th} layer.

After the cost function was implemented, the backpropagation algorithm was implemented to compute the partial derivative with respect to each of the weights (θ_{ij}). This was done by first feedforwarding the neural network using a random set of weights (this was done for symmetry breaking). Then for each training example, an error value was computed for each node and accumulated in the *del one* and *del two* matrices. Finally, separate gradient vectors for $\theta(1)$and $\theta(2)$ were computed which were "unrolled" in a vector of partial derivatives, which was used in the optimization process along with the cost function.

As with logistic regression, the fminunc algorithm (with the gradient vector and cost function as inputs) was used to minimize the cost function and train the algorithm to compute the optimal values of the weights. The gradient vector was a column vector with the partial derivatives with respect to all the weight. It was calculated using a backpropagation algorithm and checked using Gradient checking (an algorithm that computes the partial derivatives numerically).

First, we performed forward propagation to determine the values of all the activations in the network. Next, all the "error" values, represented as δ, were calculated as

$$\delta = (\theta^{(m)})^T\delta^{m+1}.g'(z^{(m)})$$

except for the last output layer. That layer was determined as the value of the last activation layer subtracted by the value of the actual output.

Next, an accumulation term, Δ, was used to accumulate each of the errors. After this, it can be shown that $\frac{\partial}{\partial\theta_{ij}} = \frac{1}{m}\Delta_{ij}^{(l)}$ for the bias terms, and $\frac{\partial}{\partial\theta_{ij}} = \frac{1}{m}\Delta_{ij}^{(l)}\lambda\theta_{ij}^{(l)}$ for all other terms. While there does exist a simpler formula for $\frac{\partial}{\partial\theta_{ij}}$ when the regularization term is not accounted for (λ= 0) , this term helps prevent overfitting – a phenomenon which occurs when the model performes satisfactory with training data, but fails to generalize to test data – [35], which is of paramount importance in a neural network. This produced the gradient vector that was used as inputs to our optimization algorithm. After this the predict function was implemented by computing both activation values for each training example, and the largest one was the output.

Evaluation of Models

The predicted results are laid down against ground truth to evaluate each model so that True Positive (TP), True Negative (TN), False Positive (FP) and False Negative (FN) values are calculated **(Table 1)**. The general formula for F score is

$$F_\beta = (1+\beta^2) * \frac{precision * recall}{(\beta^2 * precision) + recall}$$

where precision is a function of TP and FP: precision = TP/(TP+FP); recall is a function of TP and FN: recall = TP/(TP+FN); and specificity is a function of TN and FP: specificity = TN/(TN+FP).

The algorithms were assessed based on their F2 scores. An F2 score is a weighted average of precision and recall and is used when minimizing the number of false negative cases is of greater importance than minimizing the number of false positive cases. The equation for F2 scores is as follows:

$$F_2 = 5 * \frac{precision * recall}{4 * precision + recall}$$

Algorithms were considered successful if F2 is greater than 0.87. This value was chosen because this algorithm is one of high recall, used when minimizing the amount of false-negative cases are of greater importance than minimizing the amount of false-positive cases [36–38]. Moreover, this value has been used by other researchers to evaluate model success [39].

Differentiating between PD stages

The same models were created for PD stage differentiation. The data set was split into 2 stages based on the 4-year mark. This specific benchmark was used as it is a common benchmark for early and late stage diagnosis

[40] . The split data set was evaluated based upon the earlier mentioned F2 score metric.

True Positive (TP)	Diagnosed PD vs Actual PD
False Positive (FP)	Diagnosed PD vs Healthy
True Negative (TN)	Diagnosed Healthy vs Actual Healthy
False Negative (FN)	Diagnosed Healthy vs Actual PD

Table 1: Describes the attribute of the confusion matrix

Normalization

Normalization is the process of converting data into a particular standard scale. This is done so that a data sample of a bigger range is not given additional preference solely due to its size. There exist three most commons forms of data normalization and standardization: Simple Feature scaling, z-scores and min-max normalization. Simple Feature scaling consist of converting the data set into values from 0 to 1. This is done of dividing each value in a column of a data set by the maximum value of the data set. If all the values are either strictly positive or negative then each example can be divided by the largest and smallest value respectively in the data set. *Z*-score normalization (more formally written as *z*-score standardization) is a method of normalization wherein each training example is converted into its respective *z*-score. This in turn converts the data set into one with a mean of 0 and a standard deviation of 1. This is done without changing the original shape distribution of the data set. The formula to calculate the z-score for a training example is given by the following equation:

Where z_i is the transformed value of the training example

$$z_i = \frac{x_i - \mu}{\sigma}$$

μ. epresents the mean of the data set and α represents the standard deviation of the data set.

