ASEAN Engineering Journal

Full Paper

MACHINE LEARNING METHODS FOR EARLY-STAGE DIAGNOSIS OF PARKINSON'S DISEASE THROUGH HANDWRITING DATA

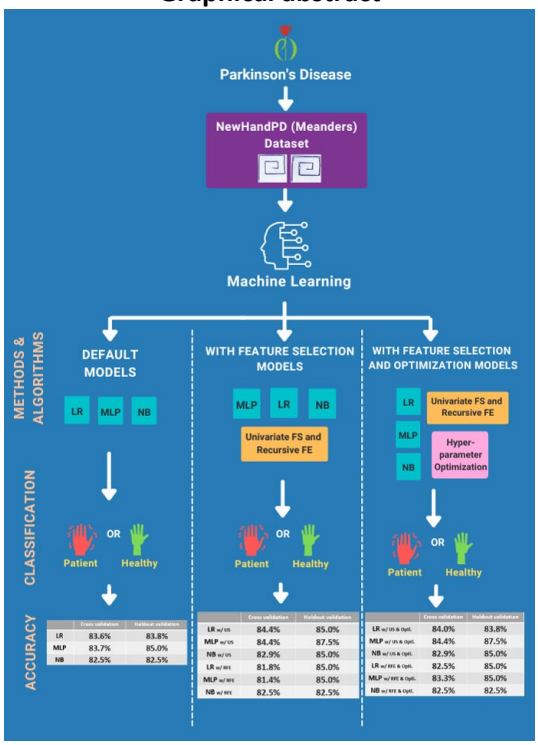
Matthew C. Dionela*, Carey Louise B. Arroyo, Mhica S. Torres, Miguel P. Alaan, Sandy C. Lauguico, Ryan Rhay Vicerra, Ronnie Concepcion II

Department of Manufacturing Engineering and Management, De La Salle University, Manila, Philippines

Article history
Received
20 June 2022
Received in revised form
25 January 2023
Accepted
29 January 2023
Published online
31 August 2023

*Corresponding author
matthew_dionela@dlsu.edu.ph

Graphical abstract



Abstract

Parkinson's disease (PD) deteriorates human cognitive and motor functions, causing slowness of movements and postural shakiness. PD is currently incurable, and managing symptoms in its late stages is difficult. PD diagnosis also has gaps in accuracy due to several clinical challenges. Thus, early-stage detection of PD through its symptoms, such as handwriting abnormality, has become a popular research area using machine learning. Since most related studies focus on advanced algorithms, this study aims to determine the classification accuracies of simpler classical models using the NewHandPD-NewMeander dataset. This study used the 9 features extracted from the meanders drawn by healthy participants and participants diagnosed with Parkinson's disease and 3 features about the individual. The same features were reduced to the 8 best according to univariate selection and recursive feature elimination. The machine learning algorithms used for the models in this study are Logistic regression, Multilayer perceptron, and Naive Bayes. Additionally, hyperparameter optimization was done. Results have shown that feature selection improved the performances of the default model, while optimization had varying effects depending on the feature selection method used. Among 15 models built, Multilayer perceptron, which utilized top 8 features from univariate selection with default hyperparameters (MLPU8), performed best. It yielded an accuracy of 84.4% in cross-validation, 87.5% in holdout validation, and an F1-score of 87.5%. Remaining models had accuracies ranging from 81.4% - 84.4% in cross-validations and 82.5% - 85.0% in holdout validations. Other studies done on diagnosing PD using similar handwritten datasets resulted in lower accuracies of 87.14% and 77.38% despite utilizing complex algorithms for its models. This proved that the 15 models built using simple architecture can outperform complex classification methods. The 15 models built accurately classify meander data and can be used as an early assessment tool for detecting PD.

Keywords: Feature selection and extraction, handwriting recognition, machine learning, Parkinson's disease diagnosis, vision-based classification

© 2023 Penerbit UTM Press. All rights reserved

1.0 INTRODUCTION

Parkinson's disease (PD) is a neurodegenerative disease that affects motor and cognitive functions of the body and progresses with age. PD is an incurable disease, and its root causes are not yet known, but research over the years paved the way to understanding its development which involves genetic and environmental factors [1–4]. Since PD is influenced by the loss of neurons that produce dopamine, symptoms of the disease on motor functions involve slowness of movements and resting tremors or postural shakiness [5, 6]. On the other hand, non-motor symptoms include olfactory and sleep disorders, loss of memory, behavioral changes, and cognitive impairment [7, 8]. The risk of having the disease is highly unpredictable with its slow progression of symptoms, making PD heavily reliant on the early stages of diagnosis [9]. However, as the number of PD

patients in the Philippines increases, accumulating 120,000 cases in 2016 according to the 5th Asian and Oceanian Parkinson's Disease and Movement Disorders Congress, experts still believe that there are inaccuracies and healthcare gaps in PD diagnoses in the country. In the hopes of increasing the accuracy of PD diagnosis, researchers continue to study biomedical markers for the disease such as human voice, gait patterns, and handwriting, since impairments in the aforementioned markers are essential and prevalent symptoms of PD [9].

One way to predict and diagnose a disease based on any of its symptoms and test its accuracy is by conducting data analysis through machine learning on the collected samples for the symptom/s. In PD early detection through voice, gait patterns, and handwriting data, commonly used classification algorithms are support vector machine (SVM), random forest, k-nearest neighbors (KNN), as well as Adaboost classifier [10–16]. The detection of Parkinson's disease heavily relies on the early stages of diagnosis, given its complexity and clinical challenges that prevent definitive diagnosis that cause difficulty at later stages in managing the symptoms [9, 17]. However, there is also difficulty in detecting PD at its early stages due to the symptoms being indistinguishable from the conditions of healthy persons [18]. As such, it is necessary to conduct studies on optimizing PD-detection accuracy through machine learning methods on any of the possible predictors of the disease.

Machine learning plays a vital role in developing automated systems for detecting PD at its early stages through handwriting data. This has gained popularity in research for its clinical benefits for the diagnosis of the disease [19]. Based on preliminary research for related literature, it was found that utilizing handwriting data as the focus for detection of PD using machine learning has less research available compared to voice data, which already has over 20 published papers. As for gait patterns, datasets are usually exclusive and difficult to acquire. Hence, aside from the lack of existing research that utilizes handwriting data, the researchers decided to focus on handwriting data since reliable datasets are available publicly [20]. Handwriting is a motor task that is being deteriorated for persons with PD, which involves control of complex movements of the wrist and fingers [21]. Distorted or smaller handwriting (micrographia) are common early signs of PD due to stiffness, shakiness, imbalances, and slowness, making handwriting an important biomarker for PD [19]. Gathering data on handwriting can be done by performing specific writing examinations on patients by means of a smart pen to record handwritten dynamics [12, 20]. Such is currently an active research area, and researchers continue to find efficient methods and systems of algorithms to implement for efficient and accurate PD detection given certain circumstances and specific applications. This type of experimentation can make use of datasets that are available online and test classification accuracy among PD patients and non-PD persons. One handwriting dataset for detecting PD was gathered from participants at the Movement Disorders Center at the First Department of Neurology, Masaryk University, and St. Annes Hospital in Brno, Czech Republic. A study made use of the said dataset and performed support vector machine (SVM) classification and obtained 80% overall accuracy [11], while another study used K-nearest neighbors (KNN) and an ensemble Adaptive Boosting (AdaBoost) classifier in addition to SVM [15]. The HandPD and NewHandPD datasets are also popular datasets that contain handwritten exam data of tracing spirals and meanders from a healthy group and patient group with PD, involving Convolutional Neural Network (CNN) in their study of the data collected [20]. Random forest and extra trees classifiers were used in another study for the NewHandPD dataset in proposing a novel cascade ensemble learning method, yielding classification performance results of 81.17% accuracy [12]. Classification was also conducted in another study on both HandPD and NewHandPD data as well as PaHaW and Parkinson's Drawing datasets by using deep transfer learning-based algorithms. Transfer learning architectures used were Alexnet, GoogleNet, VGG16 and VGG19, and ResNet50 and ResNet101 for both pre-trained and from scratch procedures, with the addition of ImageNet for the pre-trained procedure. The study was able to achieve 99.22% PD classification accuracy [19]. HandPD dataset was also used to support the proposal of another study to use a random undersampling method to balance the training process of the dataset to further improve PD detection accuracy by using a cascaded Chi-squared model with AdaBoost model. The overall accuracy of the model was 76.44%, sensitivity of 70.94%, and specificity of 81.94% [16]. Another study tested an ensemble model of the Random Forest (RF) classifier using the NewHandPD dataset and got an accuracy of 89.4%, specificity of 93.7%, sensitivity of 84.5%, and F1-score of 87.7%, performing better compared to Logistic Regression (LR) and SVM [13]. Aside from just classification algorithms, implementing feature selection methods can also aid PD-detection in early stages. A study found out that a non-nested feature selection method, using Fisher's score filter and wrapper Recursive Feature Elimination, performed best with an accuracy of 84.86% for the PaHaW dataset and 92.16% for the DraWritePD dataset, integrated with different classification training algorithms [14]. Another study performed feature selection on the PDMultiMC Handwriting dataset based on general statistical analysis such as the Shapiro-Wilk test and Mann-Whitney tests, yielding an accuracy of 96.875% using SVM classifier as a basis [22].

