Feature selection with artificial bee colony algorithms for classifying Parkinson's Diseases

Rafet Durgut[1][0000−0002−6891−5851], Yusuf Yargı Baydilli[1][0000−0002−4457−2081], and Mehmet Emin Aydin[2][0000−0002−4890−5648]

[1] Karabuk University, Dept. of Coputing Engineering, Karabuk, Turkey
{rafetdurgut, yusufbaydilli}@karabuk.edu.tr
[2] UWE Bristol, Dept. of Computer Science and Creative Technologies, Bristol, UK
mehmet.aydin@uwe.ac.uk

Abstract. Parkinson's is a brain disease that affects the quality of human life significantly with very slow progresses. It is known that early diagnosis is of great importance to arrange relevant and efficient treatments. Data analytics and particularly predictive approaches such as machine learning techniques can be efficiently used for earlier diagnosis. As a typical big data problem, the number of features in the collected data of Parkinson's symptoms per case matters crucially. It is known that the higher the number of features considered the more complexities incur in the handling algorithms. This leads to the dimensionality problem of datasets, which requires optimisation to overcome the trade-off between complexity and accuracy. In this study, artificial bee colony-based feature selection methods are employed in order to select the most prominent features for successful Parkinson's Disease classification over the datasets. The optimised set of features were used in training and testing *k nearest neighbourhood* algorithm, and then verified with *support vector machine* algorithm over the public dataset. This study demonstrates that binary versions of artificial bee colony algorithms can be significantly successful in feature selection in comparison to the relevant literature.

Keywords: Parkinson's Disease Classification · Speech Analysis · Feature Selection · Artificial Bee Colony.

1 Introduction

According to a report published in 2015, 6.2 million people globally suffer from Parkinson's Disease (PD), and about 177 thousand of these cases have resulted in death [1]. Although the main reasons leading to the emergence of PD are not fully known and there is no known cure, it is possible to apply some treatments to improve the symptoms observed in the patient [2]. In this way, it is aimed that the patient lives its remaining life with a relatively higher standard. From this point of view, early diagnosis of the disease is very important. The fact that the disease has a neurodegenerative structure, that is, targets motor reflexes, negatively affecting the patient's movement and mental activities, makes it possible to diagnose the disease through the tests carried out [3]. One of these tests is "speech analysis". Findings such as phonetic and speech disorders observed in PD patients, even, the onset of some deformations before the PD clinically diagnosed, reveal that this test is highly effective for early diagnosis [4]. Besides, this test is very simple and cheap. So, it provides that PD can be diagnosed by medical personnel as soon as possible [5].

In the first of the studies that developed a Computer-aided Diagnosis (CAD) model with the results obtained from these tests, Little et al. [5] tried to detect the dysphonia that occurred in Parkinson's patients through the phonations obtained from 31 patients. The authors who trained Support Vector Machine (SVM) using the uncorrelated features in the dataset, stated that they achieved 91.4% classification success. As of this date, many researchers have carried out studies to select the most suitable features in the related dataset and to increase the classification success by using different machine learning algorithms [6]. For example, Das [7] used Artificial Neural Networks (ANN) and raise the classification success rate to 92.9%. Li et al. [9] used SVM and fuzzy-based non-linear method and reported their classification success as 93.47%. In their studies, Chen et al. [10] performed feature selection and achieved 96.47% classification accuracy with the hybrid extreme learning machine. On the other hand, Zuo et al. [11] achieved 97.47% success rate using fuzzy k-Nearest Neighbor (k-NN) method improved with Particle Swarm Optimization (PSO). Finally, Gök [12] was able to increase the classification score to 98.46% by using rotation-forest ensemble k-NN.

Since speech analysis has an important place in the classification of PD, Sakar et al. [13] have brought a new dataset to the literature. The authors who examined 40 subjects (20 Parkinson's patients, 20 normal) in their study, took samples of words and sentences as well as the vowel letter 'a'. At the end of the study, they reported that vowel samples had more distinctive features than word and sentence samples. They trained a SVM using the features extracted from the samples and achieved 77.5% classification success as a result. In the other studies performed on this dataset; while Zhang et al. [14] employed ensemble learning with the multi-edit-nearest-neighbor algorithm, Abrol et al. [15] used the kernel sparse greedy algorithm. Abrol et al. was able to increase classification success up to 99.4%.