Results

We constructed two different models to diagnose PD using two measures of voice patterns: Spread1 and PPE. These data were obtained from the Machine Learning Repository of University of California Irvine [20]. Both algorithms were trained on a random sample of 60% of the dataset and tested on the remaining 40%.

Logistic Regression

The first algorithm we trained was a logistic regression. We first normalized the data by converting raw values into respective z-scores and then dividing by the largest absolute value. This algorithm also uses three coefficient values that are stored in a matrix called Theta **(Table 2)**. Theta (1) has the constant value, Theta (2) has the coefficient values for Spread1 and Theta (3) has coefficient values for PPE. Using these values our logistic regression failed to pass the 0.87 threshold on the training data.

Therefore, we applied feature normalization to put Spread1 and PPE values in the same scale. This was done in two ways. Firstly, this was done by converting each number in a data set to its respective *z*-score , thus, normalizing the data set. In the second type of normalization, division, both data sets were divided by the highest absolute valued number in their respective data set. Then the absolute values of all numbers were taken after division. When the model was trained with the above normalized values, a different matrix for Theta was found **(Table 2)**.

	Initial Values	Z-Score Normalized	Divided Normalized
Theta 1	11.0235	2.43125	16.45383
Theta 2	1.7788	2.60668	16.97537
Theta 3	7.2889	-0.13596	-0.49524

Table 2: Theta Matrix for Logistic Regression

After these measures, prediction was performed with the test data and the algorithm achieved a F2 score of 0.93 **(Table 3).** Within the 78 voice patterns tested, the logistic regression achieved a specificity of 0.70 and a sensitivity of 0.95. The calculated F2 score of 0.93 was well above our threshold of 0.87 and indicates that this algorithm could successfully discriminate PD voice patterns. This is evidenced by our receiver-operator curve **(ROC; Figure 3),** which shows high specificity over various threshold values.

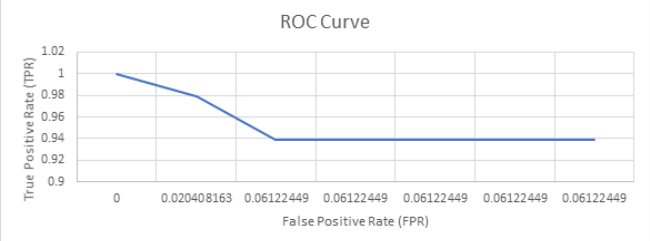

Figure 3: ROC Curve for Logistic Regression

Artificial Neural Network (ANN)

Initially, we implemented an ANN with one hidden layer. The ANN had two controllable matrices of weights (Theta), which were optimized using the fminunc algorithm and randomly initialized for symmetry breaking. After training, the model produced optimized weights **(Figure 2)**. The result of ANN with one hidden layer is in table 3.

The ANN with one hidden layer did not achieve the success criteria **(Table 3)**. To make the model more accurate and robust, we added an additional hidden layer. This layer was comprised of 4 activation units and a bias unit, and forward and backward propagation were performed to train the algorithm producing different optimized weights **(Figure 4)**.

Using the ANN with two hidden layers, our model passed the success criteria on the test data achieving an F2 score of 0.94 (table 3). The sensitivity of this model was 0.96,

the specificity was 0.61 using 78 voice patterns. The ROC curve **(Figure 5)** demonstrates high specificity over various threshold values.