Despite the available research on various machine learning models aimed towards determining optimal classification accuracies of handwriting datasets for detecting PD, no study has paid attention to classification algorithms with simpler architectures. Most studies focus on those with complex architectures such as the CNN model by the researchers who collected the NewHandPD dataset [20]. This establishes the current problem of limited availability of research on classification accuracies of handwriting data for PD-detection using classification algorithms with simpler architectures. Moreover, existing research today has also not yet focused on the inclusion of any feature selection method with simple classification algorithms as well as hyperparameter optimization.

The main objective of the study can now be defined, which is to determine which among selected classical machine learning classification algorithms with simple architectures is the most accurate in classifying handwriting data, particularly meander data, gathered from two groups of individuals - healthy group and patient group - to detect Parkinson's disease. The selected machine learning algorithms for this study are Logistic Regression (LR), Multilayer Perceptron (MLP), and Naive Bayes (NB). Moreover, this study also aims to determine classification accuracies for each of the algorithms mentioned when implemented with feature selection methods and optimization of hyperparameters, as well as the implementation of feature selection only. For feature selection, univariate selection and recursive feature elimination were used as they are simpler and classical feature selection methods, so that consistency can be

met in the usage of simple methods throughout the study. The best model would be named according to its algorithm abbreviation, followed by the initial of the feature selection method and its number of features used, then an indication of optimization if applied. The methodology will make use of the NewHandPD dataset, particularly the meander dataset by Pereira et. al [20]. With this, the novelty of the paper is its focus on classical and simpler classification algorithms (LR, MLP, and NB), its exploration of univariate and recursive feature selection, as well as optimization in its built models, for the purpose of early stage diagnosis of PD thorugh handwriting meander data. The data and results of the study will contribute to the: (1) confirmation of suitability of selected machine learning algorithms, feature selection, and optimization methods with simple architectures and structures in detecting PD from handwriting data; (2) potential implementation of a machine learning system for PD-diagnosis in the health/medical industry using simple architectures given a level of accuracy; and (3) development of the research field that continuously aims to find accurate machine learning methods to detect PD at early stages.

2.0 METHODOLOGY

In this study, fifteen models were built using data extracted from meander drawings from the NewHandPD dataset available online. These models were built using three machine learning algorithms, namely, Logistic Regression, Multilayer Perceptron, and Naive Bayes. The models built may be divided into three types: default models, models with feature selection, and optimized models with feature selection. All models used data which were imported from the NewHandPD dataset and were cleaned as part of data preprocessing. This clean data was then used directly in building default models by using them in implementing the three machine learning algorithms. On the other hand, data used for the second type of models had to undergo an additional step which is feature selection. For this, two types of feature selection methods were used - Univariate Selection and Recursive Feature Elimination. Top 8 features for each feature selection method were then used to build the models categorized under models with feature selection. The last type of models built were extensions of the previous ones. Specifically, models built with feature selection had their hyperparameters optimized using Exhaustive Grid Search. As a result, models categorized under optimized models with feature selection were built. Metrics such as classification accuracy, specificity, precision, false positive rate, and F1 score were then determined and used to compare the models to each other (Figure 1). Confusion matrices were generated as well. Among all models built, the one that used MLP and Univariate Selection of top 8 features and default hyperparameters (MLPU8) performed best.

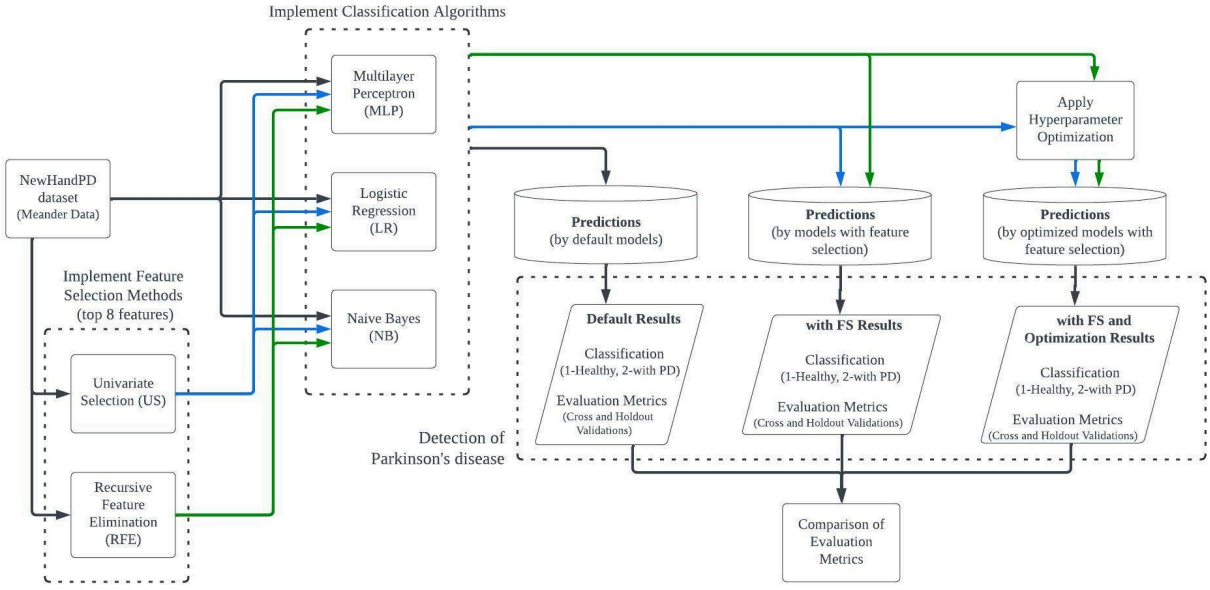


Figure 1 Developmental architecture in building models for classifying meander drawing data of healthy individuals and individuals with Parkinson's disease as a way of detecting Parkinson's disease.

2.1 Computing Platform And Libraries

Jupyter Notebook was utilized in coding and documenting the implementation of this data analysis. In line with this, the codes were written in Python 3. The group imported and used the following libraries: numpy, pandas, matplotlib, seaborn, itertools, and sklearn. Of these, pandas and sklearn were used the most. Pandas was utilized to read and handle the dataset. On the other hand, sklearn was used in model building and evaluation.

2.2 Data Acquisition

There are two available datasets from Botucatu Medical School, São Paulo State University - Brazil that may be used for handwritten Parkinson's disease detection. The first one is the HandPD dataset which is composed of handwritten exams data, from 92 individuals divided into two types of groups: 18 individuals considered as the healthy group and 74 individuals

suffering from Parkinson's disease (PD) considered as the patient group [23]. Each individual of the two groups was asked to answer a form by tracing four spirals and four meanders. Feature extraction was performed on these traced spirals and meanders which were then organized into CSV files and made available on the HandPD website.

The second dataset from the Botucatu Medical School is the NewHandPD dataset, which is a more balanced version of the HandPD dataset described earlier [20]. The NewHandPD dataset consists of the handwriting data recorded from 66 individuals which consists of a healthy group with 35 individuals, and a patient group with 31 individuals. The healthy group is composed of 18 males and 17 females aged 14 to 79 years old. Five of them are left-handed while 30 are right-handed. As for the patient group, it is made up of 21 males and 10 females aged 38 to 78 years old wherein 2 are left-handed and 29 are right-handed. Each individual in the study was tasked to perform 12 examinations involving the tracing of spirals and meanders, performing circled movements, and testing diadochokinesis. Similar to the earlier dataset, feature extraction was performed on the traced spirals and meanders. The data obtained from feature extraction were then recorded into CSV files. In this research, the group used the meander data from the NewHandPD dataset by accessing the available CSV file (NewMeander.csv) from the HandPD website. This CSV file contained the information about the image files, individuals who drew the images, and the features extracted from the collected meander images. Of this, 140 meander images were from the healthy group while 124 meander images were from the patient group (Figure 2). The NewMeander dataset consists of 16 features including exam ID, image filename, patient ID, class type, gender, dominant hand, age, and 9 other statistical quantities concerning the given exam template (ET) and the hand tracings (HT) of the individuals.

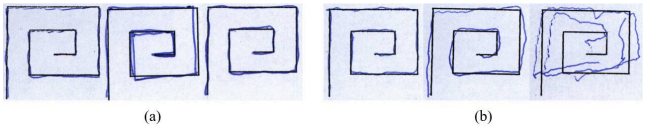


Figure 2 Sample meander images from the NewHandPD dataset collected from (a) healthy group and (b) patient group.

2.3 Data Cleaning

The NewMeander CSV file was initially imported using pandas.read_csv. Features 'ID_EXAM', 'IMAGE_NAME', 'ID_PATIENT' were then dropped as these were identifiers of the exam, image, and patient that the group deemed unnecessary for this data analysis. After dropping, the dataframe contained 264 rows (instances) and 13 columns (attributes). Two of these attributes were of datatype int64. Another two were datatype object. The remaining nine attributes were of datatype float64.

Data cleaning was performed after importing the data. It was found that there were no missing values in the dataset. Similarly, there were no duplicate entries in the dataset. For the handling of the outliers, the features 'Gender' and 'Right/Left-Handed' were dropped from the dataframe as these contained nominal data. The feature 'CLASS' was also dropped since it is the target variable and is also categorical. Afterwards, the boxplots for the remaining features were generated. The boxplots for the remaining features had circles, indicating that outliers were present (Figure 3).

To handle these, every outlier on the upper side was replaced by the upper whisker. Similarly, every outlier on the lower side was replaced by the lower whisker. To validate this method of handling, the boxplots for the features were generated once more. It could be seen that the circles were no longer present, indicating that the dataframe no longer had outliers (Figure 4).

