

An Optimized Machine Learning Approach for Predicting Parkinson's Disease

Mousumy Kundu

Department of Computer Science and Engineering, Jashore University of Science and Technology, Jashore, 7408, Bangladesh
Email: moukunduse@gmail.com

Md Asif Nashiry, Atish Kumar Dipongkor, Shauli Sarmin Sumi and Md. Alam Hossain

Department of Computer Science and Engineering, Jashore University of Science and Technology, Jashore, 7408, Bangladesh
Email: asif.nashiry@just.edu.bd, atish.cse@just.edu.bd, shauli.sumi@gmail.com, alam@just.edu.bd

Received: 25 July 2020; Revised: 26 August 2020; Accepted: 25 October 2020; Published: 08 August 2021

Abstract: Parkinson's disease (PD) is an age-related neurodegenerative disorder affecting millions of elderly people world-wide. The early and accurate diagnosis of PD with available treatment might delay neurodegeneration and prevent disabilities. The existing diagnosis method such as brain scan is an expensive process. The use of speech recognition with machine learning technologies for the diagnosis of PD patients could be less expensive. In this work, we have worked with the voice recorded dataset from UCI machine learning repository. Several studies were performed to identify PD patients from the healthy individuals by using voice recorded data with machine learning algorithms. In this paper, we have proposed an optimized approach of data pre-processing that enhances prediction accuracy for diagnosing PD. We obtain 97.4% prediction accuracy with higher sensitivity, specificity, precision, F1 score and kappa value by using AdaBoost. These improved performance evaluation metrics indicate, the use of voice recording with our optimised machine learning approach is highly reliable in prediction of PD. This approach may have significant implications for early stage diagnosis of PD in a cost-effective manner.

Index Terms: Parkinson's disease (PD), voice recording data, machine learning models, normalization, hyperparameter tuning.

1. Introduction

Parkinson's disease (PD) is a chronic movement disorder of the nervous system. As the disease progresses, complications such as rigidity, tremor, bradykinesia and dementia arise [1]. The PD patients might encounter other complications such as imbalance, changes of voice and facial expression, lose sense of smell and have sleep disorders [2]. However, early diagnosis of PD with available treatment can slow down this neurodegenerative process. Since there is no specific biochemical test to diagnose PD, physicians diagnose PD based on the patients' medical history, signs and symptoms. The current gold-standard for diagnosis relies on clinical neurological tests and brain scans which are expensive and require high level of expertise [3]. A diagnosis method based on voice recording data of suspected patients can be utilized to facilitate early diagnosis of PD. This tactic is cost effective and can be integrated with machine learning (ML) approaches for accurate prediction. However, the major challenges of ML are to achieve higher predictive accuracy and reliability of the models. The use of several ML models with improved evaluation metrics will ensure reliability of the predictions [4, 5].

There have been several studies attempted to predict PD by using a variety of machine learning (ML) techniques. Sakar et al., used a mutual information-based method called maximum relevance minimum redundancy for feature selection [6]. The authors applied Support Vector Machine (SVM) for building a classification model with leave-one-individual-out cross validation and found 92.75% of accuracy for detecting PD. Canturk et al., applied four feature selection algorithms such as LeastAbsolute Shrinkage and Selection Operator, Relief, Local Learning-Based Feature Selection, Minimal-Redundancy-Maximal-Relevance to discover highly relevant features for classifications [7]. They applied six classifiers including Adaboost, SVM, Neural Network, Naive Bayes, K-Nearest Neighbor and Multilayer Perceptron and two validation methods i.e., k-fold and Leave-One-Subject-Out (LOSO). They obtained 68.75% accuracy using K-Nearest Neighbor. Dinesh et al., used Microsoft Azure Machine Learning Studio to analyze the data for classification and data processing [8]. They applied several ML models in their experiments and obtained the