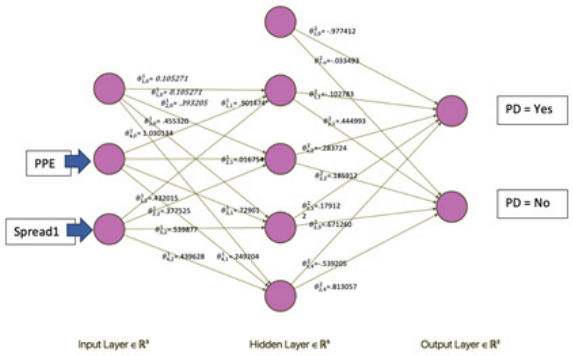

Figure 2: Weights and activations of ANN with single hidden layer

	Logistic Regression	ANN 1-Layer	ANN 2-Layers
Total Test Data	78	78	78
TP	54	53	53
FP	7	2	9
TN	14	12	14
FN	3	11	2
Precision	0.89	0.96	0.85
Recall/Sensitivity	0.95	0.83	0.96
Specificity	0.70	0.86	0.61
F1 Score	0.92	0.89	0.91
F2 Score	0.93	0.85	0.94

Table 3: Summary Data from Models

Figure 4: Weights and activations of ANN with two hidden layers

Differentiating between PD stages

PD is a neurodegenerative disease with multiple stages [2,4,5,7,12,14,16,36]. Therefore, we wanted to determine if the length of PD diagnosis had an impact on our model's performance. Patients with PD typically exhibit a response to L-DOPA therapy, as well as L-DOPA induced dyskinesias, within four years of diagnosis [40]. Moreover, the earliest that patients reached stage three of the disease was at 4 years [40], suggesting that a 4-year cutoff would differentiate early PD from late PD. Additionally, there are other neurodegenerative diseases, that while similar to PD in presentation, exhibit poor responses to L-DOPA (e.g. progressive supranuclear palsy, corticobasal degeneration) that are typically identified by this point in time [11]. Therefore, we divided patients into groups who had a PD diagnosis for less than 4 years or more than 4 years to determine if our results were confounded by disease stage.

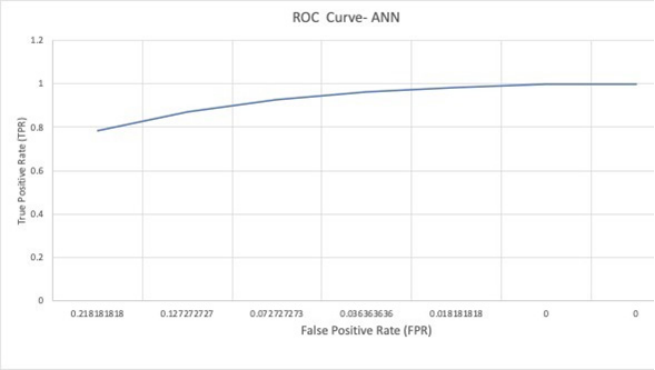

Figure 5: ROC Curve of ANN with two hidden layers.

We examined whether the logistic regression (used above) would perform differently on this divided dataset. The amount of true positive and false positive cases in the result for the test case when the algorithm was run was approximately the same. When the Logistic Regression was run to distinguish between the 65 PD patients who are below 4 years and above 4 years since diagnosis, with 49.2 % true positive, 47.7% false positive, 1.5% true negative and 1.5 %false negative were observed. This suggests that PPE and Spread1 are robust indicators and consistent indicators of PD and that our models were not impacted by the stage of PD. However, because there was little variability in these voice metrics, the logistic regression classifier performed poorly in identifying early PD from late PD (approximately 50% training accuracy, in this case). In other words, PPE and Spread1 values do not change significantly over time. Thus, this lack of variability means that the earlier stated algorithms can be used for detection at early stages.

Received: January 6, 2020
Acccepted: September 13, 2020
Published: October 10, 2020

REFERENCES

1. Wichmann, Thomas, and Mahlon R. DeLong. "Neurotransmitters and Disorders of the Basal Ganglia." Basic *Neurochemistry*, 2012, doi:10.1016/B978-0-12-374947-5.00049-3.
2. Kalia, Lorraine V., and Anthony E. Lang. "Parkinson's Disease." *The Lancet*, 2015, doi:10.1016/S0140-