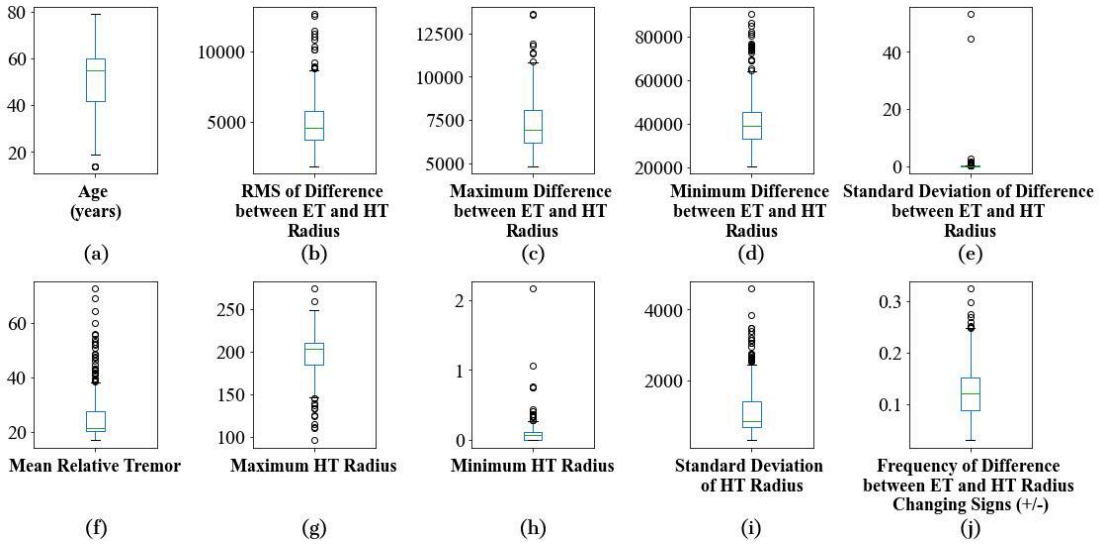


Figure 3 Initial boxplots for the features corresponding to (a) age, (b) Root Mean Square of the difference between exam template and handwritten trace radius, (c) maximum difference between exam template and handwritten trace radius, (d) minimum difference between exam template and handwritten trace radius, (e) standard deviation of the difference between exam template and handwritten trace radius, (f) mean relative tremor, (g) maximum handwritten trace radius, (h) minimum handwritten trace radius, (i) standard deviation of handwritten trace radius, and (j) number of times the difference between exam template and handwritten trace radius changes from negative to positive, or vice-versa. Outliers are represented as circles on the ends of these boxplots.

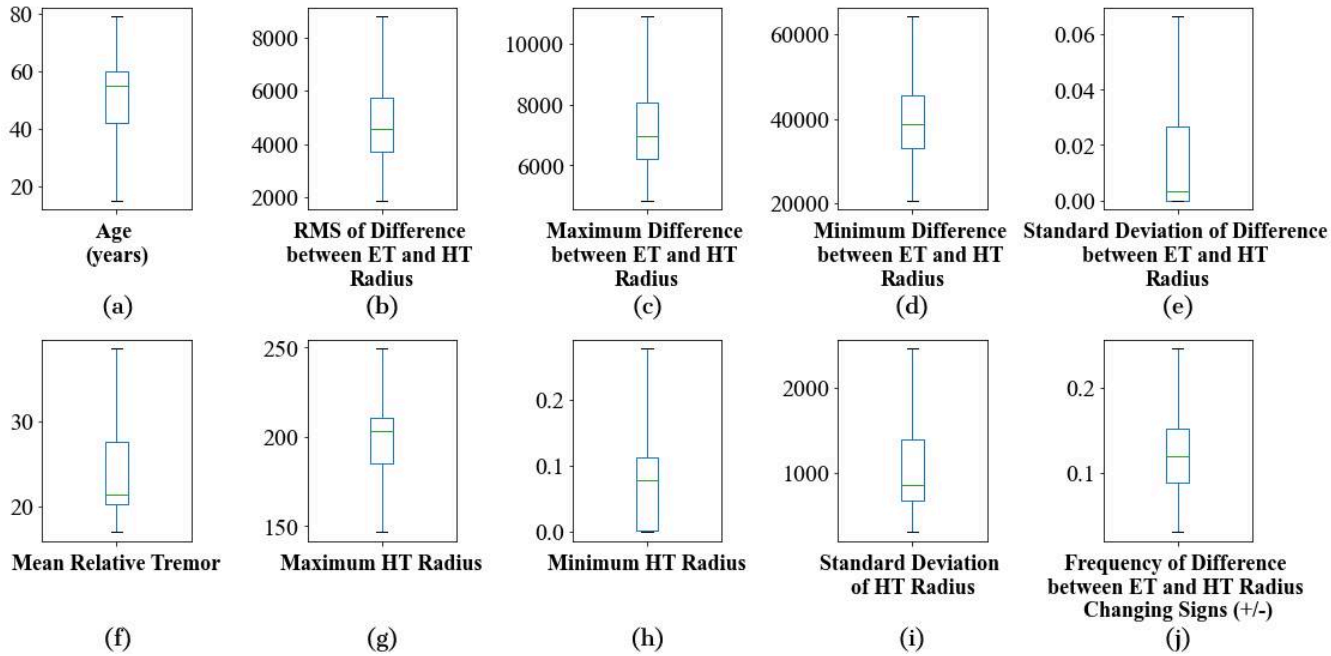


Figure 4 Boxplots after handling the outliers for the features corresponding to (a) age, (b) Root Mean Square of the difference between exam template and handwritten trace radius, (c) maximum difference between exam template and handwritten trace radius, (d) minimum difference between exam template and handwritten trace radius, (e) standard deviation of the difference between exam template and handwritten trace radius, (f) mean relative tremor, (g) maximum handwritten trace radius, (h) minimum handwritten trace radius, (i) standard deviation of handwritten trace radius, and (j) number of times the difference between exam template and handwritten trace radius changes from negative to positive, or vice-versa. The absence of circles on the boxplots indicates the absence of outliers in the cleaned dataset.

As mentioned earlier, two features were dropped as they contained nominal data. These features were 'GENDER' and 'RIGHT/LEFT-HANDED'. Under 'GENDER', data could be either F for Female or M for Male. Under 'RIGHT/LEFT-HANDED', data could be either L for left-handed or R for right-handed. Since both features only contained two categories, Label Encoding was performed to transform these nominal data into numerical forms that could be understood by the machine learning algorithms. This was done using the class LabelEncoder from the sklearn.prepocessing module. After implementation, F was 0 and M was 1 under the feature 'GENDER'. Similarly, L was 0 while R was 1 under the feature 'RIGHT/LEFT-HANDED'. This concludes the data cleaning step under data preprocessing techniques.

2.4 Exploratory Data Analysis

After cleaning the dataset, Exploratory Data Analysis (EDA) was performed. The target variable used in this project was "CLASS_TYPE" which corresponded to the classification of whether the person who drew the meander image was from the healthy or patient group. The remaining 12 variables were used as predictors. For this, the correlation heatmap was generated in order to determine the correlation of the target variable with respect to the predictors (Figure 5). Additionally, a univariate analysis was performed to determine the distribution of the variables (Figure 6).

From the heatmap, it could be seen that the predictor variables, 'STD_HT', ''MRT', and 'AGE' have positive correlations with the target variable 'CLASS_TYPE'. It follows that the standard deviation of the handwritten trace radius, mean relative tremor, and age of the individual who drew the meander image has an influence on the classification of the individual as to whether s/he has Parkinson's disease or not. Aside from these three, predictor variables 'RMS' and 'MAX_BETWEEN_ET_HT' have a moderate positive correlation with 'CLASS_TYPE'. This means that the root mean square and the maximum difference between the exam template and handwritten trace radius have some influence on the classification of the individual. Other predictors have little or negligible correlation with 'CLASS_TYPE' and can be said to have little to no influence on the classification of the individual (Figure 5).

The graphs generated for the Univariate Analysis showed that all predictors had at least one gaussian. As for the distribution behavior of these features, it could be seen that 'GENDER', 'AGE', 'RIGHT/LEFT-HANDED', and 'MAX_HT' showed a left-skewed distribution. On the other hand, the graph for 'CHANGES_FROM_NEGATIVE_TO_POSITIVE_BETWEEN_ET_HT' illustrated a moderately right-skewed distribution. The remaining predictors had graphs that illustrated right-skewed distributions (Figure 6).

2.5 Feature Standardization

Feature standardization was also done by removing the mean and scaling it to unit variance. To do this, the training and test sets were first generated using the sklearn splitter function train_test_split wherein 70% of the dataset was included in the training set while 30% was included in the testing set. Afterwards, the class StandardScaler from the sklearn.preprocessing module was used to standardize training

and test sets which apply Eq. 1 on the sets [24]. In this equation, z is the standardized value, x is a single input value, μ is the mean of the set, and s is the standard deviation of the set. This in effect would normalize the distribution of data.

