CLASSIFICATION MODELS COMPARISON FOR PARKINSON'S DISEASE DETECTION THROUGH SPEECH FEATURES

CHRISTOPHER PUTRA SETIAWAN[1], EDWARD PRATAMA PUTRA[2], GEDE PUTRA KUSUMA[3]

Computer Science Department, BINUS Graduate Program - Master of Computer Science, Bina Nusantara University, Jakarta, Indonesia, 11480

E-mail: [1]christopher.setiawan002@binus.ac.id, [2]edward.putra002@binus.ac.id, [3]inegara@binus.edu

ABSTRACT

Prior researches have used speech features to classify Parkinson's disease. This research was conducted to compare thirteen classification models in order to achieve a better performance in the newly published dataset. The method offered in this paper involves the usage of hyperparameter tuning, feature selection using genetic algorithm, and model stacking with ANN as the final classifier. The final stacked model has achieved an accuracy of 90.13% in the test dataset, with an F1- score of 93.7%. The results indicate that classification accuracy of Parkinson's disease classification through speech features can be increased by utilizing hyperparameter tuning, feature selection, and ensemble stacking. This improvement also indicated that the usage of feature selection in tandem with ensemble stacking method can yield better result compared to prior state of the art models for Parkinson's disease detection.

Keywords: *Parkinson's Disease Detection, Machine Learning, Ensemble Stacking, Hyperparameter Tuning, Feature Selection*

1. INTRODUCTION

According to the US' National Institute of Aging [1], Parkinson's disease is a type of brain disorder which leads to incessant shaking, stiffness, and difficulty with walking, self-balancing, and coordination. Currently, there are more than 10 million people suffering from Parkinson's disease worldwide [2]. Due to the increasing number of patients and the severity of the symptoms, an early detection of Parkinson's disease can be indispensable in slowing down the disease progression, while giving them a chance to seek proper treatment or medication therapy [3].

Currently, there is no conclusive screening or test criteria to diagnose a patient with Parkinson's disease. Therefore, doctors and medical professionals would usually weigh the symptoms that are experienced by the patient alongside family history, as opposed to giving a conclusive test, such as a blood test [4]. While inconclusive, prior researches have analyzed speech data of patients diagnosed with Parkinson's disease in order to find underlying patterns that could be used to create classification models for Parkinson's disease classification and prediction.

As of the writing of this paper, the majority of researches regarding Parkinson's disease classification have used the UCI Parkinson's dataset created by Little et al. [5], consisting of speech data from 31 people (23 with Parkinson's disease). Due to the popularity of this dataset, recently proposed classification methods have reached an accuracy of 100% [6] However, a new dataset has recently been released by [7], consisting of preprocessed speech data gathered from 188 patients with Parkinson's disease and 64 healthy individuals as control group.

Therefore, this paper aims to compare multiple classification algorithms using the recently released dataset, in order to classify Parkinson's disease through preprocessed speech data. Through this research, it is expected that an increase in classification accuracy can be discerned from prior research of the same dataset by Sakar et al., thus providing a potential baseline for assisting doctors in diagnosing Parkinson's disease. Furthermore, a high degree of correlation and accuracy would also open the door for future development of a self-diagnosis kit in the form of mobile application, as a gateway to getting professional help from healthcare professionals.

The classification algorithms that will be compared and evaluated in this study include: Decision Tree, Support Vector Machine (SVM), Artificial Neural Network (ANN), Random Forest, Bagging Tree, Gradient Boosted Trees (GBT), and XG-Boost. Furthermore, the comparison and evaluation of the various classification models used in this research will be conducted in three phases. The first phase is by utilizing all 754 speech features to create the models and tune the hyperparameters. The second phase is utilizing Genetic Algorithm (GA) to conduct feature selection to further increase the obtained accuracy, The last phase involves creating an ensemble stacking based on the three best performing models with ANN as the final classifier.

The paper is structured as follows: section two discusses several related works on Parkinson's disease classification, section three discusses the dataset and methodology used in this research, section four discusses the findings of the experiment, and section five discusses the conclusion of this research and potential for future works.