As seen in the literature, the studies mostly propose hybrid feature selection and machine learning approaches for higher success in classifications in expense of various aspects. It is observed that the high success rates achieved by the researchers seem proportional to the cross-validation methods used (e.g. Leave-one-out CV or 10-fold CV), which implies that different samples from the same cases are used for both training and validation purposes. Obviously, this is a tricky approach that has potential to undermine the real success level with respect to generalisation of the learning approaches [16]. In addition, the datasets used in the studies under-consideration contain samples from a small number of cases and need wider range of samples from larger dataset highly accurate early diagnosis of PD. Sakar et al. [17] have created a dataset consisting of 252 cases for this purpose. The dataset covers a wide range of features including the basic features, many additional discriminative features extracted from the audio signals using various techniques. The corresponding study reports a wider use of feature selection and machine learning techniques to hit the performance of 86% by SVM over 50 efficiently selected features.

The motivation of this study is to enhance the efficency and performance of classifiers, particularly *k nearest heighbourhood (k-NN)* via efficent feature selection using the varients of one of the prominent and recent swarm intelligecen approaches; *artificial bee colony (ABC)* algorithms. This is due to the attractive performance of binary versions of ABC in the recent publications. At the end of the study, the classification success was observed with competetive results and verified with SVM, too.

The rest of the paper is organised as follows: Section 2 reviews and introduces feature selection while Section 3 overviews the original and binary versions of ABC algorithm. The experimental results are provided and discussed with relevant works in Section 4 and 5, respectfully, while the study is concluded in Section 6.

2 Feature Selection

Feature selection is one very prominent areas of data analysis and machine learning, which plays crucial role in the success of the approaches with respect ot algorithmic complexity and the accuracy of the results. Big data studies emerge to particularly take care this very issue in data analysis due to the fact that the size of data tables, particularly the number of columns/attributes, hugely matters in processing. However, each attribute in a dataset may not promise significant contribution to classification success. It is a fact that excessive number of features enlarges the problem size and, subsequently, causes higher complexity on machine learning algorithms that tackle the provided classification problems. On the contrary, reduced number of features ends up with underfitting and lower accuracy in results. Hence, feature selection turns in a crucial optimisation problem in which the complexity is minimised without compromising accuracy once the the most relevant and impactful features are optimally selected. This helps chose a sub-set of all features, which provide a stable classification on the test data [18].

One of the methods used in feature selection is to develop a search strategy in the features in the dataset. In this way, it is aimed to raise the classification success to the highest possible point by selecting sub-sets from the features pool. Since brute-force searching will result in serious time and computational complexity, researchers develop strategies continuously to speed up this process through heuristic methods [19], [20]. In this study, ABC-based search strategy was implemented in 756 features of the dataset. The fitness value of each solution was calculated by the k-NN algorithm. The validation of the training phase was provided by the LOSOCV method as seen in Fig. 1.

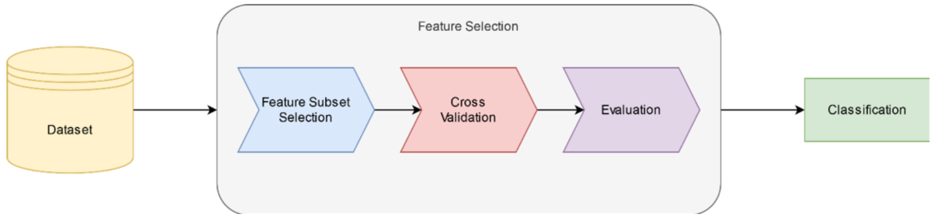

Fig. 1. Flowchart of the feature selection and classification process

3 Artificial Bee Colony (ABC) algorithms for Feature Selection

The artificial bee colony (ABC) algorithm is one of recently developed swarm intelligence approaches inspired of food search behaviors of the honey bee swarms. The original algorithm has been developed by Karaboğa [21], which imitates the collective behaviour of honey bees within their hives. The algorithm implies use of two types of bees within the hive; employed and onlooker bees. These social insects fulfil collective behaviour in three different phases as modeled into this approach, where first phase imposes each employed bee to improve its own food source, while the second phase involves each onlooker bee to look for improving the quality of its own food source. In the final phase, an exploration is initiated for new food sources by onlooker bees, subsequently transformed into scout bees, if non-adequate improvement is achieved. Further investigations and enhancements for functional optimisation problems are reported in [8].