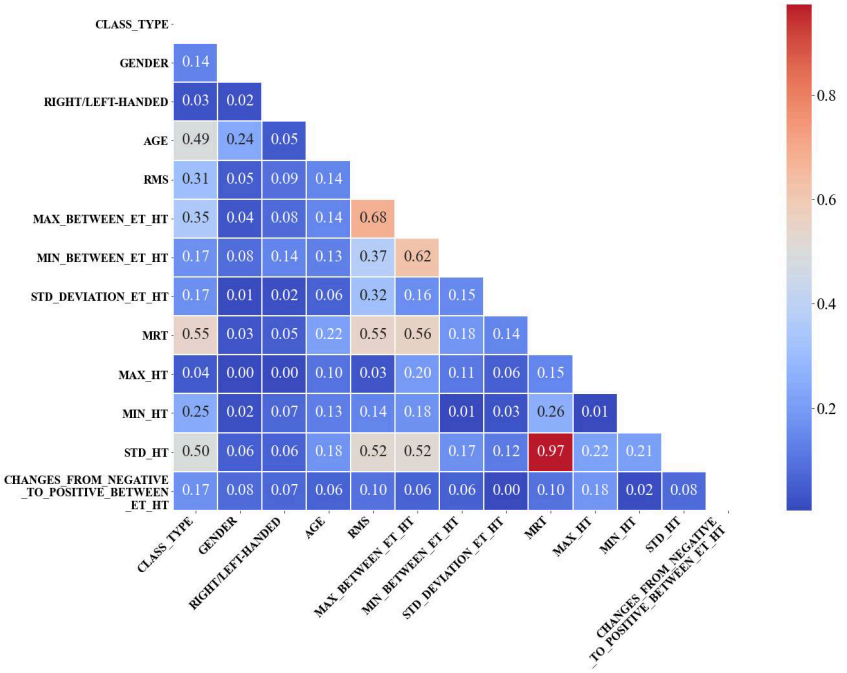


Figure 5 Heatmap that depicts the linear correlation of each feature with other features. The bluer the box is, the lower the correlation of the features indicated on its x and y-axis. Conversely, the redder the box is, the higher the correlation of the features indicated on its x and y-axis.

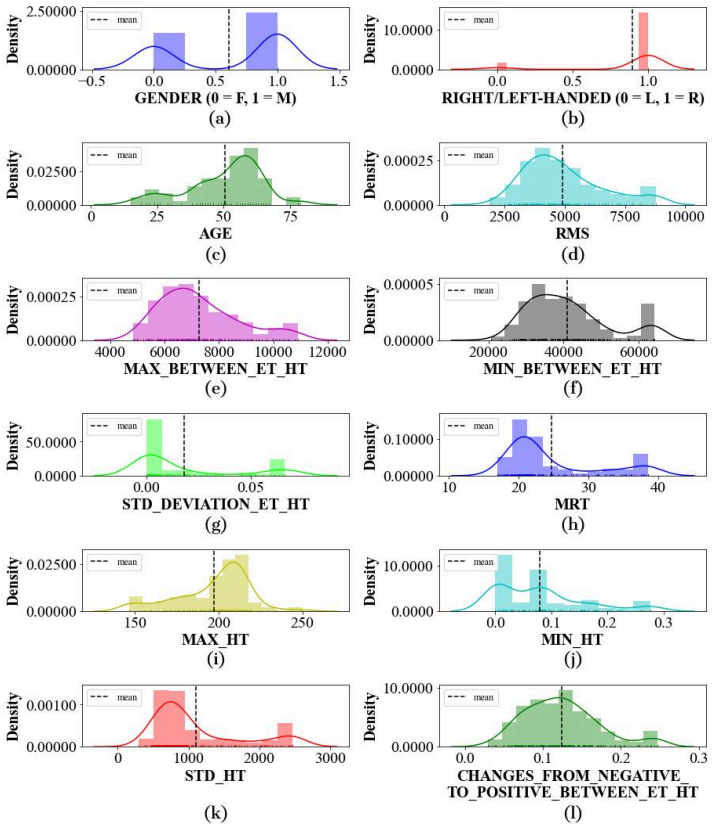


Figure 6 Distribution graphs for the features corresponding to (a) gender, (b) handedness of the individual who drew the meander image, (c) age, (d) RMS, (e) maximum difference between exam template and handwritten trace radius, (f) minimum difference between exam template and handwritten trace

radius, (g) standard deviation of the difference between exam template and handwritten trace radius, (h) mean relative tremor, (i) maximum handwritten trace radius, (j) minimum handwritten trace radius, (k) standard deviation of handwritten trace radius, and (l) number of times the difference between exam template and handwritten trace radius changes from negative to positive, or vice-versa.

$$Z = \frac{(x-\mu)}{s} \quad (1)$$

2.6 Machine Learning Algorithms

Three Machine Learning algorithms were utilized to build the models for this paper. These were Logistic Regression, Multilayer Perceptron, and Naive Bayes. Numerous research has shown that these algorithms are able to create accurate models, some even outperform other highly sophisticated classification methods [25, 26, 27, 29]. As such, these three algorithms were chosen for their simple structure, functionality, as well as their availability in the scikit learn software. For each algorithm, five models were built. One used all 12 features and default hyperparameters, two were built after feature selection, and another two were built after performing feature selection and optimizing hyperparameters.

Logistic Regression (LR) is a machine learning algorithm used for predicting binary classes. It is helpful in classification problems, which use prediction based on probability as its analysis. Logistic Regression utilizes a logistic function to model the possibilities describing the possible outcomes of a single trial as shown in Eq. 2 [25]. In this equation, P(x) is defined as the probability of the dependent variable equaling a success rather than a failure, β_0 is the intercept term, β_1 is the coefficient for the single input value x. In this paper, the class LogisticRegression from the sklearn.linear_model module was used to implement Logistic Regression in building the five models.

$$P(x) = \frac{1}{1+e^{-(\beta_0+\beta_1 x)}} \quad (2)$$

Multilayer Perceptron (MLP) is a classifier that executes classification by heavily depending on a feedforward Neural Network with the goal of approximating a nonlinear function. MLP networks are composed of many connected neurons that correspond to each layer of the network. Since it is feedforward, each neuron in an MLP network is connected to every neuron in the layer before it. Each layer can be represented as Eq. 3 [26]. In this equation, y_i is the function passed to the neuron, s is the activation function, N is the layer number, w_{ij} are its weight values, x_j are the values of the previous neurons, and b_i are the bias values. In this paper, the class MLPClassifier from the sklearn.neural_network module was used to implement the MLP in building the five models.

$$y_i = s\sum_{j=1}^{N} w_{ij}x_j + b_i \quad (3)$$

Naive Bayes (NB) is a collection of supervised learning algorithms commonly used in classification problems. It operates on the assumption that every feature in a class is conditionally independent of every other feature given the class [27]. In this paper, the class GaussianNB from the sklearn.naive_bayes module was used to implement Gaussian Naive Bayes in building the five models. Under this classifier, the likelihood of the features is assumed to be Gaussian also known as normally distributed and can be written as Eq. 4 [28]. In this equation, x is the continuous attribute, y is the class, μ_y is the mean of the class y, σ^2_y is the Bessel corrected variance of the values in x associated with class y, and P(x|y) is the probability density of x given a class y.

$$P(x|y) = \frac{1}{\sqrt{2\pi\sigma_y^2}} \exp\left(-\frac{(x-\mu_y)^2}{2\sigma_y^2}\right) \quad (4)$$

2.7 Feature Selection

To improve the default models, feature selection was done. As of writing, sklearn can only implement univariate filter selection methods and the recursive feature elimination algorithm. As such, Univariate Selection (US) and Recursive Feature Elimination (RFE) were done as implemented by [29].

Firstly, Univariate Selection (US) was performed by using the class SelectKBest from the sklearn.feature_selection module. This made it possible to select features according to the k highest scores. In implementing this feature selection method, ANOVA F-test was used for features with categorical data, specifically 'GENDER' and 'RIGHT/LEFT-HANDED'. For the remaining ten features, chi-squared test (chi2) was used as these features contained continuous data. The scores of twelve features were then used to create a bar graph to visualize the descending ranking of the features based on their scores (Figure 7). It could be seen that under Univariate Selection, descending ranking of the features were as follows: 'STD_HT', 'MIN_BETWEEN_ET_HT', 'RMS', 'MAX_BETWEEN_ET_HT', 'AGE', 'MRT', 'GENDER', 'MIN_HT', 'MAX_HT', 'STD_DEVIATION_ET_HT', 'RIGHT/LEFT-HANDED', and 'CHANGES_FROM_NEGATIVE_TO_POSITIVE_BETWEEN_ET_HT'. As done by [29], features with 0 k scores were dropped. The remaining nine features were then selected and used to build three models. Additional models using the top 8 and top 10 features were also built for comparison purposes. It was found that the highest accuracy was obtained when using the top 8 features in building the models. As such, eight attributes were selected as the final standard for the number of features used in the second feature selection method and in the latter discussions.

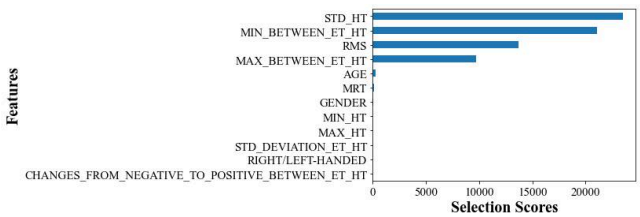


Figure 7 Descending ranking of features based on selection scores from Univariate Selection.