highest accuracy (91-95%) with the Boosted Decision Tree. Abos et al., proposed an approach to discriminate the PD patients based on cognitive status with machine learning [9]. They used randomized logistic regression for feature selection and SVM model with leave-one-out cross-validation. They achieved 82.6% accuracy and found connection-wise patterns of functional connectivity for discriminating PD patients. Berus et al., exploited four feature selection techniques in [10] such as Filter-based Pearson's correlation, Kendall's correlations, Principal Component Analysis and Self-Organizing Map. They used multiple Artificial Neural Networks for classification and the results were validated by LOSO cross validation. The authors achieved 86.47% accuracy in detecting PD patients from the dataset. All these studies tried to improve the accuracy for early stage identification of PD patients from healthy individuals. Among them, Dinesh et al., 2017 achieved relatively higher prediction accuracy of 91.21% with precision and F1 scores. However, other ML evaluation matrices such as sensitivity, specificity and kappa values measures were missing from this study. Also, none of these studies considered the use of hyperparameter tuning with cross validation, in order to achieve higher reliability and accuracy of the ML model, which is getting particular attention in this field [11, 12]. In this work, we have utilized supervised ML algorithms with an existing voice recording dataset to predict the PD patient status with improved level of accuracy. We have pre-processed the dataset and applied a normalization technique. Then, we have trained the normalized dataset with hyperparameter tuning and evaluated the performance of the models.

The rest of this paper is organized as follows. Section II describes the dataset, pre-processing and normalization of data, training and hyperparameter tuning of the ML models. The experimental results are presented and discussed in Section III. Section IV concludes the entire work.

2. Materials and Methods

2.1. Dataset

The dataset is based on voice recording data for PD status prediction from University of California Irvine machine learning (UCI ML) repository [13]. This dataset is composed of a range of biomedical voice measurements from 31 people of which 23 from Parkinson's disease (PD) and 8 from healthy (NP) individuals. The columns of the dataset represent the PD patients voice measurement features with the identification and status of the patients. Each row corresponds to one of 195 voice recordings from these 31 individuals. There are nearly 6 recordings per patient, the name of the patient is identified in the first column. The value of the *status* column which is set to either NP for healthy individuals or PD for Parkinson's disease. The dataset consists of 23 features of which 22 numeric features are known as predictor features and one categorical feature (*status*) is known as response feature.

2.2. Data Pre-Processing and Normalization

In this study, we have used the R statistical environment (version 3.6.1) [14] and the 'caret' package (short for classification and regression training) to process the model training for classification problems [15]. We have replaced the value of the 'status' column to NP (healthy person) for 0 and PD (Parkinson disease) for 1. Next, we have applied a normalization technique to transform the values of the features. Without normalization, a ML algorithm might be dominated by features that have a wider range of values, which adversely affect model performance [16]. We have normalized the numerical feature values to the range between 0 and 1. The general formula for a min-max of [0, 1] is given as:

$$X = \frac{x - \min(x)}{\max(x) - \min(x)} \quad (1)$$

where x is an original value and X is the normalized value [17]. We have found the values of the features on the dataset are in different ranges. For example, the feature MDVP:Fo(Hz) contain values ranges from 88:333 to 260:105. On the other hand, feature MDVP:Jitter(Abs) contain values ranges from 7:0e-06 to 2:6e-04. For ML analysis, the feature MDVP:Fo(Hz) will intrinsically influence the result more due to its larger value. But this does not necessarily mean this particular feature is more important as a predictor. Thus, we normalize the data to bring all the values of the variables to the same range (Fig. 1). Following normalization of the data between 0 and 1, the dataset is separated into training data (80%) and testing data (20%). Both training and test data contains 75% of PD and 25% of NP patients.

	MDVP.Fo.Hz.	MDVP.Fhi.Hz.	MDVP.Flo.Hz.	MDVP.Jitter...	MDVP.Jitter.Abs.	MDVP.RAP	MDVP.PPQ	Jitter.DDP	MDVP.Shimmer				
min	0	0	0	0	0	0	0	0	0	0			
max	1	1	1	1	1	1	1	1	1	1			
	MDVP.Shimmer.dB.	Shimmer.APQ3	Shimmer.APQ5	MDVP.APQ	Shimmer.DDA	NHR	HNR	RPDE	DFA	spread1	spread2	D2	PPE
min	0	0	0	0	0	0	0	0	0	0	0	0	0
max	1	1	1	1	1	1	1	1	1	1	1	1	1

Fig.1. The normalization process that convert all the numeric features to range between 0 and 1.