6736(14)61393-3.
3. Chaudhuri, K. Ray, et al. "Non-Motor Symptoms of Parkinson's Disease: Diagnosis and Management." *Lancet* Neurology, 2006, doi:10.1016/S1474-4422(06)70373-8.
4. Pedrosa, David J., and Lars Timmermann. "Review: Management of Parkinson's Disease." *Neuropsychiatric Disease and Treatment,* 2013, doi:10.2147/NDT.S32302.
5. Olanow, C. Warren. "Levodopa: Effect on Cell Death and the Natural History of Parkinson's Disease." *Movement Disorders,* 2015, doi:10.1002/mds.26119.
6. Sonne, James, and Morris R. Beato. "Neuroanatomy, Substantia Nigra." *StatPearls,* 2018.
7. Bhat, Shreya, et al. "Parkinson's Disease: Cause Factors, Measurable Indicators, and Early Diagnosis." *Computers in Biology and Medicine*, 2018, doi:10.1016/j.compbiomed.2018.09.008.
8. Venderova, Katerina, and David S. Park. "Programmed Cell Death in Parkinson's Disease." *Cold Spring Harbor Perspectives in Medicine,* 2012, doi:10.1101/cshperspect.a009365.
9. Farooqui, Tahira, and Akhlaq A. Farooqui. "Lipid-Mediated Oxidative Stress and Inflammation in the Pathogenesis of Parkinson's Disease." *Parkinson's Disease,* 2011, doi:10.4061/2011/247467.
10. Heinrichs-Graham, Elizabeth, et al. "The Cortical Signature of Symptom Laterality in Parkinson's Disease." *NeuroImage*: Clinical, 2017, doi:10.1016/j.nicl.2017.02.010.
11. Ropper, Allan H., and Robert H. Brown. "Abnormalities of Movement and Posture Due to Disease of the Basal Ganglia." *Adams and Victor's Principles of Neurology,* 2005, doi:10.1036/0071469710.
12. Hornykiewicz, Oleh. "A Brief History of Levodopa." *Journal of Neurology*, 2010, doi:10.1007/s00415-010-5741-y.
13. Connolly, Barbara S., and Anthony E. Lang. "Pharmacological Treatment of Parkinson Disease: A Review." *JAMA - Journal of the American Medical Association,* 2014, doi:10.1001/jama.2014.3654.
14. Goldenberg, Marvin M. "Medical Management of Parkinson's Disease." *P and T,* 2008.
15. Trancoso, Isabel, et al. "Analysing Speech for Clinical Applications." *Lecture Notes in Computer Science* (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 2018, doi:10.1007/978-3-030-00810-9_1.
16. Nutt, John G., and G. Frederick Wooten. "Diagnosis and Initial Management of Parkinson's Disease." *New England Journal of Medicine*, 2005, doi:10.1056/NEJMcp043908.
17. Goberman, Alexander M., and Michael Blomgren. "Parkinsonian Speech Disfluencies: Effects of L-Dopa-Related Fluctuations." *Journal of Fluency Disorders,* 2003, doi:10.1016/S0094-730X(03)00005-6.
18. Erman, Audrey B., et al. "Disorders of Cranial Nerves IX and X." *Seminars in Neurology,* 2009, doi:10.1055/s-0028-1124027.
19. Cannito, Michael P., et al. "Vocal Aging and Adductor Spasmodic Dysphonia: Response to Botulinum Toxin Injection." *Clinical Interventions in Aging,* 2008, doi:10.2147/cia.s1416.
20. Little, Max, et al. "Exploiting Nonlinear Recurrence and Fractal Scaling Properties for Voice Disorder Detection." *Nature Precedings*, 2007, doi:10.1038/npre.2007.326.1.
21.Belić, M., et al. "Artificial Intelligence for Assisting Diagnostics and Assessment of Parkinson's Disease—A Review." *Clinical Neurology and Neurosurgery,* 2019, doi:10.1016/j.clineuro.2019.105442.
22. Erdogdu Sakar, Betul, et al. "Analyzing the Effectiveness of Vocal Features in Early Telediagnosis of Parkinson's Disease." *PloS One*, 2017, doi:10.1371/journal.pone.0182428.
23. Lahmiri, Salim, et al. "Performance of Machine Learning Methods in Diagnosing Parkinson's Disease Based on Dysphonia Measures." *Biomedical Engineering Letters*, 2018, doi:10.1007/s13534-017-0051-2.
24.Dinov, Ivo D., et al. "Predictive Big Data Analytics: A Study of Parkinson's Disease Using Large, Complex, Heterogeneous, Incongruent, Multi-Source and Incomplete Observations." *PLoS ONE*, 2016, doi:10.1371/journal.pone.0157077.
25. Adams, Warwick R. "High-Accuracy Detection of Early Parkinson's Disease Using Multiple Characteristics of Finger Movement While Typing." *PLoS ONE,* 2017, doi:10.1371/journal.pone.0188226.
26. Bratić, Brankica, et al. "Machine Learning for Predicting Cognitive Diseases: Methods, Data Sources and Risk Factors." *Journal of Medical Systems*, 2018, doi:10.1007/s10916-018-1071-x.
27. Rovini, Erika, et al. "Comparative Motor Pre-Clinical Assessment in Parkinson's Disease Using Supervised Machine Learning Approaches." *Annals of Biomedical Engineering*, 2018, doi:10.1007/s10439-018-2104-9.
28. Cavallo, Filippo, et al. "Upper Limb Motor Pre-Clinical Assessment in Parkinson's Disease Using Machine Learning." *Parkinsonism and Related Disorders*, 2019, doi:10.1016/j.parkreldis.2019.02.028.
29. Bhattacharyya, Saptashwa. "'Logit' of Logistic Regression; Understanding the Fundamentals." *Medium - Towards Data Science*, 2018, towardsdatascience.com/logit-of-logistic-regression-understanding-the-fundamentals-f384152a33d1.
30. Chauhan, Nagesh Singh. "Introduction to Artificial Neural Networks (ANN)." *Medium - Towards Data Science,* 2019, towardsdatascience.com/introduction-to-artificial-neural-networks-ann-1aea15775ef9.
31. Parsaeian, M., et al. "Comparison of Logistic Regression and Artificial Neural Network in Low Back Pain Prediction: Second National Health Survey." *Iranian Journal of Public Health*, 2012.
32. Tsoulos, Ioannis G., et al. "Application of Machine Learning in a Parkinson's Disease Digital Biomarker Dataset Using Neural Network Construction (NNC) Methodology Discriminates Patient Motor Status." *Frontiers in ICT*, 2019, doi:10.3389/fict.2019.00010.
33. Tucker, Conrad, et al. "A Data Mining Methodology for Predicting Early Stage Parkinson's Disease Using

