

Speech-Based Ensemble Method For Detecting Parkinson's Disease

Shah Parveez Ahmed^{1,*}, Dr. Siddhartha Baruah²

¹Research Scholar, Assam Science and Technology University, Guwahati, India

²Professor, Dept. of Computer Application, Jorhat Engineering College, Jorhat, India

Corresponding author: Shah Parveez Ahmed

Abstract

Parkinson's disease (PD) is ranked next to the Alzheimer as the most common neurodegenerative disease. PD is a progressive disease which is incurable as yet. Early detection of PD is vital to slow down its progress. About 90% of the PD patients suffer from speech impairment. Speech is a crucial biomarker for the early detection of PD. Speech signals can be effectively utilized in machine learning to predict the disease in non-invasive and cost-effective way. In several studies of machine learning, effort has been made to predict PD. But, in most of the studies issues like data preprocessing, data imbalance, and feature selection are often overlooked. In this study, different ensemble techniques like Random Forest, Bagging with various base estimators, GradientBoosting, XGBoost, AdaBoost, Stacking and Voting are applied to speech features. Class imbalance in a dataset causes machine learning algorithms to be biased to the majority class, leading to the misclassification problem. This issue is addressed by applying data balancing techniques. The different approaches to data balancing have adopted different algorithms for producing a balanced dataset. As a result, different data balancing techniques have different impacts on the machine learning algorithms with the same dataset, producing varying classification accuracy. Hence, different data balancing techniques are explored in this study to obtain 98.31% of accuracy. ANOVA F-test is performed for feature selection. 118 machine learning models are tested. Model performance is measured in terms of accuracy, sensitivity, specificity, F1-score, execution Time and ROC curve.

Keywords: Ensemble Learning, Machine Learning, Parkinson's Disease, Speech Impairment, Speech Signals.

1. INTRODUCTION

Parkinson's disease (PD) is a neurodegenerative disease caused by the death of dopamine producing neurons in the substantia nigra of the brain [1][2]. Dopamine is a chemical which carries signals related to the movement control of our body. Bradykinesia, rigidity, tremor and postural instability are the major motor symptoms that PD patients suffer from [2]. Speech impairment is a motor symptom that may appear five years before the medical diagnosis of PD [3][4]. Speech impairment is one of the early and major motor symptoms of PD [4], [5] since it manifests in approximately 90% of the patients [5]. There are non-motor symptoms in the form of constipation, urinary and sexual dysfunction, gastrointestinal dysfunction and cardiovascular dysfunction [6], olfactory deficit, REM (Rapid eye movement) sleep behavior disorder, depression, fatigue, nausea and pain [7], all together makes the life of the patient extremely painful. According to the World Health Organization [8], over 8.5 million people with PD were estimated globally in 2019 causing 329000 to die, a rise of over 100% deaths since 2000. Yet, there is no cure for PD. Hence early detection of PD may help slow down or prevent the progression of the disease.

Increasing applications of ML in healthcare are riding on its popularity due to its capability of pattern recognition, predictability and higher accuracy for assisting medical experts in disease diagnosis.

Globally several researches have been conducted to detect PD by using different biomarkers such as SPECT images [9], MRI data [10], functional MRI [11], gait measurements [12], typing [13] and speech [14]. Previous researches encountered problems of imbalanced datasets and did not seem to cope with it properly. Imbalanced datasets affect adversely and produce biased and inaccurate results. By addressing the issues of imbalanced dataset, data preprocessing and feature selection, in this study different ensemble techniques are explored using speech signals to develop a hybrid model that predicts PD accurately and efficiently.

2. RELATED WORKS

H. Ma et al. [25] used both Recursive Feature Elimination (RFE) and Principal Component Analysis (PCA) for dimensionality reduction with Support Vector Machine (SVM) as a classifier. SVM-RFE was found better than SVM with PCA with an accuracy of 96.29%. Z. A. Bak et al. [26] trained two Artificial Neural Network (ANN) based algorithms, viz., Levenberg-Marquardt (LM) and Scaled Conjugate Gradient (SCG) to find LM outdoing SCG with an accuracy of 90%. They used ANOVA for feature selection. A. U. Haq et al. [27] performed a comparative analysis of ML classifiers and deep neural network (DNN) by using performance evaluation metrics such as accuracy, sensitivity, specificity and Matthews's correlation coefficient. They found DNN to be outperforming other ML classifiers with an accuracy of 98%. M. S. Islam et al. [28] conducted comparative analysis of heterogeneous classifiers including Random Tree (RT), SVM and Feedforward Back-propagation based Artificial Neural Network (FBANN). To evaluate model's performance, they used 10-fold cross validation. Their proposed model achieved 97.37% accuracy. A. U. Haq et al. [29] proposed an SVM based PD prediction system. L1-norm SVM was used for feature selection and K-fold cross-validation was performed for performance evaluation. K. Shyamala et al. [19] proposed Interpretable Feature Ranking XGBoost (IFRX) model to classify PD that achieved an accuracy of 96.61%. They used eXplainable Artificial Intelligence (XAI) like SHAP that explains the inner functioning of a machine learning model such as ranking the features according to their relevance in detecting PD. S. N. H. Bukhari et al. [23] proposed a model based on the AdaBoost classifier with an accuracy of 96%. They used SMOTE for data balancing and feature selection was made by using PCA.

3. METHODOLOGY

In the process of building a machine learning model for the prediction of PD, data exploratory analysis is followed by the subsequent steps of data preparation, feature selection, data balancing, data splitting, model training, model testing and evaluation are performed. Figure1 outlines the entire process.

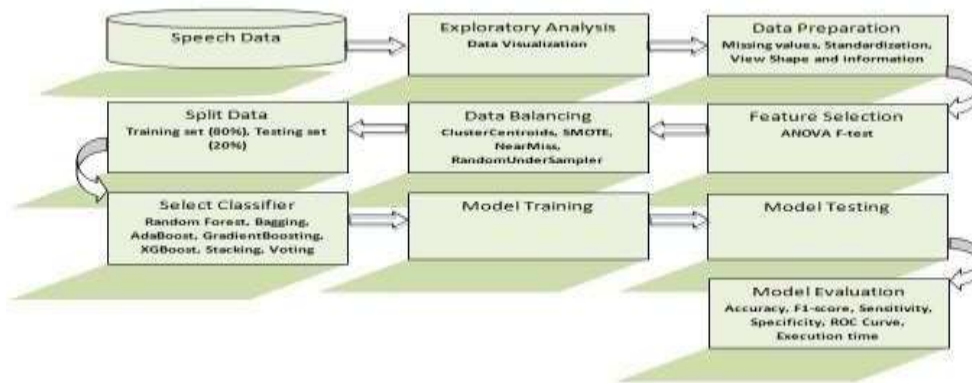


Figure1. Subsequent steps of the model building process of this study

3.1. Dataset

Data for this research is collected from UCI machine learning repository [15], [16]. In collaboration with the National Centre for Voice and Speech, Denver, Colorado, Max Little of the University of Oxford created the dataset. The dataset comprises 195 voice samples with 22 features. An overview of the features used in this study is presented in Table1.