The conceptualisation of the ABC algorithm translates the natural processes and activities into algorithmic components and functionalities, where "food source" is translated into a"feasible solution" denoted with $\mathbf{x}_i$, while "nectar amount" is recognised as the fitness of a solution denoted by $F(\mathbf{x}_i)$ as given in Eq. 1.

$$F(\mathbf{x}_i) = \begin{cases} \frac{1}{1+f(\mathbf{x}_i)} & f(\mathbf{x}_i) \geq 0 \\ 1 + |f(\mathbf{x}_i)| & \text{otherwise} \end{cases} \tag{1}$$

The probability of a particular food source to be selected through the process of ABC algorithm is calculated with Eq. 2, while a neighbouring solution such as $\mathbf{x}_n = \mathbf{x}_i + \mathbf{v}_i$ generated using Eq. 3

$$p(\mathbf{x}_i) = \frac{F(\mathbf{x}_i)}{\sum_{j=1}^{N} F(\mathbf{x}_j)} \tag{2}$$

$$\mathbf{v}_i = \mathbf{x}_i + \phi_i(\mathbf{x}_i - \mathbf{x}_n) \tag{3}$$

where $\mathbf{x}_i, \mathbf{x}_n, \mathbf{v}_i$ in the equations refer to the current solution, neighbor solution and candidate solution, respectively. ϕ_i is a randomly generated number in the scale of $[-1, 1]$. $i = 1, 2 \ldots, N$ indicates the index of the food source, where N

indicates the number of food sources. On the other hand, the scout bees can be genereted using Eq. 4 when no improvement is realised by onlooker bees.

$$x_{i,j} = LB_j + rand(0,1) \times (UB_j - LB_j) \tag{4}$$

where, $x_{i,j}$ is the j^{th} decision variable as the member of $\mathbf{x}_i$ solution vector; $j = 1, 2, .., D$ is the index, D is the total number of decision variables, LB and UB are the upper and lower boundary values defined for the decision variable.

Feature selection is a binary optimization problem, but, the ordinary ABC algorithm is developed for the continuous domains. The ABC version for solving binary problems is suggested in [22], where Eq. 3 and Eq. 4 are replaced with Eq. 5, based on Bernoulli process, as given below.

$$x_{i,j} = \begin{cases} 0 & rand < 0.5 \\ 1 & \text{otherwise} \end{cases} \tag{5}$$

The following four methods are different variants of developed ABC for binary optimization problems and used for feature selection purposes as reported below.

a. binABC (ABCv1) Algorithm has been proposed by Kiran et al. [23] imposing Eq. 6 to replace Eq. 3 in which XOR logical operator is used to produce neighbour solutions noting that the variables provided in Eq. 3 as in vector form while are in in Eq. 6 as scalar form. The parameter of ϑ is used as the logical NOT operator with which neighbour generation is applied alongside a pre-set threshold value (e.g. 0.5), if the resulted vaule is to be taken or its complement as the output value.

$$v_{i,j} = x_{i,j} \oplus \vartheta(x_{i,j} \oplus x_{n,j}) \tag{6}$$

Table 1. XOR based neighborhood operation.

Current Solution ($x_{i,j}$)	Neighbor Solution ($x_{n,j}$)	XOR Operation ($x_{i,j} \oplus x_{n,j}$)	State 1 ($\vartheta < 0.5$)	State 2 ($\vartheta \geq 0.5$)	State 1 ($v_{i,j}$)	State 2 ($v_{i,j}$)
0	0	0	1	0	1	0
0	1	1	0	1	0	1
1	0	1	0	1	1	0
1	1	0	1	0	0	1

As the procedure can be seen in Table 1, XOR oeprator is applied to current,$x_{i,j}$, and neighbor, $x_{i,j}$, solutions, then the output value is negated if $\vartheta < 0.5$, kept as is, otherwise. Afterwards, XOR is re-applied to the current solution, $x_{i,j}$ and the output value filtered with ϑ for the final output value, $v_{i,j}$.

b. disABC (ABCv2) Algorithm is proposed by Kashan et al. [24] which uses a similarity measure calculated by Eq. 7 in which the similarity of the bits in two compared solutions plays the key role. A dissimilarity measure, which names the algorithm, is subsequently calculated for this version of the algorithm in order to be used for neighbour solution generation. As the approach imposes, a new solution, was generated by Eq. 4 previously for non-binary problems, is now replaced with Eq. 7, which calculates Jaccard's similarity constant together with Eq. 8.