Recursive Feature Elimination (RFE) was implemented by using the class RFE from the sklearn.feature_selection module. With

this, features were selected by recursively considering smaller and smaller sets of features. The estimator used in implementing this was a model made using the class DecisionTreeClassifier from the sklearn.tree module. Using the class RFE, this estimator was initially trained on the initial set of features and the importance of each feature was obtained. Least important features were then removed from the features set. This process is then recursively repeated to obtain the top 8 features (Figure 8). It could be seen that under Recursive Feature Elimination, the top 8 features were as follows: 'STD_HT', 'MIN_HT', 'MAX_HT', 'MRT', 'STD_DEVIATION_ET_HT', 'MAX_BETWEEN_ET_HT', 'RMS', and 'AGE'.

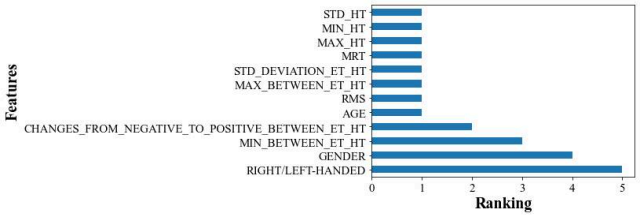


Figure 8 Descending ranking of features based on ranking results from Recursive Feature Elimination.

It was found that six features were in the top 8 rankings of both Univariate Selection and Recursive Feature Elimination. These features were 'STD_HT', 'RMS', 'MAX_BETWEEN_ET_HT', 'AGE', 'MRT', and 'MIN_HT'. The remaining features in the ranking from US were 'MIN_BETWEEN_ET_HT' and 'GENDER' while the remaining features in the ranking from RFE were 'STD_DEVIATION_ET_HT' and 'MAX_HT'. To further compare these two feature selection methods, models were built using the three machine algorithms and their classification accuracies were determined.

2.8 Optimization

Machine Learning algorithms are not always capable of predicting the behavior of data accurately. In such cases, its hyperparameters may be optimized to produce a better model with higher accuracy. Estimators have different hyperparameters which can be determined by using estimator.get_params(). However, optimizing an estimator's hyperparameters does not guarantee that it will perform better as there are some cases where the default values of the estimator already produce the best results once evaluated. With that said, there are different approaches to optimizing hyperparameters [30].

Exhaustive Grid Search or GridSearchCV is one of the methods available for optimizing hyperparameters of an estimator. This method searches for the best possible combination of hyperparameter values that would produce the highest cross-validation score. However, before performing the search, there is a need to define the parameters and their corresponding values which would be optimized. This is accomplished by creating dictionary/ies of these parameter values. Once the parameter grid has been specified, it will create all possible combinations based on the grid and evaluate each through cross-validation [30]. For this study, GridSearchCV was utilized in optimizing the hyperparameters of Logistic Regression, Multilayer Perceptron, and Naive Bayes.remaining features in the ranking from RFE were 'STD_DEVIATION_ET_HT' and 'MAX_HT'

3.0 RESULTS AND DISCUSSION

For evaluating the performance of the three machine learning algorithms, confusion matrices, cross-validation, and holdout validations were used. The confusion matrices show the number of true positive, false positive, true negative, and false-negative predictions of the estimator (Figures 9–11). Ideally, the false positive (healthy but classified as having Parkinson's disease) and false negative (has Parkinson's disease but classified as healthy) predictions are 0 since the lower the number of predictions that fall under these categories is, the better performing the estimator. As for the cross-validation, the group chose to perform k-folds cross-validation, specifically dividing the training set into ten folds before getting their mean accuracies, variance, mean f1-score, and variance in getting the target value. Lastly, for the hold-out validation, the following metrics were obtained: classification accuracy, classification error, sensitivity/recall score, specificity, false-positive rate, precision, and f1-score. The results of the cross-validation and hold-out validation for the models built were then recorded (Tables 1–3).

3.1 Performance Of Models Built With All Twelve Features And Default Hyperparameters

For each of the three machine learning algorithms mentioned earlier, one model was built using all twelve features and default hyperparameters. The Logistic Regression model yielded 12 false positives and 1 false negative with an accuracy of 83.6% in cross-validation and 83.8% in holdout validation. On the other hand, the MLP model yielded 9 false positives and 3 false negatives with an accuracy of 83.7% in cross-validation and 85% in holdout validation. Lastly, the Naive Bayes model yielded 12 false positives and 2 false negatives with an accuracy of 82.5% in both cross-validation and holdout validation.

Based on the confusion matrices created for the three machine learning algorithms and the results of the cross-validation and holdout validation conducted, the MLP model performed the best out of all the algorithms used when all 12 features and default hyperparameters are utilized. The second best in terms of classification accuracy was the Logistic Regression model, followed by the Naive Bayes model.

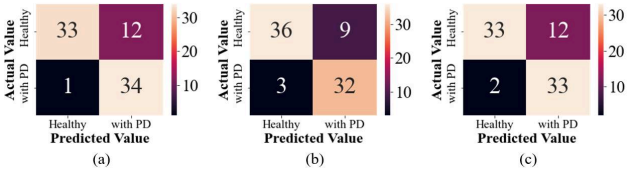


Figure 9 Confusion matrices of models from (a) Logistic Regression, (b) Multilayer Perceptron, and (c) Naive Bayes. Models were built using all twelve features and default hyperparameters set in sklearn.

Table 1 Performance of the models built with all features and default hyperparameters.

Method	Cross Validation				Holdout Validation						
	Accuracy		F1		Accuracy	Error	Recall Score	Specificity	False Positive Rate	Precision	F1 Score
	Mean	Variance	Mean	Variance							
LR	**83.6%**	8.1%	83.2%	9.4%	**83.8%**	16.3%	97.1%	73.3%	26.7%	73.9%	84.0%
MLP	**83.7%**	8.1%	83.5%	8.5%	**85.0%**	15.0%	91.4%	80.0%	20.0%	78.0%	84.2%
NB	**82.5%**	7.3%	81.2%	8.7%	**82.5%**	17.5%	94.3%	73.3%	26.7%	73.3%	82.5%

3.2 Effects Of Univariate Selection And Recursive Feature Elimination On The Performance Of The Models, Expressed In Cross And Holdout Validation Metrics

Models were also built using the top 8 features from two feature selection methods, Univariate Selection and Recursive Feature Elimination. Default hyperparameters were maintained in building these six models.

For the Logistic Regression model, Univariate Selection yielded 12 false positives and 0 false negatives with an accuracy of 84.4% in cross-validation and 85.0% in holdout validation. On the other hand, Recursive Feature Elimination yielded 12 false positives and 0 false negatives with an accuracy of 81.8% in cross-validation and 85.0% in holdout validation. Comparing the confusion matrices and results of cross-validation and holdout validation of the Logistic Regression model, which utilized only the top 8 features and default hyperparameters, to the default Logistic Regression model, which utilized all 12 features and default hyperparameters, feature selection yielded better performing models overall. After performing Univariate Selection, the number of false negatives was reduced to 0 with higher accuracy in cross-validation and in holdout validation. On the other hand, after performing Recursive Feature Elimination, there were still 12 false positives, but the number of false negatives was reduced to 0 with lower accuracy in cross-validation. However, the accuracy in holdout validation increased.

For the MLP model, Univariate Selection yielded 10 false positives and 0 false negatives with an accuracy of 84.4% in cross-validation and 87.5% in holdout validation. On the other hand, Recursive Feature Elimination yielded 11 false positives and 1 false negative with an accuracy of 81.4% in cross-validation and 85.0% in holdout validation. Comparing the confusion matrices and results of cross-validation and holdout validation of the MLP model, which utilized only the top 8 features and default hyperparameters, to the previous MLP model, which utilized all 12 features and default hyperparameters, feature selection yielded some improvement to the default models. After performing Univariate Selection, false negatives were reduced to 0 although false negatives increased by 1. Accuracies in cross-validation and in holdout validation increased. On the other hand, after performing Recursive Feature Elimination, there were more false positives but fewer false negatives with lower accuracy in cross-validation and the same accuracy in holdout validation.

For the Naive Bayes model, Univariate Selection yielded 11 false positives and 1 false negative with an accuracy of 82.9% in cross-validation and 85.0% in holdout validation. On the other hand, Recursive Feature Elimination yielded 13 false positives and 1 false negative with an accuracy of 82.5% in both cross-validation and holdout validation. Comparing the confusion matrices and results of cross-validation and holdout validation of the Naive Bayes model, which utilized only the top 8 features and default hyperparameters, to the previous Naive Bayes model, which utilized all 12 features and default hyperparameters, Univariate Selection yielded a better performing model while Recursive Feature Elimination performed the same with the default model. After performing Univariate Selection, there were fewer false positives and false negatives. The model also obtained higher accuracies in cross-validation and in holdout validation as compared with that of the default model. On the other hand, after performing Recursive Feature Elimination, there were more false positives but fewer false negatives with the same accuracy in both cross-validation and holdout validation. From these results, it was shown that feature selection improved the performance of the default models. It could also be seen that Univariate Selection yielded better models than models that used the top 8 features from Recursive Feature Elimination (Table 2). The best model in this group of models was the MLP model with Univariate Selection which had an accuracy of 84.4% in cross-validation and 87.5% in holdout validation. The second best would be the Logistic Regression model that used the top 8 features from Univariate Selection. Third best would be the Naive Bayes model that used top 8 features from Univariate Selection as well.