Table 1: Overview Of Features Used In This Study

Feature	Description
F_0, F_{10}, F_{hi}	Several measures of vocal fundamental frequencies.
Jitter (%), Jitter (Abs), RAP, PPQ, Jitter: DDP	Several measures of variation in fundamental frequency
Shimmer, Shimmer(dB), Shimmer: APQ3, Shimmer: APQ5, APQ, Shimmer: DDA	Several measures of variation in amplitude.
NHR, HNR	Measures of ratio of noise to tonal components in the voice.
RPDE, D2	Two nonlinear dynamical complexity measures.
DFA	Signal fractal scaling exponent.
spread1, spread2, PPE	Three nonlinear measures of fundamental frequency variation

3.2. Feature Selection

Feature selection facilitates the selection of relevant features from the original dataset. It helps reduce the dimensionality by eliminating the redundant and irrelevant features which in turn helps improve classification accuracy of the classifier. Four major components constitute the feature selection process namely a search policy which governs the selection of a new feature subset, an evaluation method, a terminating condition such as threshold and a validation process. Figure2 depicts the flowchart of the feature selection process. There are three approaches to feature selection, namely embedded, wrapper, and filter. Filter method is used in this study for feature selection.

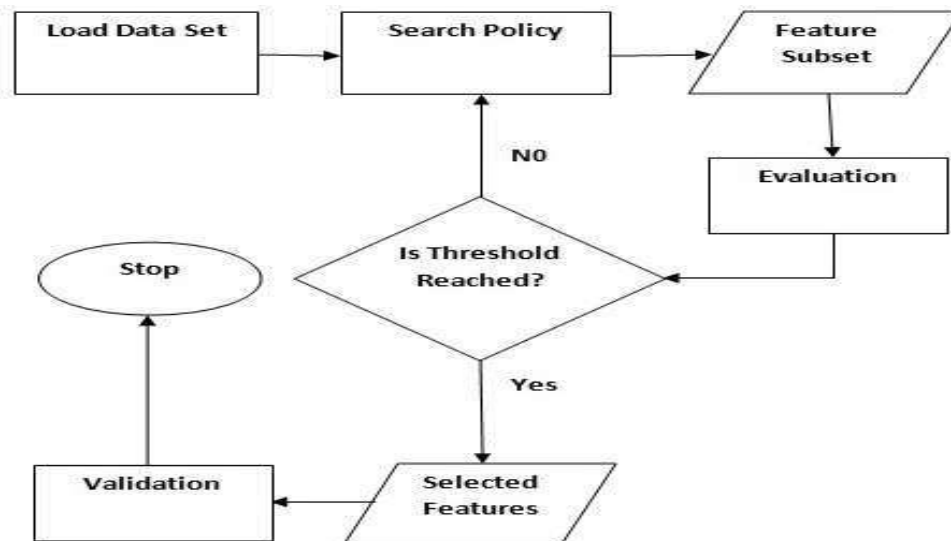


Figure2. Flowchart of Feature Selection Process

Embedded approach: Feature selection is directly integrated into the machine learning algorithm. The algorithm itself decides which features to use during the data mining operation. For example, algorithms that build decision tree classifiers, often use embedded approach.

Wrapper approach: It uses the target machine learning algorithm to find the optimum subset of features, but it is computationally intensive. Examples include Forward Selection, Recursive Feature Elimination (RFE), and Backward Elimination.

Filter approach: Feature selection is accomplished in a way which is independent of machine learning algorithm. Statistical measures are used to evaluate features which are independent of any machine learning model. As a result, it makes it faster among the three approaches. For example, ANOVA.

In theory, all possible combinations of features are searched in quest of the optimum feature subset. Ideally it should be inexpensive in terms of computational cost. The evaluation method helps to learn the goodness of a feature subset. It helps to ascertain the performance of the actual machine learning classifier with the selected feature subset. A terminating condition is necessary because it may not be possible to search a great number of subsets as it will computationally be expensive. As a final step, the performance of the target machine learning classifier with the selected subset should be validated.

3.2.1. ANOVA

One-way ANOVA F-test is a statistical technique used for feature subset selection. It uses variances of the group of features to test whether the features are significantly different or not, which helps eliminate the redundant features which in turn reduces the dimensionality and improve the overall predictive power of the classifier. The major advantages of using ANOVA as a feature selection method are outlined below:

Reduced dimensionality: ANOVA reduces the number of features in the training set by separating the redundant or irrelevant features.

Improved classifier accuracy: Predictive power of the machine learning algorithms is improved by filtering out noise in the form of irrelevant features.

Fast and scalable: Being independent of machine learning algorithm makes ANOVA computationally faster than embedded and wrapper methods which make it suitable for data preprocessing.

Significant feature selection: Features that have statistically significant effect on the dependent variable are chosen. A greater F-Score indicates that the feature can effectively classify the instance.

ANOVA F-test statistic is given by equation 1.

$$F = \frac{\text{Variability between-group}}{\text{Variability within-group}} \quad (1)$$

Variability between-group and Variability within- group are denoted by expression 2 and expression 3 respectively.

$$\sum_{i=1}^k n_i (\bar{X}_i - \bar{X})^2 / (K - 1) \quad (2)$$

where, $\bar{X}_i$, n_i , $\bar{X}$ and K denote the group mean in the i^{th} group, the number of observations in the i^{th} group, the overall mean and the number of groups respectively.

$$\sum_{i=1}^K \sum_{j=1}^{n_i} (\bar{X}_{ij} - \bar{X}_i)^2 / (N - K) \quad (3)$$

where, $\bar{X}_{ij}$, K and N denote the j^{th} observation in the i^{th} group, total number of groups (features) and sample size respectively.

Table2 presents the ANOVA F-test statistics of the baseline features of speech used in this study. The values (>1) in the F-Score column suggests that the speech features are significantly different and there is no irrelevant feature in the dataset.

Table2: F-Score and P-Value statistics of the baseline features

ID	Feature	F-Score	P-Value
18	spread1	90.42422	7.91E-18
21	PPE	75.8032	1.39E-15
19	spread2	50.34316	2.39E-11
0	MDVP: Fo (Hz)	33.28658	3.12E-08
2	MDVP: Flo (Hz)	32.61288	4.20E-08
8	MDVP: Shimmer	30.1226	1.26E-07
12	MDVP: APQ	29.53635	1.64E-07
15	HNR	29.01593	2.07E-07
11	Shimmer: APQ5	27.14503	4.82E-07
9	MDVP: Shimmer(dB)	27.06548	5.00E-07
10	Shimmer: APQ3	26.52712	6.39E-07
13	Shimmer: DDA	26.52563	6.39E-07
20	D2	25.26591	1.14E-06
4	MDVP: Jitter (Abs)	25.00173	1.28E-06
16	RPDE	20.31005	1.14E-05
6	MDVP: PPQ	17.54853	4.26E-05
3	MDVP: Jitter (%)	16.19281	8.21E-05
5	MDVP: RAP	14.77523	0.000164
7	Jitter: DDP	14.77267	0.000165
17	DFA	10.9529	0.001115
14	NHR	7.18327	0.007995
1	MDVP: F _{hi} (Hz)	5.478255	0.020276

3.3. Data Balancing

The problem of class imbalance manifests in a dataset when the data points of one class exceed the data points of another class. Imbalanced datasets affect adversely and causes machine learning model to be biased to the majority class. To address the problem of class imbalance, data balancing techniques are applied. There are two possible data-level approaches to address this problem- either to increase the minority samples or to decrease the majority samples to balance the dataset. The first and the last approaches are known as oversampling and undersampling respectively. The different approaches to data balancing have adopted different algorithms for producing a balanced dataset. Some of them are distance-based (e.g., Euclidian), some are cluster-oriented (e.g., k-means), while some others synthetically generate similar samples in the training set. As a consequence, different data balancing techniques have different impacts on the machine learning algorithms on the same dataset, producing varying classification accuracy. Hence, different data balancing techniques are explored in this study.