Non-Invasive, High-Dimensional Gait Sensor Data." *IIE Transactions on Healthcare Systems Engineering,* 2015, doi:10.1080/19488300.2015.1095256.
34. Ng, Andrew. "Machine Learning by Stanford University." *Coursera,* 2012, www.coursera.org/learn/machine-learning.
35. Nagpal, Anuja. "Over-Fitting and Regularization." *Medium - Towards Data Science,* 2017, https://towardsdatascience.com/over-fitting-and-regularization-64d16100f45c.
36. Powers, David M. W. "Evaluation: From Precision, Recall and F-Measure to ROC, Informedness, Markedness & Correlation." *Journal of Machine Learning Technology*, 2011.
37. Sasaki, Yutaka. "The Truth of the F-Measure." *Teach Tutor Mater*, May 2007.
38. Bzdok, Danilo, et al. "Statistics versus Machine Learning." *Nature Methods*, 2018, doi:10.1038/nmeth.4642.
39. Zhang, Kevin, and Dina Demner-Fushman. "Automated Classification of Eligibility Criteria in Clinical Trials to Facilitate Patient-Trial Matching for Specific Patient Populations." *Journal of the American Medical Informatics Association*, 2017, doi:10.1093/jamia/ocw176.
40.Poewe, Werner. "The Natural History of Parkinson's Disease." *Journal of Neurology*, 2006, doi:10.1007/s00415-006-7002-7.
41. Schwab, Patrick, and Walter Karlen. "PhoneMD: Learning to Diagnose Parkinson's Disease from Smartphone Data." *Proceedings of the AAAI Conference on Artificial Intelligence,* 2019, doi:10.1609/aaai.v33i01.33011118.