2. RELATED WORKS

Numerous studies in recent years have delved on the topic of Parkinson's disease classification, many of which are based on voice features. [8] early study on Parkinson disease used Neural Network, DM Neural, and Decision Tree to predict Parkinson's disease diagnosis using a voice dataset from Oxford University, claiming the neural net model reached 92% accuracy. Then Sakar et al. [9] made a new Parkinson dataset in their research, reporting 77% accuracy using SVM with linear kernel, whereas Naranjo et al. [10] used Bayesian Classifier which was reported to have 75% accuracy on their self-constructed dataset.

Fayyazifar and Samadiani [11] proposed a bagging ensemble model with GA optimization for feature selection, reporting a 98% accuracy with only seven selected features. Later on, a research conducted by Karan, Mahto, and Sahu [12] utilized variational mode decomposition and mel-frequency cepstral coefficient (MFCC) of the speech data as the key features to be fed into SVM model with the result of 96% accuracy. Polat [13] have used SMOTE data processing to deal with the imbalance dataset. A 94% accuracy is reported by using Random Forest classifier. Later, Mittal and Sharma [14] reported the result of Support Vector Machine (SVM), Logistic Regression, and K Nearest Neighbor (KNN) resulting in accuracies of 89%, 85%, and 90% respectively, using PCA feature dimension reduction focused on acoustic features.

Although the researches above have reached a significant degree of accuracy, the proposed predictive models can still be improved beyond the optimization of machine learning models, especially concerning the features used. Lately, several studies have proposed various feature extraction methods from the voice data to be analyzed and used for predictive purposes. Abhishek et al. [15] proposed a new set of features for parkinson voice predictions consisted of Fhi (Hz), Jitter(percent), Flo (HZ), Jitter (ABS), Shimmer, Fo (Hz), Jitter (RAP), Shimmer (APQ5) and HNR with the end result of 77% accuracy. Although the features are not guaranteed to be better, studies to this matter are still necessary to be deepened. Soumaya et al. [16] proposed an SVM model with Genetic Algorithm (GA) optimization for feature selection focused on the IMFCC feature (Inverse MFCC). The fact that the model could reach 91% accuracy on the new features by only using SVM and GA really shows its potential.

Recently, Sakar et al. [7] proposed a new set of features extracted from the Parkinson voice dataset using Tunable Q Wavelet Transforms, an improvement from their published 2013 Parkinson dataset. Based on the results, 755 features were extracted using seven different voice extraction method consisted of Mel-Frequency Cepstral Coefficient (MFCC), Wavelet Transform (WT), Glottis Quotient (GQ), Glottal to Noise Excitation (GNE), Vocal Fold Excitation Ratio (VFER), and Empirical Mode Decomposition (EMD). The reason behind the choosing of TQWT is its efficiency and claimed to be a more robust time-scale representation. Sakar [7] constructed different kinds of classifiers which ended up to the SVM model with the rbf kernel with the accuracy of 86%. Later, a study was conducted by Durgut, Baydilli, and Aydin [17] using those features. They reported an accuracy of 79% using SVM and Artificial Bee Colony for its feature selection method.

Therefore, by looking at the opportunities on the 754 features retrieved from Sakar et al. [7] research, this study aims to explore the potential of those features and its implementation on the best suited models in consideration to increase the accuracy from the past studies on Sakar's dataset. It is expected that the model can yield better performance by implementing feature selection and hyperparameter tuning with ensemble stacking method. We tried seven different models consisting of three singular classification models such as SVM, ANN, and Decision Tree, and four ensemble models such as Bagging Tree, Random Forest, XGBoost, and Gradient Boosted Tree. All of the models are tuned, and GA is also used for feature selection.

Those models are assessed and eliminated before combining into a stacked model using ensemble stacking method. Five models are used for the weak learners, and ANN is used for the meta learner.

3. THEORY AND METHODS

3.1 Decision Tree

Decision Tree is a set of nodes and branches that can be used for classification problems. According to Song [18], the decision tree consists of four important concepts which are the nodes, branches, stopping, and pruning. Each node represents a single unit of decision makers inside the decision tree. There are three types of nodes which are the input node, internal decision node, and result node. Input node receives a given input signal and passes the input to the first internal decision node.

Branches exist in every decision node. Every node will have a Boolean rule to decide the branch path should be taken which leads to the next node. Usually, there will be a minimum of two branches in a single decision node [19].