In oversampling, the instances of the training set bearing the minority class are increased so that the number of both the minority and the majority class instances match to each other. Consequently, it boosts model performance by reducing bias towards the majority class and improving predictive power of the model. Some major oversampling techniques are outlined below:

- **SMOTE:** SMOTE, short for Synthetic Minority Over-sampling Technique, is a statistical technique used to increase the minority-class instances of the dataset by synthetically creating them in a uniform manner, i.e., by creating synthetic instances uniformly along the line joining a minority class instance to its k-nearest neighbors.
- **Random Oversampling:** It is another statistical based technique used to address the class imbalance problem. It replicates the minority class samples by randomly selecting them (with replacement) and adds them to the training set. It is possible for a single sample to be chosen multiple times which increases the chance of model overfitting.
- **ADASYN:** ADASYN, short for Adaptive Synthetic Sampling, is an improvement upon SMOTE. It focuses on density distribution rather than uniformly generating instances of minority class,

i.e., it calculates density distribution for each minority class instance within a set radius (within k nearest neighbors) and assigns weights to them. More synthetic instances are created for the minority data point with higher weights.

In undersampling, the instances of the training set bearing the majority class are decreased so that the number of both the minority and the majority class instances match to each other. It improves the predictive power of the model by reducing bias towards the majority class, reduces model training time (in case of large datasets) and prevents model overfitting. Some major undersampling techniques are listed below:

- **Random Undersampling:** It is a statistical based technique used to handle the class imbalance problem by determining the desired class distribution before randomly removing instances of the majority class in the training set. One pitfall of this approach is the risk of losing potential information which may lead to model underfitting.
- **Cluster Centroids:** It uses clustering algorithms (e.g., k -means) to negotiate the class imbalance problem by applying it to the majority class to create clusters of similar instances and then replacing the clusters with their centroids to make a balanced dataset. The issue of losing potential information faced by random undersampling technique can be overcome by using Cluster Centroids.
- **NearMiss:** NearMiss is a distance-based approach (which often uses KNN) to handle class imbalance problem by considering average distance of the majority class instances to the three closest or furthest minority class instances, depending on the variants of the NearMiss algorithm. NearMiss-3, selects the k nearest neighbors of each minority class instance before selecting the majority class instances closest to each minority instance.

Figure3 presents the proportion of the classes of the imbalanced dataset before and after the application of data balancing techniques in this study.

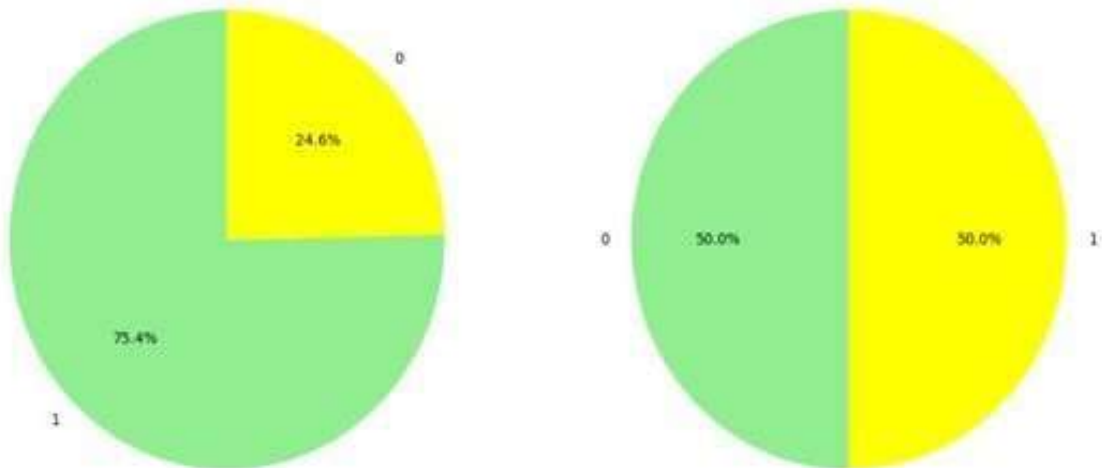


Figure 3. Proportion of the classes of the imbalanced dataset before and after applying data balancing techniques

3.4. Ensemble Technique

Ensemble Technique is applied to combine the predictions of multiple classifiers to obtain improved classification accuracy than the individual classifiers can do in isolation.

- **Bagging-** In bagging, weak learners are trained in parallel on different subsets of the same data set and finally the individual predictions are combined. Bagging stands for bootstrap aggregating. In this ensemble technique repeated samples with replacement, D_i ($i=1,2,3,\dots,n$) are taken from the original data set D . Because sampling with replacement is done, some instances of D may not be included in D_i , while others may occur several times in the same

training set. A classifier C_i is trained for each training set D_i . The class predictions returned by each classifier are treated as one vote and an unknown test instance is assigned to the class that receives the majority votes.

- **Boosting-** In boosting, weights are assigned to each training instance of the training set and a set of N classifiers are trained sequentially. Based on the classification accuracy achieved by a classifier C_i , weights of the training instance are updated so that the subsequent classifier C_{i+1} can work on the misclassified instances. AdaBoost is a popular boosting technique where each training instance is assigned equal weight. Given N classifiers, the ensemble requires N iterations, and at each iteration, the instances from the original data set D are sampled with replacement to form a training set D_i . Higher the weight, higher the chance of an instance being selected. Classifier C_i is trained and its error is calculated. The weight of a training instance is then increased if it is misclassified. Equation 4 computes the error rate of a classifier C_i by summing the weights of each instance that C_i classified incorrectly.

$$\text{Error}(C_i) = \sum_{r=1}^n w_r \times \text{err}(X_r) \quad (4)$$

Here, $\text{err}(X_r)$ denotes the error associated with the instance that C_i misclassified.

- **Random Forest-** Random forest ensemble technique is based on specifically decision tree classifiers. Each decision tree is trained on different subsets of the same data set and the final prediction is obtained by combining the predictions made by the individual decision trees. Each decision tree is generated by randomly selecting features from the training set. For generating N decision trees from the original data set D , at each iteration i (where $i=1,2,3,\dots,N$) a training set D_i is sampled (with replacement) so that some instances of D may be excluded from D_i , while others may occur several times in the same training set. The generalization error of random forest can be denoted by Expression 5.

$$\text{Generalization error} \leq \frac{\bar{\sigma}(1-s^2)}{s^2} \quad (5)$$

Here, $\bar{\sigma}$ is the mean correlation among the decision trees and s denotes the strength of the tree classifiers.