2.3. Training and Tuning of the Models

We have trained the dataset with several ML models such as Multivariate Adaptive Regression Splines (MARS) [18], Support Vector Machine (SVM) with Radial Basis Function Kernel [19], Random Forest (RF) [20], Extreme Gradient Boosting (XGBoost) [21] and Adaptive Boosting (AdaBoost) [22]. Among them RF, XGBoost and AdaBoost are decision tree-based models whereas MARS is a regression based and SVM is a linear classifier. After training with different ML models, we have measured the performance of the models in terms of sensitivity, specificity, precision, recall and kappa. All these ML models were used in different studies for the prediction of PD and they achieved an accuracy between 68% to 92% [6-9]. In addition, to improve the performance of the models, we have applied hyperparameter tuning during training the models. To do hyperparameter tuning we set the tune length parameter, a control function with 10-fold cross validation and a summary function.

2.4. Evaluation Metrics

To understand the validity and reliability of the selected models, we consider various evaluation metrics of a model such as sensitivity, specificity, precision, F1 score, accuracy, and kappa values. Sensitivity (Recall or True positive rate) is the ratio of the total number of correctly classified positive values divided by the total number of positive values.

$$Sensitivity = \frac{TP}{TP + FN} \quad (2)$$

Where, TP=True Positive, FP=False Positive, FN=False Negative, TN=True Negative.

Specificity (True negative rate) is calculated as the number of correct negative predictions divided by the total number of negatives.

$$Speicificity = \frac{TN}{TN + FP} \quad (3)$$

Precision is the total number of correctly classified positive values divided by the total number of predicted positive values.

$$Precision = \frac{TP}{TP + FP} \quad (4)$$

The F1 Score is used to measure a test's accuracy. Here the two measures (Precision and Recall) help to have a measurement that represents both of them. The calculation of F1-measure which uses Harmonic Mean in place of Arithmetic Mean.

$$F1 = 2 * \frac{Precision * Recall}{Precision + Recall} \quad (5)$$

The accuracy is the ratio of the number of correct predictions to the total number of input samples.

$$Accuracy = \frac{TP + TN}{Total} \quad (6)$$

Finally, the Kappa is a value that compares an observed accuracy with an expected accuracy (random chance). The kappa statistic is used not only to evaluate a single classifier, but also to evaluate classifiers amongst themselves. In addition, it considers random chance (agreement with a random classifier), which generally means it is less misleading than simply using accuracy as a metric. If Kappa = 1, perfect agreement exists; Kappa < 0, agreement is weaker than expected by chance; Kappa close to 0, the degree of agreement is the same as would be expected by chance.

$$Kappa = \frac{TotalAccuracy - RandomAccuracy}{1 - RandomAccuracy} \quad (7)$$

3. Result & Discussion

First, we trained the models without hyperparameter tuning and predicted on test data. The comparative performance outcomes of different models for prediction of PD patients are shown in Table 1. We have considered classification accuracy, sensitivity, specificity, precision, F1 score and kappa values for each model. As it can be seen from Table 1, model XGBoost and AdaBoost have showed same result for each performance measure with accuracy of

97.4% and sensitivity of 100%. However, we have observed relatively lower prediction accuracy for other three models. We have achieved 89.5% accuracy for both MARS and SVM where we have obtained 94.7% in the case of RF. Additionally, other parameters such as sensitivity, specificity, precision, F1 score and kappa values are also relatively lower for MARS, SVM and RF models compared to XGBoost and AdaBoost.

Table 1. The Performance of the Models before Hyperparameter Tuning

Model	Accuracy	Sensitivity	Specificity	Precision	F1	Kappa
MARS	0.8947	0.931	0.7778	0.931	0.931	0.7088
SVM	0.8947	1	0.5556	0.8788	0.9355	0.6561
RF	0.9474	0.9655	0.8889	0.9655	0.9655	0.8544
XGBoost	0.974	1	0.8889	0.9667	0.983	0.9243
AdaBoost	0.974	1	0.8889	0.9667	0.9831	0.9243