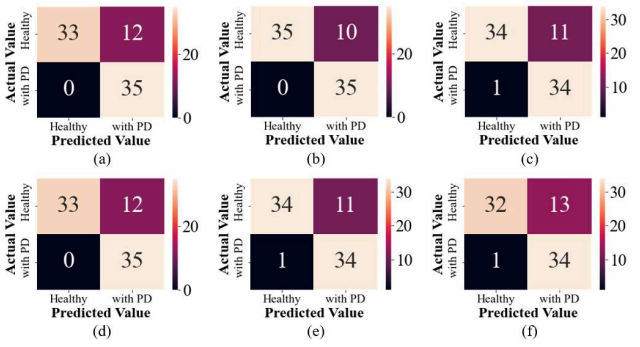


Figure 10 Confusion matrices of models from (a) Logistic Regression with US, (b) Multilayer Perceptron with US, (c) Naive Bayes with US, (d) Logistic Regression with RFE, (e) Multilayer Perceptron with RFE, and (f)

Naive Bayes with RFE. All models were built using top 8 features from the specified feature selection methods (Univariate Selection for US and Recursive Feature Elimination for RFE).

Table 2 Performance of the models using top 8 features from the two feature selection methods

Method	Cross Validation				Holdout Validation						
	Accuracy		F1		Accuracy	Error	Recall Score	Specificity	False Positive Rate	Precision	F1 Score
	Mean	Variance	Mean	Variance							
LR (with US)	**84.4%**	5.9%	84.3%	6.3%	**85.0%**	15.0%	100.0%	73.3%	26.7%	74.5%	85.4%
MLP (with US)	**84.4%**	5.6%	84.2%	6.2%	**87.5%**	12.5%	100.0%	77.8%	22.2%	77.8%	87.5%
NB (with US)	**82.9%**	6.8%	82.0%	7.5%	**85.0%**	15.0%	97.1%	75.6%	24.4%	75.6%	85.0%
LR (with RFE)	**81.8%**	7.3%	81.7%	8.0%	**85.0%**	15.0%	100.0%	73.3%	26.7%	74.5%	85.4%
MLP (with RFE)	**81.4%**	9.8%	81.3%	10.2%	**85.0%**	15.0%	97.1%	75.6%	24.4%	75.6%	85.0%
NB (with RFE)	**82.5%**	6.8%	81.3%	7.7%	**82.5%**	17.5%	97.1%	71.1%	28.9%	72.3%	82.9%

3.3 Effect Of Hyperparameter Optimization On The Performance Of The Models With Feature Selection, Expressed In Cross And Holdout Validation Metrics

To improve the performance of the three machine learning algorithms possibly further, the hyperparameters corresponding to each of the algorithms were optimized using GridSearchCV in addition to the top 8 features from Univariate Selection and Recursive Feature Elimination.

For the Logistic Regression model, Univariate Selection yielded 13 false positives and 0 false negatives with an accuracy of 84.0% in cross-validation and 83.8% in holdout validation. On the other hand, Recursive Feature Elimination yielded 12 false positives and 0 false negatives with an accuracy of 82.5% in cross-validation and 85.0% in holdout validation. In both cases, the optimized hyperparameters were used based on the results of GridSearchCV. Comparing the confusion matrices and results of cross-validation and holdout validation of the optimized Logistic Regression model, which utilized only the top 8 features and optimized hyperparameters, to the previous Logistic Regression models which used feature selection only, it was found that optimization had different results depending on the feature selection method used. In the LR model from Univariate selection, F1 score and accuracies from cross-validation and holdout validation decreased after optimizing its hyperparameters. On the other hand, optimizing the LR model from Recursive Feature Elimination improved its performance. In the optimized model, its accuracy in holdout validation and F1 score was maintained while its accuracy in cross-validation increased.

For the MLP models, Univariate Selection yielded 10 false positives and 0 false negatives with an accuracy of 84.4% in cross-validation and 87.5% in holdout validation. On the other hand, Recursive Feature Elimination yielded 12 false positives and 0 false negatives with an accuracy of 83.3% in cross-validation and 85.0% in holdout validation. In both cases, the optimized hyperparameters were used based on the results of GridSearchCV. Comparing the confusion matrices and results of cross-validation and holdout validation of the optimized MLP models, which utilized only the top 8 features and optimized hyperparameters, to the previous MLP models which used feature selection only, it was found that optimization had varying effects depending on the feature selection method used. After optimizing its hyperparameters, the performance of the MLP model from Recursive Feature Elimination improved. Specifically, its accuracy in holdout validation was maintained while its accuracy and F1 score in cross-validation increased. On the other hand, no change occurred when the MLP from Univariate Selection was optimized. Despite this, the optimized model that used top 8 features from Univariate Selection performed better than the other optimized MLP model as it had higher accuracies and F1 scores in both cross-validation and holdout validation.

For the Naive Bayes model, Univariate Selection yielded 11 false positives and 1 false negative with an accuracy of 82.9% in cross-validation and 85.0% in holdout validation. On the other hand, Recursive Feature Elimination yielded 13 false positives and 1 false negative with an accuracy of 82.5% in cross-validation and in holdout validation. In both cases, the optimized hyperparameters were used based on the results of GridSearchCV. Comparing the confusion matrices and results of cross-validation and holdout validation of the optimized Naive Bayes model, which utilized only the top 8 features and optimized hyperparameters, to the previous Naive Bayes models which used feature selection only, optimization had no effect on the performance of the models. That is, accuracies and F1 scores in cross and holdout validations were maintained for both models after optimization. This implies that the default hyperparameters were already the best hyperparameters for creating the classifier models for this particular machine learning algorithm.

It can be seen that out of the two optimized LR models, the optimized LR model that used top 8 features from Univariate Selection performed better as it had higher accuracy and F1 score in cross-validation while having a minimal difference between its values in its cross and holdout validations. As for the best optimized MLP model, it would be the optimized model that

used the top 8 features from Univariate Selection despite it maintaining the evaluation metrics prior to its optimization. This is the best optimized MLP model between the two as it had higher accuracies and F1 scores in both cross-validation and holdout validation. Lastly, for the Naive Bayes models, the optimized Naive Bayes model that used top 8 features from Univariate selection performed better than the other optimized Naive Bayes model. As such, it can be said that models that used top 8 features from Univariate Selection performed better than models that used features from Recursive Feature Elimination even after optimization.

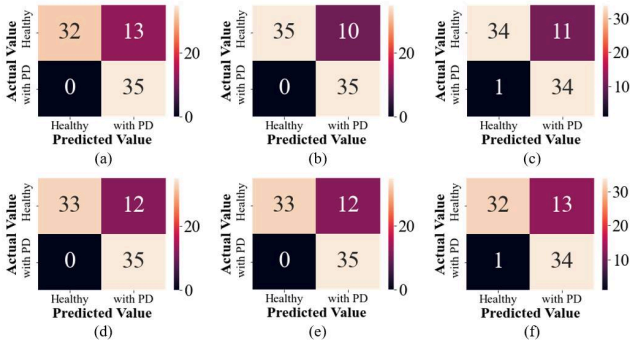


Figure 11 Confusion matrices of optimized models from (a) Logistic Regression with US, (b) Multilayer Perceptron with US, (c) Naive Bayes with US, (d) Logistic Regression with RFE, (e) Multilayer Perceptron with RFE, and (f) Naive Bayes with RFE. Models were built using top 8 features obtained from the specified feature selection methods. GridSearchCV was used to optimize hyperparameters.

3.4 Overall Performance Comparison Summary Of The Various Models Expressed In Cross And Holdout Validation Accuracies

Results showed that feature selection improved the performance of the default models. As shown in Table 3. models that used top 8 features from Univariate Selection yielded better models than those built using top 8 features from Recursive Feature Elimination. On the other hand, optimization of hyperparameters had varying effects on the models depending on the feature selection method used in building them.

Overall, the best performing model from the 15 models built would be the MLP model which used the top 8 features from Univariate Selection and default hyperparameters which had a classification accuracy of 87.5%, which the researchers call the MLPU8 model. This model can be said to have performed better than its optimized version as it obtained the same evaluation metrics as the optimized model despite costing less in terms of power and resources.

3.5 Detection Of The Presence/Absence Of Parkinson's Disease From Meander Drawings

As discussed earlier, the models were shown to accurately classify the meander image data. It follows that the health status of the individual who drew the image was correctly classified. These meander images were identified. It was found that a total of 31 meander images drawn by members of the healthy group were correctly classified during testing. Similarly, 31 meander images drawn by members of the patient group were correctly classified during testing. From this, it can be said that healthy individuals were identified from the meanders that they drew (Figure 12a). Similarly, individuals with Parkinson's disease were also identified from the meanders that they drew (Figure 12d).

Since the models built were not 100% accurate, it is a given that some meander data were incorrectly classified. From earlier discussions, it was noted that the number of meander images from the healthy group that was incorrectly classified to have been drawn by someone with Parkinson's disease ranged from 9-13 for each individual model. On the other hand, only 0-3 images drawn by someone with Parkinson's disease were incorrectly classified to have been drawn by a member of the healthy group for each model built. These incorrectly classified meander images were identified. It was found that a total of 14 meander images drawn by members of the healthy group were incorrectly classified by at least one model during testing. Of these, seven were incorrectly classified by all fifteen models during testing (Figure 12b). This means that some members of the healthy group were incorrectly classified as having Parkinson's disease. The models built may be further improved through other feature selection and optimization methods to lessen this error. An alternative take on this particular result would be that some members of the healthy group drew meanders that were similar to those drawn by patients diagnosed with Parkinson's disease. Taking it further, it may be that these individuals exhibit early symptoms of the disease. However, this cannot be verified unless these individuals are subjected to further assessments to test for Parkinson's disease. Despite this, it can still be said that there is potential in using these models in the medical field as an early assessment tool for those suspected to have Parkinson's disease.