- **Voting-** The voting scheme provides a basis for combining multiple classifiers to enhance the predictive power of the individual models. There are mainly two broad categories of voting. In soft voting scheme, the average probabilities of predictions from each of the individual classifiers are estimated and the final prediction is based on the highest average probability of the class, whereas the hard voting scheme uses majority voting for the predicted class.
- **Stacking-** Stacking is an ensemble method of combining multiple classifiers of different types, unlike bagging and boosting where multiple classifiers of the same type are combined. For example, a Naïve Bayes, a decision tree, and a KNN can be combined to train a classifier C_s for a given data set D . Stacking can be applied to overcome the pitfall of voting. Suppose three classifiers are to be combined by voting scheme. If at least two of them produce mostly incorrect predictions, then the predictive power of the ensemble will drastically fall. In this case, it is not possible to identify the classifier in the ensemble to rely upon. Hence, in order to combine the predictions of the base classifiers in an optimum way, stacking introduces a meta-model to learn which classifiers to choose from.

3.5. Model Evaluation

Evaluating a machine learning model's performance is crucial after the model is trained. There is a technique to estimate model's classification accuracy known as holdout method. In this technique, the collected data are partitioned into two subsets namely training set and test set. The partition takes place on a random basis. 80% of the original data are assigned to the training set and the remaining

20% data are assigned to the test set. Model is trained by using the training set. Test set is then used to evaluate the performance in terms of classification accuracy of the model. Figure4 depicts the holdout method.

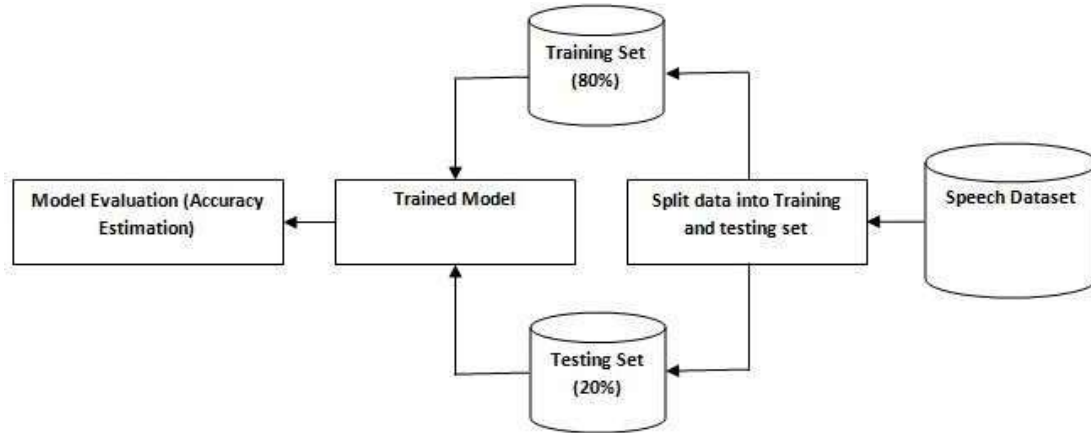


Figure 4. Model evaluation with the holdout method

Confusion matrix

Confusion matrix is a tool for evaluating performance of a machine learning model. This matrix can further be used to compute accuracy, sensitivity, specificity and F1-score. The elements of a confusion matrix are True Positive (TP), True Negative (TN), False Positive (FP) and False Negative (FN). TP is the number of positive instances that are predicted as positive by the ML model. TN is the number of negative instances that are predicted as negative by the ML model. FP is the number of negative instances that are predicted as positive by the ML model. FN is the number of positive instances that are predicted as negative by the ML model. Figure 3 depicts a Confusion Matrix.

	Predicted PD Patient	Predicted Healthy Subject
PD Patient	TP	FN
Healthy Subject	FP	TN

Figure 5. Confusion Matrix

Equation 6 through equation 9 gives the different metrics of model evaluation.

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN} \quad (6)$$

$$\text{Sensitivity} = \frac{TP}{TP+FN} \quad (7)$$

$$\text{Specificity} = \frac{\text{TN}}{\text{TN} + \text{FP}} \quad (8)$$

$$\text{F1} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (9)$$

4. RESULTS

In this section, the experimental results, the effects of feature selection, and data balancing techniques along with model evaluation are presented. Classification accuracies of different ensemble techniques are presented in Table3 through Table7. The abbreviations used in the tables are stated below:

DT-Decision Tree, KNN- K Nearest Neighbors, LR- Logistic Regression, SVC-Support Vector Classifier, GNB-Gaussian Naïve Bayes

Random Forest is trained with the speech dataset. Class-imbalance is addressed by four balancing techniques of Random under Sampler, ClusterCentroids, NearMiss and SMOTE. After applying SMOTE, ClusterCentroids and NearMiss, accuracy rate is improved by 5% for all the three data balancing techniques, while accuracy is declined by 5% after applying Random UnderSampling on the same dataset which is due to the loss of critical information during undersampling. Table3 presents the results of Random Forest.

Table 3: Classification accuracy of Random Forest

Data Balancing	Accuracy (%)
Not Applied	95.00
Random under Sampler	85.00
Cluster Centroids	96.61
NearMiss	96.61
SMOTE	96.61

A visual presentation of the results of Random Forest classifier after the application of various data balancing techniques is given in Figure6.

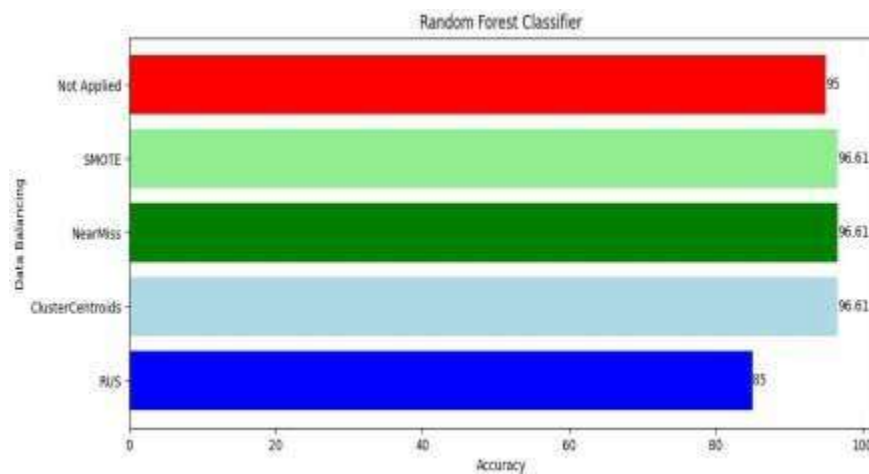


Figure 6. Results of Random Forest classifier with various data balancing techniques

Table4 presents the results of Bagging classifier which is implemented with three base estimators namely DT, KNN and LR to test their effectiveness with different types of data balancing techniques. DT implementation achieved 97% accuracy with 2.11% hike after applying SMOTE. KNN implementation stood in the second rank showing 90% accuracy with 7.95% improvement after

applying SMOTE. Bagging with LR as a base estimator was not significant with a decline of -12% in accuracy after applying SMOTE.

Table 4: Best performances of Bagging classifier (out of 15 models) in this study

Base Estimator	Data Balancing	Accuracy (%)
DT	Not Applied	95.00
DT	SMOTE	97.00
KNN	Not Applied	82.05
KNN	SMOTE	90.00
LR	Not Applied	85.00
LR	SMOTE	73.00

A visual presentation of the results of Bagging classifier after the application of data balancing techniques is given in Figure7.