$$sim(x_i, x_j) = \frac{M_{11}}{M_{01} + M_{10} + M_{11}} \tag{7}$$

$$dissim(x_i, x_j) = 1 - sim(x_i, x_j) \tag{8}$$

where M_{11} is the number of 1 bits in both x_i and x_n at the same positions, while M_{01} and M_{10} are determined, accordingly. Eq. 9 declares that the dissimilarity of the current solution with the neighbour-to-be is an approximate of the dissimilarity of two existing solutions, x_i and x_j, normalised with ϕ while Eq. 10 presents a minimisation model with a number of constraints, which imples that the new solution to-be, v_i, is expected to satisfy the constraints and let the objective function be minimum.

$$dissim(v_i, x_i) \approx \phi \times dissim(x_i, x_j) \tag{9}$$

$$\min \ |dissim(v_i, x_i) - \phi \times dissim(x_i, x_j)| \tag{10}$$

Subject to:

$$\begin{aligned} M_{11} + M_{01} &= n_1 \\ M_{10} &\leq n_0 \\ \{M_{10}, M_{11}, M_{01}\} &\geq 0 \text{ and } \in \mathbb{Z} \end{aligned}$$

where ϕ is a random positive value, n_1 and n_0 represent the number of bits with a value of 1 and 0 in the current solution, x_i . The aim in here is to determine the closest possible minimum value according to the difference between the candidate solution and the current solution. Detailed information and examples can be found in [24].

c. Improved binABC (ABCv3) ABC algorithm updates the value of only one decision variable among D number of decision variables per iteration, while various other swarm intelligence algorithms propose updating multiple variables within the complete vector of decision variables. Obviously, there is a trade-off between exploration and exploitation balance to handle while attemting the updates.

This binary version of the ABC, as discussed in [26], attempts to balance exploration and exploitation with an exponantially calculated rate, d_t as in Eq.11.

$$d_t = rand(0, \alpha) + e^{-(\frac{t}{t_{max}}) \times 0.1 \times D} + 1 \tag{11}$$

where, the α is randomly determined perturbation number, D is the problem dimension, number of decison variables, and t and t_{max} indicate the current and maximum number of iterations, respectively. It is worthwhile to note that d_t will decrease with growing t, which means that the exploration is higher in earlier iterations while exploitation gets stronger in later iterations. That is believed to help keep the balance explained above.

The neighbourhood operator in Eq.6 proposed by Kıran et al. [23] is revised to be used in this version due to the fact that ϑ is originally setup with 0.5 in Eq.6 as the exploitation factor. This pre-set fixed threshold weakens the exploitation as it involves a more random process. Eq.12 proposes a new way to determine ϑ. This rule allocates 0 to ϑ if the new solution is worse, otherwise, it is updated depending on the iteration.

$$\vartheta = \begin{cases} Q_{max} - (\frac{Q_{max}-Q_{min}}{t_{max}}) \times t & F(x_n) < F(x_i) \\ 0 & \text{otherwise} \end{cases} \tag{12}$$

where Q_{max} and Q_{min} represent the upper and lower limits of the defined range, respectively [25].

d. NBABC (ABCv4) In this binary ABC (NBABC) version that proposed by Santana et al. [26], it is ensured the influencing of the specified number of decision variables during the implementation of the neighborhood operator. Trough the *max_dim* parameter in the algorithm, the maximum number of dimension values is determined in each iteration. The pseudo code of the neighborhood operator is as follows:

Algorithm 1: NBABC Algorithm

Input: $\mathbf{x}_i$
Select $\boldsymbol{x}_j$ *where* $i \neq j$ /* A new Food Source */
Set *The number of selected dimensions* ($max\_dim \times D$)
Select *random dimensions for the food source* (***Dims***)
Foreach $d \in Dims$ **do**
    if $x_{i,d} = x_{j,d}$ **then**
        $v_{i,d} = x_{i,d}$
    end
    else
        $v_{i,d} = x_{j,d}$
    end
Return $\boldsymbol{v}_i$

4 Experimental Results

The following experimental study has been fulfilled to test the algorithms underconsideration for feature selection purposes. The dataset created by Sakar et al. [17] has been widely used. This dataset contains samples from 188 Parkinson's patients and 64 healthy cases. While constructing the dataset, the voice of patients for the vowel 'a' was recorded 3 times from each case, hence, 252 x 3 = 756 audio signals were obtained. In addition to the baseline features of audio signals, feature extraction was made through many techniques (e.g. Time Frequency Measures, Mel Frequency Cepstral Coefficients, Wavelet Transform Based Features, Vocal Fold Features, Tunable Q-Factor Wavelet Transform Based Features). Thus, they were able to create a dataset including a total of 754 feature vectors.