Stopping is important in developing the decision tree, as trees can be quite big depending on the given problem. Therefore, stopping criteria should be determined before constructing the tree. Stopping criteria can be measured by using the minimum number of records in a leaf, minimum number of records prior to splitting, and depth of the tree itself. According to Berry, Linoff, and Gordon [20] the best target proportion of records in a leaf node should be more than 0.25 to prevent underfitting and less than 1 to reduce overfitting. Pruning in a decision tree is a process of cutting down several branches that are not giving significant changes.

3.2 Artificial Neural Network

According to Shanmuganathan and Samarasinghe [21], an artificial neural network is a form of computational model that is based on the human brain. The components of ANN include neurons as processing units and weighted connections between neurons.

Besides the two main components, ANN is also defined by four primary parameters, which include: type of neuron, connection architecture such as feedforward or feedback architecture, learning algorithm for the model, and recall algorithm to calculate results.

Although ANN can be a universally useful model in approximating functions, drawbacks can be found in the knowledge extraction process as ANN models are traditionally used in black box approaches.

3.3 Support Vector Machine

Support vector machine is a classification model that relies on mathematical computations over the hyperplane parameters to separate clusters of data. According to [22], there are three main concepts in support vector machines which are the hyperplane, optimal hyperplane margin, and soft margins.

Hyperplane is a mathematical line equation that is expected to separate the data into classes. This hyperplane acts as the separator of the data that groups the data into their grouping area. Hyperplane can also be a polynomial equation that creates a curving separator. If the data cannot be separated linearly then it can be transformed into feature space [23]. Training the SVM will move the hyperplane across the result area to find the best correct line equation.

Moreover, finding the optimal hyperplane is the goal of the SVM training. The hyperplane can correctly classify the data, but the hyperplane might be not optimal. An optimal hyperplane is the hyperplane that has the maximum distance to the closest data [24]. Optimal hyperplane should have the maximum margin to the result point in both of the classes which indicates the ability to separate the classes exactly in the middle of both clusters.

However, in real world cases, not all problems can be classified cleanly. Therefore, soft margins take place in this problem. Some of the results that could not be classified correctly can be expelled from the margin so the final result will not be affected by those unpredictable points.

3.4 Random Forest

Random forest is a machine learning model that consists of many decision trees; hence it is called a random forest. According to [25] random forest is capable of predicting either regression or classification problems. Although random forest is an ensemble model, its process time can compete with other singular models due to its algorithm [26].

Each tree in a random forest is exactly the same as the decision tree which has nodes and branches to give the final result prediction. Random forest is capable of giving a better prediction result since it has more decision trees that may have different outputs that will be used for final prediction determination.

The problem in random forest is as simple as finding the best decision tree structure for all of the trees and also finding the optimal number of trees. More trees will affect the final prediction in a good manner. However, there will be a certain point where the prediction accuracy will not be significantly affected [27]. This condition means the random forest has reached its optimal number of trees.

3.5 Bagging Tree

Bagging tree is the ensemble method of tree models. The bagging tree uses trees as weak learners that will be used in the aggregating model. This model is similar to the random forest. In short, it is a model that uses a bootstrapped dataset and an aggregating method [28]. Many classification models can be used for bagging techniques. When the model used is a tree, it can be called a bagging tree. On the other hand, a random forest is an ensemble of decision trees.

Bagging tree can use varying types of trees. The trees are used as weak learners and used for the aggregating model. The main difference between bagging and decision tree is there are only several random best split features selected for the splitting decision in the decision tree. On the other hand, bagging tree use all of the features.

3.6 Gradient Boosting

According to [29], the gradient boosting tree is the next implementation of the decision tree. This ensemble technique uses many decision tree to be used for its final prediction. Although gradient boosting and random forest are similar, gradient boosting tree uses boosting technique, instead of the voting method of the random forest. Boosting means it uses the errors of the previous trees to boost the next trees.

Decision trees are created sequentially. Each decision tree will be evaluated using any loss function. The result of the evaluation will be used to train the second tree with the goal to decrease the error on the second tree using the gradient calculation. Gradient descent is used to decrease the loss as fast as possible.

3.7 XGBoost

According to [30], XGBoost is an implementation of the gradient boosting tree with claims of a faster process time up to 10 times compared to the usual implementation of gradient boosting. XGBoost is the pre-defined gradient boosting model which is the improvement of the gradient boosting implementation. Its improvement covers many different areas, including:

1) Process time is claimed to be 10 times faster than the usual
2) Can be used for several data inputs
3) Customization on the objective and loss function
4) The performance is also claimed to be better on several datasets.