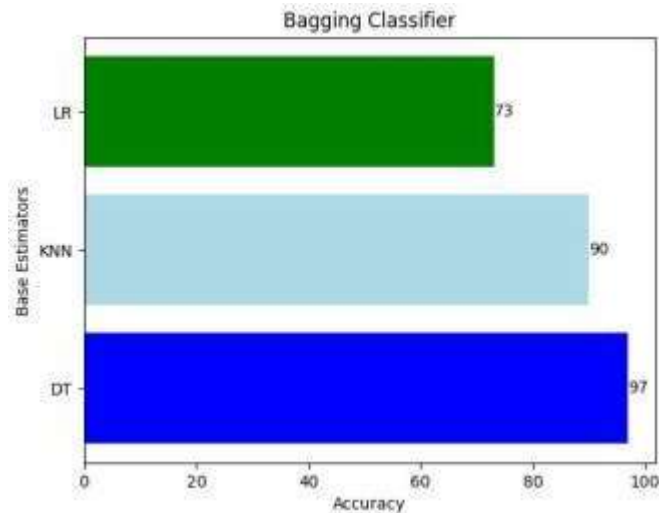


Figure 7. Results of Bagging classifier with various base estimators

Three implementations of boosting namely AdaBoost, GradientBoosting and XGBoost are tested with three undersampling techniques- Random Under Sampler, Cluster Centroids and NearMiss and one oversampling technique namely SMOTE. SMOTE outperformed the others with an improvement of 6% in AdaBoost, 2% in GradientBoosting and 3.31% in XGBoost classifier. AdaBoost and XGBoost performed equally good with an accuracy of 98.31%. Table5 presents the results of Boosting ensemble method.

Table 5: Best performances of Boosting classifiers (out of 15 models) in this study

Implementation	Data Balancing	Accuracy (%)
AdaBoost	Not Applied	92.31
AdaBoost	SMOTE	98.31
GradientBoosting	Not Applied	95.00
GradientBoosting	SMOTE	97.00
XGBoost	Not Applied	95.00
XGBoost	SMOTE	98.31

A visual presentation of the results of Boosting classifiers after the application of data balancing techniques is given in Figure8.

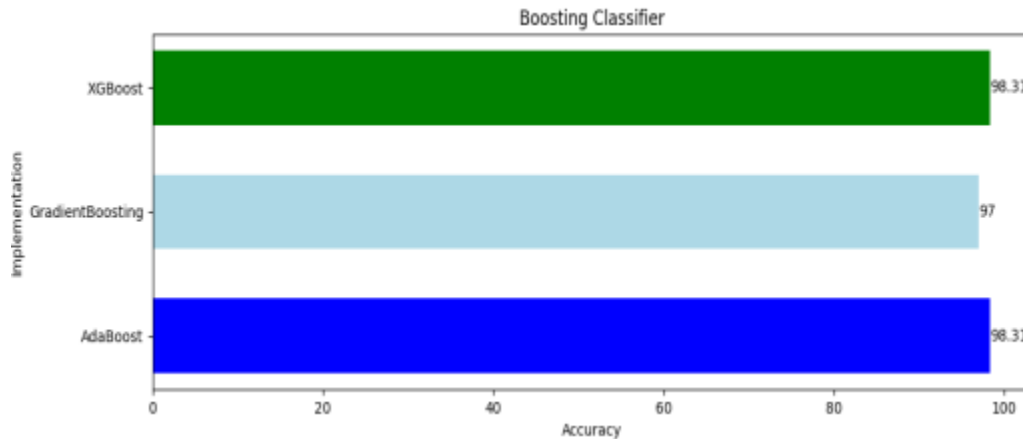


Figure 8. Results of Boosting classifier with different implementations

Table 6 presents the results of stacking ensemble method implemented with various base estimators such as KNN, SVC, LR, DT, and GNB. Logistic Regression is taken as the final estimator. This time also, SMOTE outperformed Random Under Sampler, Cluster Centroids and NearMiss. Stacking classifiers with class-imbalanced dataset gave 82% accuracy which is further increased by 15% to achieve 97% accuracy with both {LR, DT, KNN} and {LR, DT, KNN, GNB} distributions which outshine {KNN, SVC} as the base estimators to the stacking classifier.

Table 6: Best performances of Stacking classifiers (out of 13 models), Logistic Regression as the final estimator in this study

Base Estimators	Data Balancing	Accuracy (%)
KNN, SVC	Not Applied	82
KNN, SVC	SMOTE	83
LR, DT, KNN	SMOTE	97
LR, DT, KNN, GNB	SMOTE	97

A visual presentation of the results of Stacking classifiers after the application of data balancing techniques is given in Figure 9.

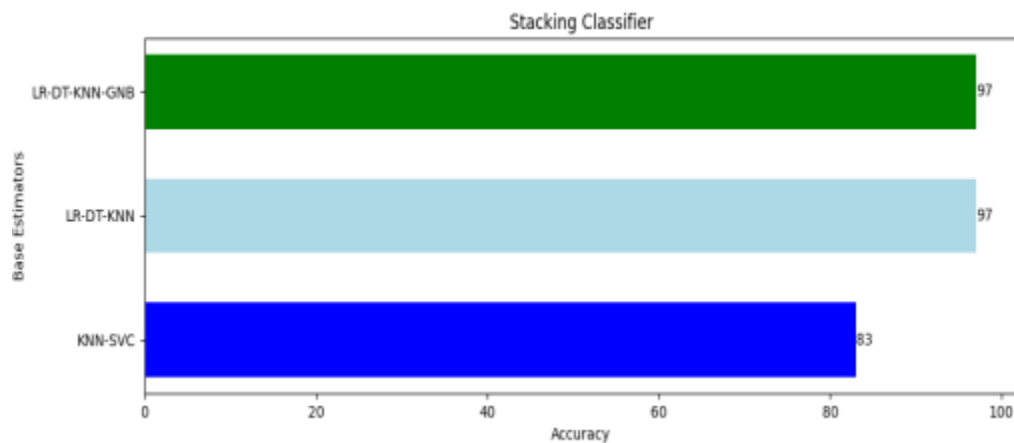


Figure 9. Results of Stacking classifier with various base estimators

Table 7 presents the results of voting ensemble method implemented with various base estimators (e.g., KNN, SVC, LR, DT, and GNB) and parameters such as soft and hard voting. The class-imbalance

problem is addressed by applying techniques such as Random Under Sampler, Cluster Centroids, NearMiss and SMOTE on the speech dataset. 70 models of voting classifier are tested in this study, but in this report only 10 best performing models are reported (Table7). The 8 distributions of base estimators namely {GNB, DT}, {LR, GNB, KNN}, {LR, GNB}, {LR, GNB, KNN}, {LR, GNB, DT, KNN}, {GNB, DT}, {GNB, DT, KNN}, and {LR, DT, KNN} achieved 5% improvement in accuracy after the application of NearMiss and Cluster Centroids resulting in 95% overall accuracy. The distributions namely {LR, GNB, DT, KNN} and {LR, DT, KNN} achieved 98% and 100% accuracy respectively after applying SMOTE. Thus, voting classifier achieved up to 10% improvement in accuracy after introducing data-balancing technique.