Table 3 Performance of the optimized models using top 8 features from the two feature selection methods

Method	Cross Validation				Holdout Validation						
	Accuracy		F1		Accuracy	Error	Recall Score	Specificity	False Positive Rate	Precision	F1 Score
	Mean	Variance	Mean	Variance							
LR (with US)	**84.0%**	6.2%	84.0%	6.6%	**83.8%**	16.3%	100.0%	71.1%	28.9%	72.9%	84.3%
MLP (with US)	**84.4%**	6.1%	84.2%	6.6%	**87.5%**	12.5%	100.0%	77.8%	22.2%	77.8%	87.5%
NB (with US)	**82.9%**	6.8%	82.0%	7.5%	**85.0%**	15.0%	97.1%	75.6%	24.4%	75.6%	85.0%

LR (with RFE)	**82.5%**	7.0%	82.4%	7.3%	**85.0%**	15.0%	100.0%	73.3%	26.7%	74.5%	85.4%
MLP (with RFE)	**83.3%**	7.9%	82.9%	8.6%	**85.0%**	15.0%	100.0%	73.3%	26.7%	74.5%	85.4%
NB (with RFE)	**82.5%**	6.8%	81.3%	7.7%	**82.5%**	17.5%	97.1%	71.1%	28.9%	72.3%	82.9%

On the other hand, a total of four meander images drawn by members of the patient group were incorrectly classified by at least one model during testing (Figure 12c). The first two meanders at the top were incorrectly identified by the default MLP model which was built using all twelve features and default hyperparameters. The third placed at the bottom left was incorrectly classified by the default Naive Bayes model which also used all twelve features and default hyperparameters. The remaining image was incorrectly classified by eight of the fifteen models built. The models that incorrectly classified it were as follows: default Logistic Regression, default MLP, default Naive Bayes, Naive Bayes with Univariate Selection, Naive Bayes with Recursive Feature Elimination, MLP with Recursive Feature Elimination, optimized Naive Bayes with Univariate Selection, and optimized Naive Bayes with Recursive Feature Elimination. This incorrect classification would mean that some of the patients diagnosed with Parkinson's disease were incorrectly classified as healthy individuals. This presents a problem since this misdiagnosis in the real world may cause repercussions to the health of the individual. As such, there is still a need to further improve the models built in order to lessen this type of error.

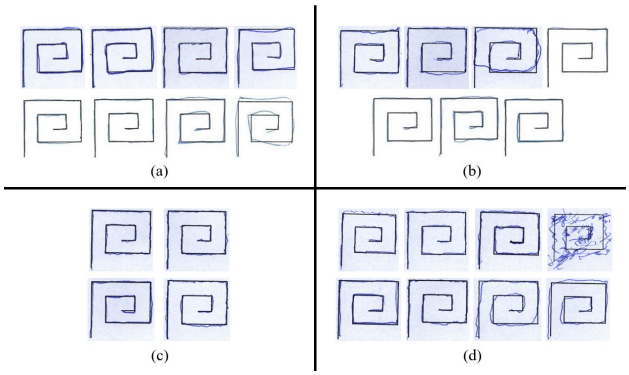


Figure 12 Sample meander images drawn from the NewHandPD dataset. (a) True negatives: Correctly classified meanders drawn by members of the healthy group. (b) False positives: Incorrectly classified meanders drawn by members of the healthy group. (c) False negatives: Incorrectly classified meanders drawn by members of the patient group. (d) True positives: Correctly classified meanders drawn by members of the patient group.

3.5 Comparison of MLPU8 with other machine learning models for Parkinson's disease detection

As shown in Table 4, other studies have also been previously done on diagnosing Parkinson's Disease using machine learning models and handwriting data.

Support vector machine (SVM) was used to builD a model for predicting Parkinsons from handwriting data in [11]. The model obtained an overall accuracy of 79.4% which is lower than the classification accuracy of the model proposed in this study, MLPU8. In [12], the best model was a novel cascade ensemble learning model that used two random forest and two extremely random trees as classifiers, as well as Principal component analysis (PCA) technique. This model only had an accuracy of 81.17%, which is also less than the classification accuracy of the best model from this study. Three classifiers for diagnosis of Parkinson's disease using handwriting data were also proposed in [15], specifically SVM, AdaBoost, and KNN. It was found that the best model was SVM with a classification accuracy of 81.3%. When compared with this SVM model, MLPU8 still had a higher classification accuracy. Similarly, the MLPU8 also had a higher classification accuracy than the cascaded learning system proposed in [16]. The system cascaded a Chi2 model with an Adaboost model, garnering a classification accuracy of 76.44%. MLPU8 also had a higher classification accuracy than the best CNN model produced by [20] which had an accuracy of 87.14%. Computer vision and machine learning was used to detect Parkinson's disease in [31]. The best fit model built for Meander data in this study was the one that used Support Vector Machines with Radial Basis Function (SVMRBF) which had a classification accuracy of 66.37%. MLPU8 still had a higher classification accuracy than this model.

The model created in this study was able to outperform the models in previous studies despite this study's simple architecture. Although some of these studies utilized a different dataset, since handwriting data was used as the basis, a relevant comparison can still be made between MLPU8 and the previously mentioned studies. This result adds to the claim that machine learning algorithms with simpler architectures can outperform other highly sophisticated classification methods [26, 27].

Table 4 Comparison of classification accuracies of the machine learning models used to detect Parkinson's Disease.

References	Machine Learning Model	Classification Accuracy (%)
[11]	SVM	79.4
[12]	Cascade ensemble with PCA	81.17
[15]	SVM	81.3
[15]	AdaBoost	78.9
[15]	KNN	71.7
[16]	Chi2-AdaBoost	76.44
[20]	CNN ImageNet	87.14
[31]	SVMRBF	66.37
This study	MLP with Univariate Selection	87.5

4.0 CONCLUSION

This study dealt with classifying Meander handwritten data using machine learning algorithms as a tool for the detection of Parkinson's disease. Logistic Regression, Multilayer Perceptron, and Naive Bayes were the algorithms used and were chosen for their simple architecture, functionality, and availability in scikit-learn. Three types of models were used: (1) default models which used all 12 available features, (2) models with feature selection which used top 8 features from either Univariate Selection or Recursive Feature Elimination, and (3) optimized models with feature selection which were the previous models but with optimized hyperparameters. Results showed that feature selection improved the performance of the models built, with models from Univariate Selection performing better than models from Recursive Feature Elimination. Optimization was found to have varying effects on the model performance depending on the feature selection method used. Among the 3 default models built, MLP model performed the best with an accuracy of 83.7% in cross-validation and 85% in holdout validation. As for the models with feature selection, the MLP model that used the top 8 features from Univariate Selection performed best out of the 6 models built for this group. It had an accuracy of 84.4% in cross-validation and 87.5% in holdout validation. For the third group of models, optimized MLP model with features from Univariate Selection also topped the list, with metrics equal to that of its non-optimized counterpart. Overall, the best model out of the 15 models built was MLPU8 which used the MLP algorithm, top 8 features from Univariate Selection, and default unoptimized hyperparameters. This model can be said to have performed better than its optimized version as it obtained the same evaluation metrics as the optimized model despite costing less in terms of power and resources. This model also performed better than the CNN model built by the creators of the dataset which had an overall accuracy of 87.14% [20]. Another study [12] that used the similar handwritten dataset had lower accuracy of 81.17% despite the application of a novel cascade ensemble learning model and Principal component analysis (PCA) technique. There are also studies that utilized meander data such as [31] which used machine learning and computer vision techniques in diagnosing Parkinson's Disease. This model had an accuracy of 66.37%. Despite these models using algorithms with more complex architecture, their accuracies were lower than the accuracy obtained from the MLP model that used the top 8 features from Univariate Selection. These results show that the models built in this study are able to accurately classify data extracted from meanders drawn by healthy individuals and Parkinson's disease patients, even outperforming models with more complex structures. As such, these models have the potential to be used as early assessment tools for those suspected to have Parkinson's disease. The group recommends for future researchers improve upon the built models and reduce the number of false negatives it yields. The group also recommends that other machine learning algorithms be explored as well as other feature selection and optimization methods in order to build better performing models.

Acknowledgement

The authors would like to express their sincerest gratitude to the Manufacturing Engineering and Management (MEM) Department of De La Salle University for providing them the opportunity to further explore their knowledge in machine learning and intelligent systems through this study.