The algorithmic hyper parameters considered through out of these experimentations have been tabulated in Table 2 per algorithm.

Table 2. Control parameters of algorithms

	ABCv1	ABCv2	ABCv3	ABCv4
Population Size	20	20	20	20
Max number of Function Evaluation	1000	1000	1000	1000
max_dim	-	-	-	0.1
ϕ_{max}/Q_{max}	-	0.9	0.3	-
ϕ_{min}/Q_{min}	-	0.1	0.1	-
Limit	100	100	100	100

Table 3 shows the success metrics on each feature category. Since the Bandwidth features in the dataset are few, all methods achieved the best results. When the algorithms run using the TQWT features that have the highest number of features, ABCv2, ABCv3 and ABCv4 showed the best performance according to calculated mean and maximum success values. Although ABCv2 performed the best on the Wavelet Transform features according to the mean results, ABCv3 and ABCv4 achieved the maximum success value. For MFCC features, ABCv4 algorithm achieved the highest scores for both mean and maximum results. Finally, ABCv2 performed better than other methods for Baseline and Vocal Fold features.

From this part of the study, it can be concluded that the MFCC features contain the most qualified features to distinguish between classes. Therefore, all four algorithms showed the best performance on these features. Another important point is that although the number of these features is 84, the algorithms have achieved the highest results with an average of 40.

In the second stage of the study, classification was performed with SVM using the features selected for k-NN algorithm. That was to verify how robust was ABC-based feature selection. As can be seen from Table 4, the features selected by ABCv3 algorithm gave the best results according to the maximum success criteria. However, according to the mean success criteria, ABCv2 and ABCv4 were able to compete with ABCv3.

The fact that SVM algorithm, which is trained with 35 features selected by ABCv3 algorithm, produces higher results compared to k-NN, shows that SVM algorithm is a more successful classification algorithm than k-NN. On the other hand, it can be concluded that the features selected by the binary ABC algorithms create a reasonable and fair benchmark environment for classification algorithms. Moreover, it is possible to achieve higher successes by using SVM instead of k-NN while calculating fitness values in the feature selection stage of ABC algorithms.

Table 3. Feature selection with k-NN results.

		Baseline (26 Features)			Bandwidth (8 Features)			Vocal Fold (22 Features)			MFCC (84 Features)			WT Applied to F0 (182 Features)			TQWT (432 Features)		
		ACC	**F1**	**MCC**	**ACC**	**F1**	**MCC**	**ACC**	**F1**	**MCC**	**ACC**	**F1**	**MCC**	**ACC**	**F1**	**MCC**	**ACC**	**F1**	**MCC**
ABCv1	*Mean*	0.79	0.87	0.39	**0.77**	**0.86**	**0.30**	0.80	0.88	0.41	0.85	0.90	0.56	0.77	0.85	0.31	0.84	0.90	0.55
	Max	0.83	0.89	0.49	**0.77**	**0.86**	**0.30**	0.81	0.89	0.45	0.87	0.92	0.63	0.79	0.87	0.37	0.86	0.91	0.59
ABCv2	*Mean*	**0.82**	**0.89**	**0.48**	**0.77**	**0.86**	**0.30**	**0.82**	**0.89**	**0.46**	0.85	0.91	0.58	**0.79**	**0.87**	**0.37**	**0.85**	**0.91**	**0.58**
	Max	0.83	0.89	0.49	**0.77**	**0.86**	**0.30**	0.83	0.89	0.49	0.86	0.91	0.61	0.79	0.87	0.39	0.86	0.91	0.61
ABCv3	*Mean*	0.80	0.87	0.40	**0.77**	**0.86**	**0.30**	0.80	0.88	0.41	0.85	0.91	0.58	0.78	0.86	0.34	**0.85**	**0.91**	**0.58**
	Max	0.83	0.89	0.49	**0.77**	**0.86**	**0.30**	0.82	0.89	0.48	0.87	0.92	0.62	0.80	0.88	0.42	0.86	0.91	0.60
ABCv4	*Mean*	0.80	0.88	0.42	**0.77**	**0.86**	**0.30**	0.81	0.88	0.44	**0.86**	**0.91**	**0.59**	0.78	0.86	0.36	**0.85**	**0.91**	**0.59**
	Max	0.83	0.89	0.49	**0.77**	**0.86**	**0.30**	0.82	0.89	0.48	0.87	0.92	0.64	0.80	0.87	0.40	0.87	0.92	0.63