Table 7: Best performances of Voting (out of 70 models) in this study

Base Estimators	Data Balancing	Parameter	Accuracy (%)
GNB, DT	Not Applied	Soft	90
GNB, DT	NearMiss	Soft	95
LR, GNB, KNN	NearMiss	Soft	95
LR, GNB	NearMiss	Soft	95
LR, GNB, KNN	Cluster Centroids	Soft, Hard	95
LR, GNB, DT, KNN	Cluster Centroids	Hard	95
GNB, DT	Cluster Centroids	Soft	95
GNB, DT, KNN	Cluster Centroids	Soft, Hard	95
LR, DT, KNN	Cluster Centroids	Hard	95
LR, GNB, DT, KNN	SMOTE	Soft	98
LR, DT, KNN	SMOTE	Soft, Hard	98.31

A visual presentation of the results of Voting classifiers after the application of data balancing techniques is given in Figure10.

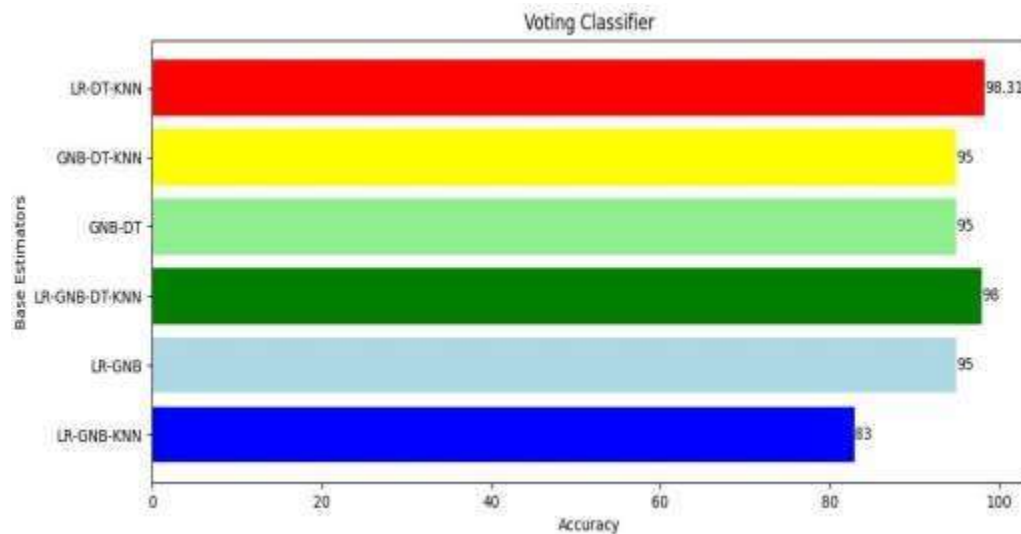


Figure 10. Results of Voting classifier with various base estimators

Total of 118 tests on the performance of machine learning models were performed and only 7 tests yielded satisfactory results. Out of these tests, XGBoost, AdaBoost and voting combined each with SMOTE, shine with outstanding performance of 98.31% detection rate.

Data balancing has a significant role, because it results in high accuracy in the prediction of PD. Table8 contrasts results with and without data balancing.

Table 8: Contrasts Results With And Without Data Balancing

Ensemble Technique	Maximum Accuracy (%)		Balancing Technique
	With Balancing	Without Balancing	
Random Forest	96.61	95	CC ^a , NM ^b , SMOTE
Soft voting*	98.31	87.18	SMOTE
Bagging**	96.61	95	CC ^a , NM ^b
GradientBoosting	97	95	SMOTE
XGBoost	98.31	95	SMOTE
AdaBoost	98.31	92.31	SMOTE
Stacking	97	82	SMOTE

^aClusterCentroids, ^bNearMiss, *Estimators are Logistic regression, Decision tree and KNN, ** Decision tree as base estimator

From Table8, by applying data balancing to the imbalanced dataset, average accuracy in the prediction of PD is increased by 6.34%. Figure11 presenting performance improvements of the classifiers after the application of data balancing techniques which shows stacking with the highest achievement of 18.29%, while GradientBoosting with the minimum gain of 2.11% of classification accuracy.

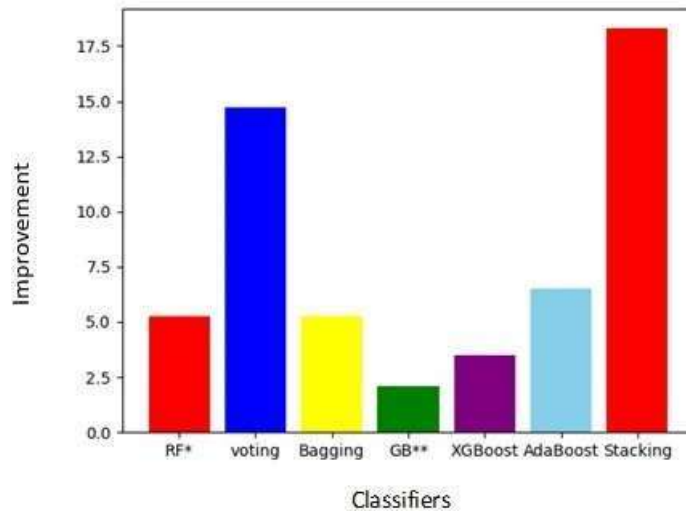


Figure 11. Performance improvement after the application of data balancing. *Random Forest, ** GradientBoosting.

Table 9 presents model-wise the metrics of performance evaluation of the best test cases out of 118 tests.

Table 9: Performance Metrics Of Best Test Cases Out Of 118 Tests

Ensemble Technique	Balancing	Accuracy (%)	Sensitivity (%)	Specificity (%)	F1-Score (%)
Random Forest	CC ^a , NM ^b , SMOTE	96.61	97	96.66	97
Voting	SMOTE	98.31	100	96.55	98

Bagging	CC, NM	96.61	93	100	97
Gradient-Boosting	SMOTE	97	93	100	97
XGBoost	SMOTE	98.31	97	100	98
AdaBoost	SMOTE	98.31	100	96.55	98
Stacking	SMOTE	97	93	100	97

^aClusterCentroids, ^bNearMiss

Table 10 compares the execution time (nanosecond) of each of the hybrid models that showed outstanding performance with accuracy of 100% in Table p.

Table 10: Listing Hybrid Models In The Ascending Order Of Their Execution Time

Ensemble + Balancing Technique	Execution Time (ns)
AdaBoost + SMOTE	10101100
XGBoost + SMOTE	104130400
Voting* (Soft) + SMOTE	160982200
Random Forest + CC ^a	201566300
Random Forest + NM ^b	205300500
Voting* (Hard) + SMOTE	220015300
Random Forest + SMOTE	246485900
Gradient-Boosting + SMOTE	303497500
Bagging + CC	326253000
Bagging + NM	329068800

^aClusterCentroids, ^bNearMiss, *Estimators taken as Logistic regression, Decision tree and KNN.

Based on the Table9 and Table10, Table11 presents accuracy and execution time along with Gain in Speed calculated from the average execution time and execution time of individual classifiers which shows that AdaBoost outperforms the rest of the classifiers with 98.31 % of accuracy and 94.52 % of speed faster than the average execution time of all the models.