3.8 Genetic Algorithm

Genetic algorithm is an algorithm that mimics the genetic combination and mutation of nature. This concept is initially introduced by [31]. This algorithm is now commonly used for feature selection.

Genetic algorithm is an algorithm that tries different sets of data to make a new population of generations. Each individual's performance is evaluated using a fitness function. Individuals that have the best fitness are chosen to be the next parent. The next generation will be created based on the genes of the parents in the process of crossover and mutation. Crossover is a process to combine N number of parent A's gene with L-N number of parent B's gene, where L is the length of the gene combination, and N is the crossover ratio point. On the other hand, mutation is a process of replacing 0 to N genes of the new individual with some random values as it mimics the biological mutation of genes.

This process of breeding individuals is repeated for a pre-defined number of generations. The goal of this algorithm is to do crossovers and mutations to breed the best individual that can be proven using the fitness function [32].

3.9 Model Stacking

Model stacking, often called ensemble of ensembles, is a method which combines various classification models into a final meta classifier in order to obtain better general performance as compared to each individual model composing it [33].

There are two general methods to perform model stacking, which includes the usage of either differing or similar types of classifiers. In the first method, differing types of classifiers are trained using the same data with the final result being derived from majority voting or average result of the classifiers.

Whereas in the second method, bootstrap samples will be drawn from the training set to be used in each classifier separately. The final result of each classifier will then be combined as an ensemble. Therefore, this method is usually used with highly flexible models that can endure variance reduction in training data.

4. RESEARCH METHODOLOGY

The methodology of this research consists of four phases such as the data acquisition, data

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

extraction, modeling, and evaluation phase.

4.1 Data Acquisition

The dataset was retrieved from Kaggle based on Sakar et al.'s [7] research. The Parkinson patients' voice recording data comes from 252 subjects consisting of 188 Parkinson patients and 64 healthy controls, with an age range of 41 to 82 years. The recording process was conducted with a microphone setting of 44.1 KHz, in the Department of Neurology in CerrahpaÅŸa, Faculty of Medicine, Istanbul University. The process was also focused on each subject's sustained phonation of the vowel "a", following a physician's examination.

4.2 Feature Extraction

The voice data has been converted into 754 distinct numerical features by using several methodologies, including: Mel-Frequency Cepstral Coefficient (MFCC), Wavelet Transform (WT), Glottis Quotient (GQ), Glottal to Noise Excitation (GNE), Vocal Fold Excitation Ratio (VFER), and Empirical Mode Decomposition (EMD). After feature extraction, the final combined data consisted of 754 features and 756 rows (three recordings per subject) to be used for training, validation, and testing.

4.3 Modeling

4.3.1 Classification Model

We implemented and evaluated seven classification models, consisting of single and ensemble models such as: Decision Tree [34], Artificial Neural Network, Support Vector Machine [35], Random Forest [36], Bagging Tree [28], Gradient Boosting [29], and XGBoost [30]. Each of the models is trained using training data.

4.3.2 Hyperparameter Tuning

In this research, we tune the hyperparameter for each of the seven models. Hyperparameter tuning can increase the accuracy of the models as it can fit better to the domain area. Each of the models are tuned to find the best hyperparameter to be used for this Parkinson classification problem. All of the parameters are tuned to find the best fit for the prediction based on the accuracy of the four-fold cross validation on the training dataset only. We use a grid search algorithm [37] to tune the hyperparameters for each of the models. Each model has their own hyperparameters to be tuned which can be seen in Table I.

Table 1: Hyperparameter Tuning per Model

Model	Hyperparameters to be Tuned
Decision Tree	Maximum depth, minimum leaf, and criterion
Support Vector Machine (SVM)	C value, gamma, and kernel
Artificial Neural Network (ANN)	Number of layers and nodes, epochs, and batch size
Random Forest	Bootstrap, maximum features, minimum samples leaf, minimum samples splits, maximum depth, number of estimators, and criterion
Bagging Tree	Number of estimators
Gradient Boosting	Number of estimators and criterion
XGBoost	Booster type