Next, we have applied hyperparameter tuning with 10-fold cross validation to train all these models and observed the performance of the models. The tuning parameters for model MARS are 'nprune' and 'degree'. These two parameters represent the maximum number of terms and the maximum degree of interaction respectively where nprune = 22 and degree = 1. In the case of SVM, the tuning parameters are 'sigma' and 'C'. The values of these parameters used for the model are sigma= 0.06140776 and C = 4. The parameter for RF is 'mtry' and the optimized value we have obtained for the model when mtry is 5. The parameters for model XGBoost are 'nrounds', 'max_depth', 'eta', 'gamma', 'subsample', 'colsample_bytree', 'rate_drop', 'skip_drop' and 'min_child_weight'. The parameter 'eta' scales the contribution of each tree by a factor of $0 < \eta < 1$ when it is added to the current approximation. It is used to prevent overfitting by making the boosting process more conservative. Other parameters such as 'gamma' is the minimum loss reduction required to make a further partition on a leaf node of the tree; 'max_depth' is the maximum depth of a tree; 'min_child_weight' is the minimum sum of instance weight needed in a child; 'subsample' is the ratio of the training instance; 'colsample_bytree' is the ratio of columns when constructing each tree; 'nrounds' is the maximum number of boosting iterations; 'rate_drop' is the dropout rate and 'skip_drop' is the probability of skipping dropout. The final values of the parameters such as 'nrounds', 'max_depth', 'eta', 'gamma', 'subsample', 'colsample_bytree', 'rate_drop', 'skip_drop' and 'min_child_weight' are 100, 2, 0.4, 0, 0.75, 0.8, 0.01, 0.05 and 1, respectively. Finally, for model AdaBoost the parameters are 'niter' (number of iteration) and 'method'. The final values used for the model were niter = 50 and method = Adaboost:M1. The comparative performance of the models after hyperparameter tuning are represented in Table 2. As mention previously in Table 1, we again computed the model's performance in terms of accuracy, sensitivity, specificity, precision and F1 score. As shown in Table 2, 97.4% accuracy is achieved for MARS, AdaBoost and SVM models whereas 94.7% level of accuracy is achieved for Random Forest and XGBoost. Model AdaBoost and SVM achieved the highest sensitivity of 100% whereas the sensitivity of model MARS, RF and XGBoost is 96.5%. The specificity is achieved 88.9% for model RF, XGBoost, AdaBoost and SVM. But we obtain the highest specificity 100% for model MARS. The F1 score is between 96.6-98.2% for all the models and the kappa value is between 85-93%.

Table 2. The Performance of the Models after Hyperparameter Tuning

Model	Accuracy	Sensitivity	Specificity	Precision	F1	Kappa
MARS	0.974	0.96552	1	1	0.98246	0.93
SVM	0.974	1	0.88889	0.96667	0.98305	0.924303
RF	0.947	0.96552	0.88889	0.965517	0.96552	0.854406
XGBoost	0.947	0.96552	0.88889	0.965517	0.96552	0.854406
AdaBoost	0.974	1	0.88889	0.96667	0.98305	0.924303

From Table 1 and Table 2 we can see the difference in performances of the models before and after hyperparameter tuning. In comparison to Table 1, the accuracy of models MARS and SVM is increased from 89.47% to 97.4%. However, the accuracy of model XGBoost gets lower and the accuracy of AdaBoost and RF stay the same even after hyperparameter tuning.

In Table 3, we compare our result with other studies. In our work, we consider metrics such as accuracy, sensitivity, specificity, precision, F1 score and kappa values. However, some of these model evaluation metrics are missing in other studies.

Table 3. Comparison with Previous Studies

Author	Model	Accuracy (%)	Sensitivity (%)	Specificity (%)	Precision (%)	F1 (%)	Kappa (%)
Berus et al., 2019	ANN	86.47	88.91	84.02	*	*	*
Abos et al., 2017	SVM	82.60%	*	*	*	*	*
Dinesh et al., 2017	Boosted Decision Tree	91.21	*	*	93.57	94.23	*
Canturk et al., 2016	K-NN	68.75	70.57	66.92	*	*	*
Sakar et al., 2010	SVM	92.75%	*	*	*	*	*
This Study	MARS	97.4	96.6	100	100	98.2	93
	AdaBoost	97.4	100	88.9	96.7	98.3	92.4
	SVM	97.4	100	88.9	96.7	98.3	92.4

* value missing

By comparing the result, we achieve the higher accuracy, sensitivity, specificity, precision, F1 score and kappa value than others (Table 3). We obtain 97.4% prediction accuracy by MARS, AdaBoost and SVM model after hyperparameter tuning. The sensitivity is higher for AdaBoost and SVM model than MARS. This indicate that, both AdaBoost and SVM is better model for predicting true positive results. However, the specificity and precision are higher for MARS model in compare to AdaBoost or SVM (Table 3). This indicate that, model MARS is better for prediction false positive results.