Table 11: Accuracy and gain in speed of the top performing models of this study

Classifier	Accuracy	Execution Time (ns)	Gain in Speed
AdaBoost	98.31 %	10101100	94.52 %
XGBoost	98.31 %	104130400	43.54 %
Voting	98.31 %	160982200	12.71 %
Random Forest	96.61 %	201566300	-9.30 %
Gradient-Boosting	97.00 %	303497500	-64.57 %
Bagging	96.61 %	326253000	-76.91 %

Receiver Operating Characteristic (ROC) is a graphical tool for the performance evaluation of a classifier by plotting the True Positive Rate (Sensitivity) and False Positive Rate (1-Specificity) along the vertical and horizontal axes respectively. It demonstrates the goodness of an ML model in distinguishing between the classes. Graphs closer to the top-left corner denotes better performance. Area Under the Curve (AUC) is used to indicate the overall performance of the models, with values closer to 1 indicate a good model. Figure12 presents the ROC curves of the ensemble classifiers picked up from Table11, where the curves of the classifiers overlap each other having AUC=1 except for AdaBoost (AUC=0.98) in the sky-blue color.

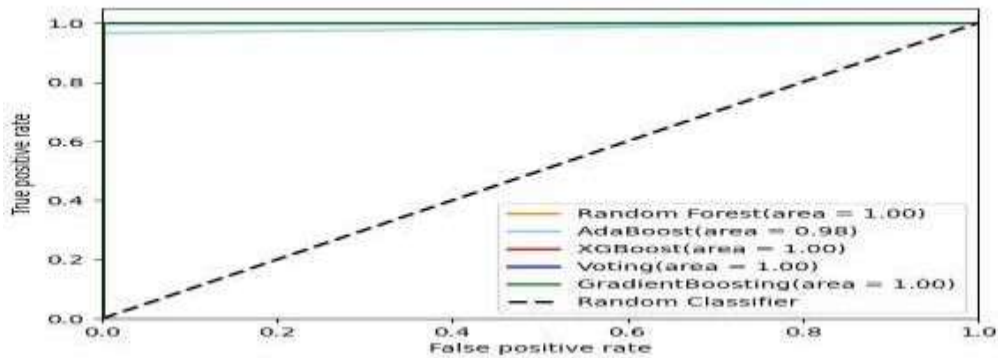


Figure 12. ROC curve of various classifiers used in this study

Figure13 presents confusion matrix of the top four classifiers with respect to accuracy and speed from Table11. In figure 13(a), AdaBoost comes up with 0 false negative whereas in figures 13(b), 13(c), and 13(d) the classifiers XGBoost, Gradient-Boosting and Random Forest respectively demonstrate false negatives, i.e., there is a chance of a PD patient remains undetected.

$\begin{bmatrix} 30 & 0 \\ 1 & 28 \end{bmatrix}$	$\begin{bmatrix} 29 & 1 \\ 0 & 29 \end{bmatrix}$
(a)	(b)
$\begin{bmatrix} 28 & 2 \\ 0 & 29 \end{bmatrix}$	$\begin{bmatrix} 28 & 1 \\ 1 & 29 \end{bmatrix}$
(c)	(d)

Figure 13. Confusion Matrix of (a) AdaBoost (b) XGBoost (c) Gradient-Boosting and (d) Random Forest

Table12 presents a comparison of the results of this study with state-of-the-art studies of predicting PD using speech.

TABLE 12: Comparison With State-Of-The-Art Methods Of Predicting Pd Using Speech

Study	Preprocessing	Classifier	Accuracy (%)
Alalayah et al. [17]	SMOTE	Decision Tree	95.00
Aishwarya et al. [18]	SMOTE	Light GBM	81.35
Shyamala et al. [19]	SVMSMOTE	XGBoost	96.61
Rahman et al. [20]	RS ^a , PCA	DNN2	95.41
Mondol et al. [21]	SMOTE	MLP	95.48
Lamba et al. [22]	Genetic Algorithm	RF	95.58
Bukhari et al. [23]	SMOTE, PCA	AdaBoost	96.00
Cesare et al. [24]	K-means clustering	SVM, KNN, Neural Network	92.30
This Study	SMOTE, ANOVA	AdaBoost	98.31

^aRandom Over Sampling, ^bClusterCentroids, ^cNearMiss

5. Conclusion And Future Work

In this study, a total of 118 machine learning models are tested with seven ensemble methods namely random forest, bagging, AdaBoost, GradientBoosting, XGBoost, stacking and voting to predict PD using speech in a non-invasive and cost-effective way. The class-imbalanced problem of the speech dataset is resolved with various data balancing techniques like SMOTE, Cluster Centroids, NearMiss and Random Under Sampler. ANOVA F-test is done to serve as a feature selection method to improve performance and reduce dimensionality. Data balancing technique improves the average accuracy up to 6.34%. Ensemble classifiers namely AdaBoost, XGBoost and Voting share the highest accuracy of 98.31% in this study; however, AdaBoost performs 94.52% faster than the average execution time of all the classifiers. Moreover, AdaBoost comes up with 0 false negative, whereas the other classifiers demonstrate false negatives. Although the result of this study is satisfactory, stratified k-fold cross validation, a more reliable validation technique can be adopted to obtain more stable and reliable estimates of classifier performance. More advanced data-balancing techniques such as ADASYN can be applied to improve classification accuracy of the machine learning models. The dataset of this study is not sufficiently large; hence study should be conducted by considering large samples so as to come up with a more reliable conclusion.

Statements And Declarations

Funding

The authors declare that no funds, grants, or other support were received during the preparation of this manuscript.

Conflicts of Interest

The authors of this research paper have no conflict of interest and there is no copyright issue.

Competing Interests

The authors have no relevant financial or non-financial interests to disclose.

Ethics approval

This article does not contain any studies with human or animal participants. No ethics approval was required.

Consent to participate

This article does not contain any studies with human. No consent to participate was required.

Consent to publish

This article does not contain any studies with human. No consent to publish was required.

Data Availability Statement

The data used in this research is taken from UCI machine learning repository. The URL of the data source is:

<https://archive.ics.uci.edu/dataset/174/parkinsons>

REFERENCES

- [1] R. A. Shirvan and E. Tahami, "Voice Analysis for Detecting Parkinson's Disease Using Genetic Algorithm and KNN Classification Method," no. December, pp. 14–16, 2011.
- [2] Ahlrichs, C., and Lawo, M. (2013). Parkinson's disease motor symptoms in machine learning: a review. arXiv preprint arXiv:1312.3825. doi: 10.5121/hij.2013.2401
- [3] Tsanas, A., Little, M. A., McSharry, P. E., Spielman, J., and Ramig, L.O. (2012). Novel speech signal processing algorithms for high-accuracy classification of Parkinson's disease. IEEE Trans. Bio. Med. Eng. 59, 1264–1271. doi: 10.1109/TBME.2012.2183367
- [4] Harel B, Cannizzaro M, Snyder PJ. Variability in fundamental frequency during speech in prodromal and incipient Parkinson's disease: A longitudinal case study. Brain and cognition. 2004 Oct 31; 56 (1):24–9.
<https://doi.org/10.1016/j.bandc.2004.05.002> PMID: 15380872