4.3.3 Feature Selection Using Genetic Algorithm

From all of the models that were tuned, the top five models that have the highest four-fold cross validation accuracy were selected for the feature selection phase. Previously, the models were trained using all of the 756 features without any feature selection. Some of the features create noises that can affect the performance of the models. Therefore, feature selection is needed to eliminate and select the best features for the model. Machine learning models do not always work better on bigger features, in fact feature selection can identify the irrelevant features on the dataset to be excluded [38]. Each of the models might have different features compared to the others, since the feature selection is done separately to each of the models

We use the Genetic Algorithm [31] for our feature selection method. Genetic algorithm was proven to be an effective feature selection method compared to the other feature selection methods [39]. Genetic Algorithm takes several features and uses it to train the models. The best feature combinations are selected and combined together for the next couple of generations until it reaches the limit of the stated maximum number of generations or termination criteria. We set the maximum number of features to 100, number of populations to 50, and number of generations to 100. Crossover and mutations are allowed in this research. We set the crossover probability to 0.5, and the mutation probability to 0.2. The measurements for the best fitted features are calculated in accuracy in the four-fold cross validation.

The fitness criteria of this GA is the accuracy of the four fold cross validation on the training dataset. After the feature selection process, now the models have their own set of features that are retrieved from the genetic algorithm process. These models will be used as the weak learners for the stacked model.

4.3.4 Stacked Classification Model

The stacking ensemble classification model is one of the stacking techniques that is commonly used. This stacked model is using a stacking ensemble method [40]. This technique involves a certain number of weak learners that are tasked to predict the weak results of the problems, and a single meta learner that predicts the final results based on those weak prediction results. This model will have two phases of training. The first phase is to train weak learners. Then the second phase is to predict some testing data and use the prediction result to train the meta learner.

The first phase of the training involves our top five models from the previous hyperparameter tuning phase. All of the models are trained using weak train set data. Each model is trained using their own GA features that were retrieved from the feature selection phase, hence all of the models accept a different set of features. Then we make predictions using the weak test data on each of those models. The predicted values are collected in arrays to be used as train data for the meta learner.

The second phase of the training involves a single meta learner model. We use ANN as the meta learner of our stacking model. ANN receives five inputs/features, each comes from the weak learners. The training data for this meta model are the predictions from the weak learners with the y value from the weak test set. We train the ANN to be able to predict the final result from the five given results. We train the model using four-fold cross validation and also tune the hyperparameters. We search for the best layer architecture, epochs, and batch size of the ANN meta learner.

4.4 Evaluation

The stacking ensemble classification model is one of the stacking techniques that is commonly used. This stacked model is using a stacking ensemble method [40]. This technique involves a certain number of weak learners that are tasked to predict the weak results of the problems, and a single meta learner that predicts the final results based on those weak prediction results. This model will have two phases of training. The first phase is to train weak learners. Then the second phase is to predict some testing data and use the prediction result to train the meta learner. Training data are constructed by splitting the dataset into train and test sets. The data was split into training and testing with a ratio of 80 to 20. In order to maintain class proportion, we used stratified random sampling with a constant seed value.

Initially, all of the 754 features are used for the prediction. Training and validation are done using four-fold cross validation to the training data. The best train model is retrieved from the four models to be used for validation. All of the 7 trained models are retrieved and used for further evaluation. Then, hyperparameter tuning is done to all of the models. This process aims to find the best parameters that suit the models specifically based on the validation result. Evaluation in the hyperparameter tuning is done by comparing the average accuracy retrieved from the four-fold cross validation of the training dataset. Grid Search Algorithm is used for the hyperparameter tuning process. The result of the performance of each model can be seen in Table II.

The models that have validation accuracy above 80% are taken for feature selection. Genetic Algorithm is used to find the best feature combination out of those 754 features provided by Sakar’s [7] study for each of the models. This feature selection is done separately for each of the models; hence each model will have a different set of features than the others. The evaluation criteria of the GA fitness function is the mean accuracy of the four fold cross validation. The selected features are combined into a set of features for the stacked model development. Some of the features from a model may overlap with the other models. The result of the models after GA can be seen in Table III.

Utilizing the testing set, we obtained accuracy, precision, recall, and F1 score as the evaluation metrics for all of the models including the single models, ensemble models, and stacked model after before and after the hyperparameter tuning and feature selection process applied. Furthermore, the confusion matrix is also used to show the number of true positive, true negative, false positive, and false negative of the testing predictions.