From the above result, we can say some of the models performing better during prediction of PD and NP individuals. The way of normalization and training the models with hyperparameter tuning helps us to acquire higher performance measures for the models. We can see before hyperparameter tuning (Table 1), model AdaBoost and XGBoost have achieved the highest accuracy of 97.4%, but after hyperparameter tuning (Table 2) model MARS, AdaBoost and SVM have obtained the highest accuracy of 97.4%. By comparing the results of the models, we can say AdaBoost works better with or without hyperparameter tuning for predicting Parkinson's Disease.

4. Conclusion

In this paper, we have presented the results of evaluation of various algorithms for predative analysis of PD in voice recorded data. The algorithms are evaluated and compared with previous findings for their effectiveness in predicting PD patients from healthy individuals. It appears that all models provided >90% accuracy with improved range of ML matrices. Even with AdaBoost, MARS and SVM, we have obtained ~98% accuracy which is higher than the similar studies performed by Dinesh et al., [8]. This indicates that, our optimised machine learning approach based on voice recording is highly consistent in predicting PD from healthy individual. However, there are two major challenges remain for which this approach can be evaluated. First, the scalability of algorithms for growing data sizes [23] and second, further performance evaluation, if the features size grows. Additional optimization might require introducing feature selection with deep learning algorithms to tackle the growing data sizes.

We have pre-processed and normalized the Parkinson's dataset and have used five popular ML models with hyperparameter tuning to analyze the normalized data. The performance evaluations are demonstrated using accuracy, specificity, sensitivity, kappa, precision and F1 score. By applying these optimized models to the voice recorded data of patients will help to predict whether a person has Parkinson's disease or not at an early stage in a cost-effective manner. The future work for Parkinson's disease prediction based on voice recording will be to incorporate more data for analysis to improve the scalability and prediction accuracy of Parkinson's disease.

Acknowledgment

The authors wish to thank Dr. Shuvro Prokash Nandi, Assistant Professor, Department of Microbiology, Jagannath University, Dhaka for his helpful comments during preparation of this manuscript and formatting the manuscript.

References

- [1] J. Jankovic, "Parkinson's disease: clinical features and diagnosis," *Journal of neurology, neurosurgery & psychiatry*, vol. 79, no. 4, pp. 368-376, 2008.
- [2] S. Sveinbjornsdottir, "The clinical symptoms of Parkinson's disease," *Journal of neurochemistry*, vol. 139, pp. 318-324, 2016.
- [3] A. W. Michell, S. J. G. Lewis, T. Foltynie, and R. A. Barker, "Biomarkers and Parkinson's disease," *Brain*, vol. 127, no. 8, pp. 1693-1705, 2004.
- [4] Z. Bosnić and I. Kononenko, "An overview of advances in reliability estimation of individual predictions in machine learning," *Intelligent Data Analysis*, vol. 13, no. 2, pp. 385-401, 2009.
- [5] O. S. S. Alsharif, K. M. Elbayoudi, A. A. S. Aldrawi, and K. Akyol, "Evaluation of Different Machine Learning Methods for Caesarean Data Classification," *International Journal of Information Engineering and Electronic Business*, vol. 11, no. 5, p. 19, 2019.