- [5] Tsanas A, Little MA, McSharry PE, Ramig LO. Accurate telemonitoring of Parkinson's progression by noninvasive speech tests. *IEEE transactions on Biomedical Engineering*. 2010 Apr; 57(4):884– 93. <https://doi.org/10.1109/TBME.2009.2036000> PMID: 19932995
- [6] Palma, J.A., Kaufmann, H.: Autonomic disorders predicting Parkinson disease. *Parkinsonism Relat Disord*. 2014 January; 20(0 1): S94–S98. doi:10.1016/S1353 8020(13)70024-5.
- [7] Chaudhuri, K.R., Healy, D.G., Schapira, A.H.V.: Non-motor symptoms of Parkinson's disease: diagnosis and management. *Lancet Neurol* 2006; 5: 235–45
- [8] World Health Org. Parkinson Disease. Accessed: Nov. 20, 2025. [Online]. Available: <https://www.who.int/news-room/fact-sheets/detail/parkinson-Disease>
- [9] Castillo-Barnes, D., Ramírez, J., Segovia, F., Martínez-Murcia, F. J., Salas-Gonzalez, D., and Górriz, J. M. (2018). Robust ensemble classification methodology for I123-Ioflupane SPECT images and multiple heterogeneous biomarkers in the diagnosis of Parkinson's disease. *Front. Neuroinf.* 12:53. doi: 10.3389/fninf.2018.00053
- [10] Adeli, E., Shi, F., An, L., Wee, C.-Y., Wu, G., Wang, T., et al. (2016). Joint feature-sample selection and robust diagnosis of Parkinson's disease from MRI data. *NeuroImage* 141, 206–219. doi: 10.1016/j.neuroimage.2016.05.054
- [11] Xu, N., Zhou, Y., Patel, A., Zhang, N., Liu, Y. Parkinson's Disease Diagnosis beyond Clinical Features: A Bio-marker using Topological Machine Learning of rs-fMRI. Available: <https://ssrn.com/abstract=4165341>
- [12] Abujrida, H., Agu, E., and Pahlavan, K. (2017). "Smartphone-based gait assessment to infer Parkinson's disease severity using crowdsourced data," in 2017 IEEE Healthcare Innovations and Point of Care Technologies (HI-POCT) (Bethesda, MD), 208–211. doi: 10.1109/HIC.2017.8227621
- [13] Adams, W. R. (2017). High-accuracy detection of early Parkinson's Disease using multiple characteristics of finger movement while typing. *PLoS ONE* 12: e0188226. doi: 10.1371/journal.pone.0188226
- [14] Al-Fatlawi, A. H., Jabardi, M. H., and Ling, S. H. (2016). "Efficient diagnosis system for Parkinson's disease using deep belief network," in 2016 IEEE Congress on Evolutionary Computation (CEC) (Vancouver, BC), 1324–1330. doi: 10.1109/CEC.2016.7743941
- [15] UCI Machine Learning Repository, website: <https://archive.ics.uci.edu/dataset/174/parkinsons>
- [16] 'Exploiting Nonlinear Recurrence and Fractal Scaling Properties for Voice Disorder Detection', Little MA, McSharry PE, Roberts SJ, Costello DAE, Moroz IM. *BioMedical Engineering OnLine* 2007, 6:23 (26 June 2007)
- [17] K. M. Alalayah, E. M. Senan, H. F. Atlam, I. A. Ahmed, and H. S. A. Shatnawi, "Automatic and early detection of Parkinson's disease by analyzing acoustic signals using classification algorithms based on recursive feature elimination method," *Diagnostics*, vol. 13, no.11, p. 1924, May 2023.
- [18] R. Aishwarya, K. Pavitra, P. V. Miranda, K. Keerthana, and L. Kamatchi Priya, "Parkinson's disease prediction using Fisher score based recursive feature elimination," in *Proc. Int. Conf. Advancement Comput. Comput. Technol. (InCACCT)*, May 2023, pp. 1–8.
- [19] Shyamala, K., and Navamani, T.M. Design of an Efficient Prediction Model for Early Parkinson's Disease Diagnosis. *IEEE Access*, vol. 12, Oct. 2024. doi: 10.1109/ACCESS.2024.3421302
- [20] Rahman, S.; Hasan, M.; Sarkar, A.K.; Khan, F. Classification of Parkinson's disease using speech signal with machine learning and deep learning approaches. *Eur. J. Electr. Eng. Comput. Sci.* 2023, 7, 20–27.
- [21] Mondol, S.R., Kim, R.; Lee, S. Hybrid machine learning framework for multistage Parkinson's disease classification using acoustic features of sustained korean vowels. *Bioengineering* 2023, 10, 984.

- [22] Lamba, R.; Gulati, T., Alharbi, H.; Jain, A. A hybrid system for Parkinson's disease diagnosis using machine learning techniques. *Int. J. Speech Technol.* 2022, 25, 583–593.
- [23] Bukhari, S.N.H.; Ogudo, K.A. Ensemble Machine Learning Approach for Parkinson's Disease Detection Using Speech Signals. *Mathematics* 2024, 12, 1575. <https://doi.org/10.3390/math12101575>
- [24] Di Cesare, M.G.; Perpetuini, D.; Cardone, D.; Merla, A. Machine Learning-Assisted Speech Analysis for Early Detection of Parkinson's Disease: A Study on Speaker Diarization and Classification Techniques. *Sensors* 2024, 24, 1499. <https://doi.org/10.3390/s24051499>
- [25] Ma, H., Tan, T., Zhou, H., and Gao, T. (2016). "Support Vector Machine- recursive feature elimination for the diagnosis of Parkinson disease based on speech analysis," in 2016 Seventh International Conference on Intelligent Control and Information Processing (ICICIP) (Siem Reap), 34–40. doi: 10.1109/ICICIP.2016.7885912
- [26] Bakar, Z. A., Ispawi, D. I., Ibrahim, N. F., and Tahir, N. M. (2012). "Classification of Parkinson's disease based on Multilayer Perceptrons (MLPs) neural network and ANOVA as a feature extraction," in 2012 IEEE 8th International Colloquium on Signal Processing and its Applications (Malacca), 63–67. doi: 10.1109/CSPA.2012.6194692
- [27] Haq, A. U., Li, J., Memon, M. H., Khan, J., Din, S. U., Ahad, I., et al. (2018). "Comparative analysis of the classification performance of machine learning classifiers and deep neural network classifier for prediction of Parkinson disease," in 2018 15th International Computer Conference on Wavelet Active Media Technology and Information Processing (ICCWAMTIP) (Chengdu), 101–106.
- [28] Islam, M. S., Parvez, I., Hai, D., and Goswami, P. (2014). "Performance comparison of heterogeneous classifiers for detection of Parkinson's disease using voice disorder (dysphonia)," in 2014 International Conference on Informatics, Electronics & Vision (ICIEV) (Dhaka), 1–7. doi: 10.1109/ICIEV.2014.6850849
- [29] Haq, A. U., Li, J. P., Memon, M. H., Khan, J., and Malik, A., Ahmad, T., et al. (2019). Feature selection based on L1-Norm support vector machine and effective recognition system for Parkinson's disease using voice recordings. *IEEE Access* 7, 37718–37734. doi: 10.1109/ACCESS.2019.2906350