5. RESULTS AND DISCUSSION

5.1 Validation Results

Table 2 depicts the hyperparameter tuning result of three single models and four ensemble models using the training dataset. The metrics that are used include: validation accuracy, validation standard deviation. Validation accuracy is used to compare the tunings. Each model is tuned in their own hyperparameter domain.

Grid search was used to find the best hyperparameters for each model based on the evaluation criteria of cross validation accuracy and standard deviation. Based on the results in table II, it can be seen that both SVM and ANN have a significantly lower performance in comparison to other models which yielded cross validation accuracies of over 80%.

Table 2: Hyperparameter Tuning Results

Model	Hyperparameters to be Tuned	Cross Validation Accuracy
Decision Tree (HT)	criterion: gini, max_depth: 3, min_samples_leaf: 5	82.12% ± 2.65
SVM (HT)	C: 0.1, gamma: 1, kernel: poly,	74.67% ± 0.29
ANN (HT)	batch_size: 80, epochs: 100	75.66% ± 3.39
Random Forest (HT)	criterion: entropy, max_depth: 70, n_estimator: 100	88.41% ± 1.78
Bagging Tree (HT)	n_estimators: 50	87.09% ± 0.57
GBT (HT)	criterion: friedman_mse, n_estimators: 70	89.24% ± 1.89
XG-Boost (HT)	booster: gbtree	87.58% ± 2.06

5.2 Testing Results

Table 2 depicts the hyperparameter tuning result of three single models and four ensemble models using the training dataset. The metrics that are used include: validation accuracy, validation standard deviation. Validation accuracy is used to compare the tunings. Each model is tuned in their own hyperparameter domain.

Table 3: Models Testing Result

Models	Accuracy	Precision	Recall	F1 Score
Decision Tree (HT)	80.3%	83.2%	92.0%	87.4%
SVM (HT)	74.3%	74.3%	100%	100%
ANN (HT)	74.3%	74.3%	100%	85.3%
Random Forest (HT)	85.5%	85.3%	97.3%	90.9%
Bagging Tree (HT)	86.8%	86.6%	97.3%	91.7%
GBT (HT)	85.5%	86.4%	95.6%	90.8%
XG-Boost (HT)	86.8%	87.8%	95.6%	91.5%
Decision Tree (HT+GA)	77.6%	82.6%	88.5%	85.5%
Random Forest (HT+GA)	85.5%	87.6%	94.7%	90.7%
Bagging Tree (HT+GA)	89.5%	89.4%	97.3%	93.2%
GBT (HT+GA)	88.8%	88.7%	97.3%	92.8%
XG-Boost (HT+GA)	88.8%	88.1%	98.2%	92.9%
Stacked Model	90.13%	88.9%	99.1%	93.7%

Even though the highest accuracy was achieved by the stacked model, it can be observed that the ensemble models consistently outperform the individual models across all evaluation metrics. This performance gap may be caused by the large amount of features in the dataset. The confusion matrix for the final stacked model result can be seen in Figure 1.

As can be seen through the confusion matrix in Fig.1, the rate of false negatives is very low at just one result in the test dataset. The low false negative rate is desirable for the purpose of this research, which includes the creation of a precursor to a preemptive self-diagnose tool before consulting with a medical professional.

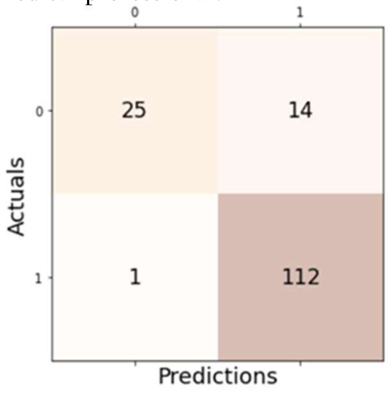

Figure 1: Confusion Matrix Of The Final Stacked Model On Test Dataset

6. CONCLUSION AND FUTURE WORKS

In this research, we have evaluated thirteen classification models resulting from hyperparameter tuning using Grid Search Algorithm, feature selection using Genetic Algorithm, and ensemble stacking. Based on the obtained results, it can be concluded that improvement in Parkinson's disease classification through speech features can be achieved by utilizing hyperparameter tuning, feature selection, and model stacking. We have achieved a better model with 90.13% accuracy using GA for feature selection and hyperparameter tuning of ensemble stacking model consisting of Decision Tree, Random Forest, Bagging Tree, Gradient Boosted Tree, and XGBoost compared to its original implementation from Sakar's study on the same dataset with 81% accuracy. Model Stacking using stacking ensemble technique is proven to be the best model compared to the other single or ensemble models which can achieve 90.13% test accuracy on Sakar's dataset.