- [6] C. O. Sakar and O. Kursun, "Tediagnosis of Parkinson's disease using measurements of dysphonia," *Journal of medical systems*, vol. 34, no. 4, pp. 591-599, 2010.
- [7] İ. Cantürk and F. Karabiber, "A machine learning system for the diagnosis of Parkinson's disease from speech signals and its application to multiple speech signal types," *Arabian Journal for Science and Engineering*, vol. 41, no. 12, pp. 5049-5059, 2016.
- [8] A. Dinesh and J. He, "Using machine learning to diagnose Parkinson's disease from voice recordings," 2017: IEEE, pp. 1-4.
- [9] A. Abós *et al.*, "Discriminating cognitive status in Parkinson's disease through functional connectomics and machine learning," *Scientific reports*, vol. 7, p. 45347, 2017.
- [10] L. Berus, S. Klancnik, M. Brezocnik, and M. Ficko, "Classifying Parkinson's Disease Based on Acoustic Measures Using Artificial Neural Networks," *Sensors*, vol. 19, no. 1, p. 16, 2019.
- [11] P. Schratz, J. Muenchow, E. Iturritxa, J. Richter, and A. Brenning, "Hyperparameter tuning and performance assessment of statistical and machine-learning algorithms using spatial data," *Ecological Modelling*, vol. 406, pp. 109-120, 2019.
- [12] M. Rahman, Y. Zhou, S. Wang, and J. Rogers, "Wart Treatment Decision Support Using Support Vector Machine," 2020.
- [13] M. Little, P. McSharry, E. Hunter, J. Spielman, and L. Ramig, "Suitability of dysphonia measurements for telemonitoring of Parkinson's disease," *Nature Precedings*, pp. 1-1, 2008.
- [14] R. C. Team, "R: A language and environment for statistical computing," 2013.
- [15] M. Kuhn, "A Short Introduction to the caret Package," *R Found Stat Comput*, pp. 1-10, 2015.
- [16] R. Islam, A. Satter, A. K. Dipongkor, M. S. Siddik, and K. Sakib, "A Novel Approach for Converting N-Dimensional Dataset into Two Dimensions to Improve Accuracy in Software Defect Prediction."
- [17] A. Pandey and A. Jain, "Comparative analysis of KNN algorithm using various normalization techniques," *International Journal of Computer Network and Information Security*, vol. 9, no. 11, p. 36, 2017.
- [18] J. H. Friedman, "Multivariate adaptive regression splines," *The annals of statistics*, pp. 1-67, 1991.
- [19] C. Cortes and V. Vapnik, "Support-vector networks," *Machine learning*, vol. 20, no. 3, pp. 273-297, 1995.
- [20] L. Breiman, "Random forests," *Machine learning*, vol. 45, no. 1, pp. 5-32, 2001.
- [21] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," 2016, pp. 785-794.
- [22] Y. Freund and R. E. Schapire, "A desicion-theoretic generalization of on-line learning and an application to boosting," 1995: Springer, pp. 23-37.
- [23] S.-H. Teng, "Scalable algorithms for data and network analysis," *Foundations and Trends® in Theoretical Computer Science*, vol. 12, no. 1-2, pp. 1-274, 2016.

Authors' Profiles



Mousumy Kundu, M.Sc. student, department of Computer Science and Engineering, Jashore University of Science and Technology, Jashore, Bangladesh. She is interested in Machine Learning and Bioinformatics.



Md Asif Nashiry received his PhD degree in reversible computing from the University of Lethbridge, Alberta, Canada. He is currently working as an Associate Professor in the department of Computer Science and Engineering of Jashore University of Science and Technology (JUST), Jashore, Bangladesh. His current research interests include machine learning, data science, and bioinformatics. He is the director of the machine learning and data science lab of JUST.



Atish Kumar Dipongkor is a faculty member of Computer Science and engineering at Jashore University of Science and Technology (JUST), Bangladesh. He has earned his Master of Science in Software Engineering (MSSE) from the Institute of Information Technology (IIT), University of Dhaka, Bangladesh. Before joining JUST as a lecturer, he has worked as a senior software engineer in a multinational IT organization (Brain Station 23 Ltd.). His core areas of interest are Code Smell, Refactoring, System Architecture Design, Web Technologies, and Bangla Text Processing.



Shauli Sarmin Sumi received her B.Sc degree in ICE from the Islamic University, Bangladesh and M.Sc degree in CS from the University of Lethbridge, Alberta, Canada. She is currently working as an Assistant Professor in the department of Computer Science and Engineering of Jashore University of Science and Technology, Jashore, Bangladesh. Her current research interests include data mining, natural language processing.



Dr. Md. Alam Hossain is working as an Associate Professor at the department of Computer Science & Engineering in Jashore University of Science & Technology, Bangladesh. He completed his B.Sc, M.Sc (Thesis) and PhD on Cloud Computing Security in Computer Science & Engineering from Islamic University, Bangladesh. His research interest fields are Cloud Computing, Cloud Computing Architecture, Cloud Computing Security Issues, Cloud Computing Application Services and Quality, Cyber Security, Banking Solutions Security, Network Security, Digital Forensic Science, Steganography, Information Security, Internet Security.

How to cite this paper: Mousumy Kundu, Md Asif Nashiry, Atish Kumar Dipongkor, Shauli Sarmin Sumi, Md. Alam Hossain, "An Optimized Machine Learning Approach for Predicting Parkinson's Disease ", International Journal of Modern Education and Computer Science(IJMECS), Vol.13, No.4, pp. 68-74, 2021.DOI: 10.5815/ijmecs.2021.04.06