References

[1] Tansey, M. G., Wallings, R. L., Houser, M. C., Herrick, M. K., Keating, C. E., & Joers, V. 2022. Inflammation and immune dysfunction in Parkinson disease. In *Nature Reviews Immunology.* 22(11): 657-673. DOI: 10.1038/s41577-022-00684-6

[2] Deus, C. M., Teixeira, J., Raimundo, N., Tucci, P., Borges, F., Saso, L., & Oliveira, P. J. 2022. Modulation of cellular redox environment as a novel therapeutic strategy for Parkinson's disease. In *European Journal of Clinical Investigation.* 52(10). 13820 DOI: 10.1111/eci.13820

[3] Fröhlich, F. 2016. Chapter 23 - Parkinson's Disease. In Network Neuroscience. 291-296. DOI: http://dx.doi.org/10.1016/B978-0-12-801560-5.00023-9

[4] C Korczyn, A. D., Balash, Y., and Gurevich, T. 2017. Parkinson's disease. In *International Encyclopedia of Public Health* (Second Edition). 409-415. DOI: http://dx.doi.org/10.1016/B978-0-12-803678-5.00323-4

[5] De Araújo, F. M., Cuenca-Bermejo, L., Fernández-Villalba, E., Costa, S. L., Silva, V. D. A., & Herrero, M. T. 2022. Role of Microgliosis and NLRP3 Inflammasome in Parkinson's Disease Pathogenesis and Therapy. In *Journal of Cellular and Molecular Neurobiology*. 42(5): 1283-1300. DOI: 10.1007/s10571-020-01027-6

[6] Rana, A., Dumka, A., Singh, R., Panda, M. K., Priyadarshi, N., & Twala, B. 2022. Imperative role of machine learning algorithm for detection of Parkinson's disease: Review, challenges and recommendations. In *Journal of Diagnostics*. 12(8): 2003. DOI: 10.3390/diagnostics12082003

[7] Babaei, Z. A., and Majumdar, P., "Genetic Variations Associated with Parkinson's Disease," In *The Handbook of Neuromodulation*, 1, Nova Science Publishers, Inc., 63-82, 2022.

[8] Nassan, M., and Videnovic, A. 2022. Circadian rhythms in neurodegenerative disorders. In *Nature Reviews Neurology.* 18(1): 7-24. DOI: 10.1038/s41582-021-00577-7

[9] Ayap, N. F. M., Eugenio, B. A., Hinolan, J. I. V., Puno, J. C. V., Baldovino, R. G., & Billones, R. K. C. 2021. A biomedical voice measurement diagnosis of Parkinson's disease through the utilization of artificial neural network. In *Journal of Physics: Conference Series.* 2071: 12038. DOI: http://dx.doi.org/10.1088/1742-6596/2071/1/012038

[10] Rohit Surya, A. T., Yaswanthram, P., Nair, P. R., Rajendra Prasath, S. S., & Akella, S. V. V. S. 2022. Prediction of Parkinson's Disease Using Machine Learning Models—A Classifier Analysis. In *International Conference on Advanced Computing and Intelligent Technologies*, 218: 453-460. DOI: 10.1007/978-981-16-2164-2_35

[11] Drotár, P., Mekyska, J., Smékal, Z., Rektorová, I., Masarová, L., & Faundez-Zanuy, M. 2013. Prediction potential of different handwriting tasks for diagnosis of Parkinson's. In *Proceedings of the 2013 E-Health and Bioengineering Conference.* 1-4. DOI: http://dx.doi.org/10.1109/EHB.2013.6707378

[12] Xu, S., Zhu, Z., and Pan, Z. 2020. A Cascade Ensemble Learning Model for Parkinson's Disease Diagnosis Using Handwritten Sensor Signals. In *Journal of Physics: Conference Series*. 1631: 12168. DOI: http://dx.doi.org/10.1088/1742-6596/1631/1/012168

[13] Xu, S., and Pan, Z. 2020. A novel ensemble of random forest for assisting diagnosis of Parkinson's disease on small handwritten dynamics dataset. In *International Journal of Medical Informatics*. 144: 104283. DOI: http://dx.doi.org/10.1016/j.ijmedinf.2020.104283

[14] Valla, E., Nõmm, S., Medijainen, K., Taba, P., & Toomela, A. 2022. Tremor-related feature engineering for machine learning based Parkinson's disease diagnostics. In *Biomedical Signal Processing and Control*, 75: 1746-8094. DOI: https://doi.org/10.1016/j.bspc.2022.103551

[15] Drotár, P., Mekyska, J., Rektorová, I., Masarová, L., Smékal, Z., & Faundez-Zanuy, M. 2016. Evaluation of handwriting kinematics and pressure for differential diagnosis of Parkinson's disease. In *Artificial Intelligence in Medicine*. 67: 39-46. DOI: http://dx.doi.org/10.1016/j.artmed.2016.01.004

[16] Ali, L., Zhu, C., Golilarz, N. A., Javeed, A., Zhou, M., & Liu, Y. 2019. Reliable Parkinson's Disease Detection by Analyzing Handwritten Drawings: Construction of an Unbiased Cascaded Learning System Based on Feature Selection and Adaptive Boosting Model. In *IEEE Access.* 7: 116480-116489. DOI: http://dx.doi.org/10.1109/ACCESS.2019.2932037

[17] Wolfsegger, T., Pichler, R., Assar, H., & Topakian, R. 2022. Quantitative trunk sway analysis under challenging gait conditions in early and untreated Parkinson's disease. In *Neurological Sciences.* 43(2): 1411-1413. DOI: doi:10.1007/s10072-021-05699-w

[18] Sadiq, M., Khan, M. T., and Masood, S. 2022. Attention-based deep learning model for early detection of Parkinson's disease. In *Computers, Materials and Continua*. 71: 5183-5200. DOI: http://dx.doi.org/10.32604/cmc.2022.020531

[19] Kamran, I., Naz, S., Razzak, I., and Imran, M. 2021. Handwriting dynamics assessment using deep neural network for early identification of Parkinson's disease. In *Future Generation Computer Systems*. 117: 234-244. DOI: http://dx.doi.org/10.1016/j.future.2020.11.020

[20] Pereira, C. R., Weber, S. A. T., Hook, C., Rosa G. H., & Papa, J. P. 2016. Deep Learning-Aided Parkinson's Disease Diagnosis from Handwritten

Dynamics. In 2*9th SIBGRAPI Conference on Graphics, Patterns and Images*. 340-346. DOI: http://dx.doi.org/10.1109/SIBGRAPI.2016.054

[21] Gopal, A., Hsu, W., Allen, D. D., & Bove, R. 2022. Remote assessments of hand function in neurological disorders: Systematic review. In *JMIR Rehabilitation and Assistive Technologies*. 9(1): 33157 DOI: 10.2196/33157

[22] Taleb, C., Khachab, M., Mokbel, C and Likforman-Sulem, L. 2017. Feature selection for an improved parkinson's disease identification based on handwriting. In Proceedings of the 2017 *International Workshop on Arabic Script Analysis and Recognition*. 52-56. DOI: http://dx.doi.org/10.1109/ASAR.2017.8067759.

[23] Pereira, C. R., Pereira, D. R., Da Silva, F. A., Hook, C., Weber, S. A. T., Pereira, L. A. M., & Papa, J. P. 2015. A Step Towards the Automated Diagnosis of Parkinson's Disease: Analyzing Handwriting Movements. In *2015 IEEE 28th International Symposium on Computer-Based Medical Systems*. 171-176. DOI: http://dx.doi.org/10.1109/CBMS.2015.34

[24] Alqahtani, A., Alsubai, S., Sha, M., Vilcekova, L., Javed, T. 2022. Cardiovascular Disease Detection using Ensemble Learning. In *Computational Intelligence and Neuroscience*. 2022: 5267498. DOI: 10.1155/2022/5267498

[25] Jawa, T. M. 2022. Logistic regression analysis for studying the impact of home quarantine on psychological health during COVID-19 in Saudi Arabia. In *Alexandria Engineering Journal*. 61: 7995-8005. DOI: 10.1016/j.aej.2022.01.047

[26] Bazrafashan, O., Ehteram, M., Latif, S. D., Huang, Y. F., Teo, F. Y., Ahmed, A. N., El-Shafie, A. 2022. Predicting crop yields using a new robust Bayesian averaging model based on multiple hybrid ANFIS and MLP models. In *Ain Shams Engineering Journal*. 13 (5): 101724. DOI: 10.1016/j.asej.2022.101724

[27] Ma, T. M., Yamamori, K., and Thida, A. 2020. A Comparative Approach to Naïve Bayes Classifier and Support Vector Machine for Email Spam Classification. In *Proceedings of the 2020 IEEE 9th Global Conference on Consumer Electronics*. 324-326. DOI: http://dx.doi.org/10.1109/GCCE50665.2020.9291921

[28] Ismail, S., Hassan, R. 2022. Evaluation of Naïve Bayesian Algorithms for Cyber-Attacks Detection in Wireless Sensor Networks. In *2022 IEEE World AI IoT Congress (AIIoT)*. 283-289. DOI: http://dx.doi.org/10.1109/AIIoT54504.2022.9817298.

[29] Lauguico, S.C., Concepcion, R. S., Alejandrino, J. D., Tobias, R. R., and Dadios, E. P. 2020. Lettuce life stage classification from texture attributes using machine learning estimators and feature selection processes. In *International Journal of Advances in Intelligent Informatics*. 6: 173-184.nDOI: http://dx.doi.org/10.26555/ijain.v6i2.466

[30] Lauguico, S.C., Concepcion, R. S., Alejandrino, J. D., Tobias, R. R., Macasaet, D. D., and Dadios, E. P. 2020. A Comparative Analysis of Machine Learning Algorithms Modeled from Machine Vision-Based Lettuce Growth Stage Classification in Smart Aquaponics. In *International Journal of Environmental Science and Development* 11(9): 442-449. DOI: http://dx.doi.org/10.18178/ijesd.2020.11.9.1288

[31] Pereira, C. R., Pereira, D. R., Da Silva, F. A., Masieiro, J. P., Weber, S. A. T., Hook, C., & Papa, J. 2016. A new computer vision-based approach to aid the diagnosis of Parkinson's disease. In C*omputer Methods and Programs in Biomedicine*. 136: 79-88. DOI: http://dx.doi.org/10.1016/j.cmpb.2016.08.005