However, since this study was only conducted using the dataset provided by a prior study, there exist some limitations regarding the implementability of the results in real world scenario. Therefore, future developments of Parkinson's disease classification may entail the enrichment of both the training and testing dataset before utilizing the model in a applicable fashion. Such as the development of a mobile application equipped with both speech processing algorithms, as well as the ensemble stacked classification model, to provide an accessible tool for Parkinson's disease classification through speech feature.

REFERENCES:

[1] "Parkinson's Disease," National Institute on Aging. http://www.nia.nih.gov/health/parkinsons-disease (accessed Feb. 04, 2022).

[2] "Statistics," Parkinson's Foundation. https://www.parkinson.org/Understanding-Parkinsons/Statistics (accessed Feb. 04, 2022).

[3] W. Wang, J. Lee, F. Harrou, en Y. Sun, "Early detection of Parkinson's disease using deep learning and machine learning", IEEE Access, vol 8, bll 147635–147646, 2020.

[4] "How Parkinson's Disease Is Diagnosed." https://www.hopkinsmedicine.org/health/treatment-tests-and-therapies/how-parkinson-disease-is-diagnosed (accessed Feb. 04, 2022).

[5] M. Little, P. Mcsharry, S. Roberts, D. Costello, en I. Moroz, "Exploiting nonlinear recurrence and fractal scaling properties for voice disorder detection", Nature Precedings, bll 1–1, 2007.

[6] B. Karan, S. S. Sahu, en K. Mahto, "Parkinson disease prediction using intrinsic mode function based features from speech signal", Biocybernetics and Biomedical Engineering, vol 40, no 1, bll 249–264, 2020.

[7] C. O. Sakar et al., "A comparative analysis of speech signal processing algorithms for Parkinson's disease classification and the use of the tunable Q-factor wavelet transform", Applied Soft Computing, vol 74, bll 255–263, 2019.

[8] R. Das, "A comparison of multiple classification methods for diagnosis of Parkinson disease", Expert Systems with Applications, vol 37, no 2, bll 1568–1572, 2010.

[9] B. E. Sakar et al., "Collection and analysis of a Parkinson speech dataset with multiple types of sound recordings", IEEE Journal of Biomedical and Health Informatics, vol 17, no 4, bll 828–834, 2013.

[10] L. Naranjo, C. J. Perez, Y. Campos-Roca, en J. Martin, "Addressing voice recording replications for Parkinson's disease detection", Expert Systems with Applications, vol 46, bll 286–292, 2016.

[11] N. Fayyazifar en N. Samadiani, "Parkinson's disease detection using ensemble techniques and genetic algorithm", in 2017 Artificial Intelligence and Signal Processing Conference (AISP), 2017, bll 162–165.

[12] B. Karan, K. Mahto, en S. S. Sahu, "Detection of Parkinson disease using variational mode decomposition of speech signal", in 2018 International Conference on Communication and Signal Processing (ICCSP), 2018, bll 0508–0512.

[13] K. Polat, "A hybrid approach to Parkinson disease classification using speech signal: the combination of smote and random forests", in 2019 Scientific Meeting on Electrical-Electronics & Biomedical Engineering and Computer Science (EBBT), 2019, bll 1–3.

[14] V. Mittal and R. K. Sharma, "Machine learning approach for classification of Parkinson disease using acoustic features", Journal of Reliable Intelligent Environments, vol 7, no 3, bll 233–239, 2021.

[15] M. S. Abhishek, C. R. Chethan, C. R. Aditya, D. Divitha, en T. R. Nagaraju, "Diagnosis of Parkinson's Disorder through Speech Data using Machine Learning Algorithms".

[16] Z. Soumaya, B. D. Taoufiq, N. Benayad, K. Yunus, en A. Abdelkrim, "The detection of Parkinson disease using the genetic algorithm and SVM classifier", Applied Acoustics, vol 171, bl 107528, 2021.