Table 4. SVM results for selected features

		Baseline (26 Features)			Bandwidth (8 Features)			Vocal Fold (22 Features)			MFCC (84 Features)			WT Applied to F0 (182 Features)			TQWT (432 Features)		
		ACC	**F1**	**MCC**	**ACC**	**F1**	**MCC**	**ACC**	**F1**	**MCC**	**ACC**	**F1**	**MCC**	**ACC**	**F1**	**MCC**	**ACC**	**F1**	**MCC**
ABCv1	*Mean*	0.79	0.87	0.35	**0.77**	**0.86**	**0.29**	0.76	0.86	0.27	0.82	0.89	0.50	0.76	0.86	0.24	0.82	0.89	0.49
	Max	0.82	0.89	0.46	**0.77**	**0.86**	**0.29**	0.78	0.87	0.34	0.85	0.91	0.59	0.77	0.86	0.29	0.85	0.90	0.56
ABCv2	*Mean*	0.76	0.85	0.28	**0.77**	**0.86**	**0.29**	0.73	0.83	0.18	**0.84**	**0.90**	**0.54**	**0.77**	**0.86**	**0.25**	**0.83**	**0.89**	**0.51**
	Max	0.76	0.85	0.28	**0.77**	**0.86**	**0.29**	0.73	0.83	0.18	0.84	0.90	0.54	0.77	0.86	0.25	0.83	0.89	0.51
ABCv3	*Mean*	**0.79**	**0.87**	**0.37**	**0.77**	**0.86**	**0.29**	0.76	0.85	0.26	0.83	0.89	0.51	0.76	0.86	0.24	0.82	0.89	0.48
	Max	0.83	0.90	0.51	**0.77**	**0.86**	**0.29**	0.79	0.87	0.38	0.88	0.92	0.66	0.78	0.87	0.32	0.84	0.90	0.55
ABCv4	*Mean*	0.79	0.87	0.36	**0.77**	**0.86**	**0.29**	**0.77**	**0.86**	**0.30**	0.83	0.89	0.52	0.76	0.86	0.24	0.82	0.89	0.48
	Max	0.82	0.89	0.48	**0.77**	**0.86**	**0.29**	0.79	0.87	0.36	0.86	0.91	0.60	0.77	0.86	0.29	0.84	0.90	0.55

5 Related Works and Discussions

Since the dataset used in this study has been made public, several studies have been carried out to improve the classification successes. For example, Altay and Atlas [27] used two different evolutionary algorithms in their work. Badem et al. [28] employed ABC algorithm for feature selection. Tuncer and Dogan [29] performed feature extraction with Singular Value Decomposition (SVD) and feature selection with Neighborhood Component Analysis (NCA). In the classification phase, they used SVM. Castro et al. [30] provided classification with ANN. Finally, Tuncer et al. [31] extracted features from the dataset using minimum average maximum tree and SVD, and then performed classification with k-NN. It was observed that some of their results remian better in accuracy than our score. However, with closer look into the details, it was seen that the 10-fold CV method was used during the training and validation phase. But, as mentioned in the work by Sakar et al. [17], using this kind of cross-validation cannot provide generalization for all subjects and causes in biased results. Therefore, it is found unfair to make a comparison between the results in [31] and our study's. For this reason, our results are evaluated with the results obtained in the original study, namely the study using LOSOCV (Leave-One-Subject-Out Cross Validation). The relevant comparison is as in Table 5 below.

Table 5. Comparison of the results with the original study

	Feature Selection Algorithm	# of Features	Classification Algorithm	Acc. (mean)	Acc. (max)
Sakar et al	mRMR	50	SVM	-	0.84
This study	ABCv2	39	k-NN	0.86	0.87
	ABCv4	35	SVM	0.83	0.88

In general, it is observed that the MFCC features in the Parkinson dataset are the best definitive and the most discriminative features for classification. In this respect, it can be concluded that the MFCC features will provide more stable and successful results in more comprehensive Parkinson datasets to be created in the future. However, as seen in the original study, the classification success rate achieved using only MFCC features remained at the level of 0.84. This leads to the question of whether or not more successful results can be obtained through feature selection algorithms.