[17] R. Durgut, Y. Y. Baydilli, en M. E. Aydin, “Feature Selection with Artificial Bee Colony Algorithms for Classifying Parkinson’s Diseases' ', in International Conference on Engineering Applications of Neural Networks, 2020, bll 338–351.

[18] Y.-Y. Song en L. U. Ying, “Decision tree methods: applications for classification and prediction”, Shanghai archives of psychiatry, vol 27, no 2, bl 130, 2015.

[19] A. J. Myles, R. N. Feudale, Y. Liu, N. A. Woody, en S. D. Brown, “An introduction to decision tree modeling”, Journal of Chemometrics: A Journal of the Chemometrics Society, vol 18, no 6, bll 275–285, 2004.

[20] M. J. A. Berry, G. S. Linoff, en S. Gordon, “The Art and Science of customer relationship management”. John Wiley & Sons New York, NY, USA:, 2000.

[21] S. Shanmuganathan and S. Samarasinghe, Eds., Artificial Neural Network Modelling, vol. 628. Cham: Springer International Publishing, 2016. doi: 10.1007/978-3-319-28495-8.

[22] W. S. Noble, “What is a support vector machine?”, Nature biotechnology, vol 24, no 12, bll 1565–1567, 2006.

[23] S. Suthaharan, “Support vector machine”, in Machine learning models and algorithms for big data classification, Springer, 2016, bll 207–235.

[24] A. Widodo en B.-S. Yang, “Support vector machine in machine condition monitoring and fault diagnosis”, Mechanical systems and signal processing, vol 21, no 6, bll 2560–2574, 2007.

[25] G. Biau en E. Scornet, “A random forest guided tour”, Test, vol 25, no 2, bll 197–227, 2016.

[26] V. Svetnik, A. Liaw, C. Tong, J. C. Culberson, R. P. Sheridan, en B. P. Feuston, “Random forest: a classification and regression tool for compound classification and QSAR modeling”, Journal of chemical information and computer sciences, vol 43, no 6, bll 1947–1958, 2003.

[27] A. Paul, D. P. Mukherjee, P. Das, A. Gangopadhyay, A. R. Chintha, en S. Kundu, “Improved random forest for classification”, IEEE Transactions on Image Processing, vol 27, no 8, bll 4012–4024, 2018.

[28] L. Breiman, “Bagging predictors”, Machine learning, vol 24, no 2, bll 123–140, 1996.

[29] J. H. Friedman, “Greedy function approximation: a gradient boosting machine”, Annals of statistics, bll 1189–1232, 2001.

[30] T. Chen en C. Guestrin, “Xgboost: A scalable tree boosting system”, in Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining, 2016, bll 785–794.

[31] A. Fraser, D. Burnell, en Others, “Computer models in genetics”, Computer models in genetics., 1970.

[32] M. Kumar, M. Husain, N. Upreti, en D. Gupta, “Genetic algorithm: Review and application”, Available at SSRN 3529843, 2010.

[33] P. Dangeti, Statistics for machine learning: techniques for exploring supervised, unsupervised, and reinforcement learning models with Python and R. Birmingham, UK: Packt Publishing, 2017.

[34] K. Wittkowski, “Classification and regression trees-L. Breiman, JH Friedman, RA Olshen and CJ Stone”, Metrika, vol 33, bll 128–128, 1986.

[35] C.-C. Chang en C.-J. Lin, “LIBSVM: a library for support vector machines”, ACM transactions on intelligent systems and technology (TIST), vol 2, no 3, bll 1–27, 2011.

[36] L. Breiman, “Random forests”, Machine learning, vol 45, no 1, bll 5–32, 2001.

[37] J. Bergstra en Y. Bengio, “Random search for hyper-parameter optimization”, Journal of machine learning research, vol 13, no 2, 2012.

[38] G. Chandrashekar en F. Sahin, “A survey on feature selection methods”, Computers & Electrical Engineering, vol 40, no 1, bll 16–28, 2014.

[39] H. Frohlich, O. Chapelle, en B. Scholkopf, “Feature selection for support vector machines by means of genetic algorithm”, in Proceedings. 15th IEEE International Conference on Tools with Artificial Intelligence, 2003, bll 142–148.

[40] D. H. Wolpert, “Stacked generalization”, Neural networks, vol 5, no 2, bll 241–259, 1992.