In their study, Sakar et al. [17] stated that the highest score was obtained with an SVM trained with 50 features selected from all features (756). However, as can be seen in Tab. 5, when evaluated on the basis of mean values, it can be observed that the NBABC (ABCv4) algorithm can achieve more successful results with less features (0.86). When the algorithms were evaluated in terms of the maximum success rate, all algorithms except disABC (ABCv2) produced the highest accuracy value (0.87). Moreover, if SVM is trained with the selected features, it has been observed that the success rates can be increased a little more. Accordingly, improved binABC (ABCv3) algorithm obtained the highest success value (0.88). As a result, it can be said that thanks to the feature selection to be applied in Parkinson datasets, higher successes can be achieved and binary ABC methods are highly capable in this task.

6 Conclusions

The datasets of Parkinson's sympthoms collated provide great support to study data-driven computational approaches if they are helpful in diagnosis of the disease. The dimensionality problem of such datasets has to be eased before devicing automatic methods to help medical staff. One of the collated datasets are of sound samples they receive from patients through signal processing techniques to apply speech analysis, which help understand any anomlies detected. Machine learning techniques are prominantly used in predictive analysis of the data for diagnosis purposes. However, the dimansionality problem matters and the number of features has to be studied and optimsed accordingly.

In this study, the variants of binary ABC algorithms have been studied for feature selection and dimensionality problem of the datasets underconsideration, which includes 756 features. In the first stage of the study, feature selection was made using four different binary ABC algorithms, and the success values of the selected features were evaluated with the k-NN algorithm. In the second stage of the study, SVMs were trained with the selected features and it was seen that the selected features could increase the classification success compared to the original study. Thus, it has been shown that effective results can be achieved in PD classification by the methods used in the study.

References

1. H. Wang et al., "Global, regional, and national life expectancy, all-cause mortality, and cause-specific mortality for 249 causes of death, 1980–2015: a systematic analysis for the Global Burden of Disease Study 2015", The Lancet, 388(10053), 1459–1544, Oct. 2016.
2. W. H. Oertel, "Recent advances in treating Parkinson's disease," F1000Research, 6, Mar. 2017.
3. J. Massano and K. P. Bhatia, "Clinical approach to Parkinson's disease: Features, diagnosis, and principles of management," Cold Spring Harbor Perspectives in Medicine, 2(6), Jun. 2012.
4. M. A. Little, P. E. McSharry, S. J. Roberts, D. A. Costello, and I. M. Moroz, "Exploiting nonlinear recurrence and fractal scaling properties for voice disorder detection," BioMedical Engineering OnLine, 6(1), 1–19, Dec. 2007.
5. M. A. Little, P. E. McSharry, E. J. Hunter, J. Spielman, and L. O. Ramig, "Suitability of dysphonia measurements for telemonitoring of Parkinson's disease," IEEE Transactions on Biomedical Engineering, 56(4), 1015–1022, Apr. 2009.
6. Z. Cai et al., "An intelligent Parkinson's disease diagnostic system based on a chaotic bacterial foraging optimization enhanced fuzzy knn approach," Computational and Mathematical Methods in Medicine, 2018, 1–24, Jun. 2018.
7. R. Das, "A comparison of multiple classification methods for diagnosis of Parkinson's disease," Expert Systems with Applications, 37(2), 1568–1572, Mar. 2010.
8. M. Düğenci, and M.E. Aydin, "A honeybees-inspired heuristic algorithm for numerical optimisation", Neural Computing and Applications, 1-15, 2019.
9. D.-C. Li, C.-W. Liu, and S. C. Hu, "A fuzzy-based data transformation for feature extraction to increase classification performance with small medical data sets," Artificial Intelligence in Medicine, 52(1), 45–52, May 2011.
10. H.-L. Chen, G. Wang, C. Ma, Z.-N. Cai, W.-B. Liu, and S.-J. Wang, "An efficient hybrid kernel extreme learning machine approach for early diagnosis of Parkinson's disease"" Neurocomputing, 184, 131–144, Apr. 2016.
11. W.-L. Zuo, Z.-Y. Wang, T. Liu, and H.-L. Chen, "Effective detection of Parkinson's disease using an adaptive fuzzy k-nearest neighbor approach," Biomedical Signal Processing and Control, 8(4), 364–373, Jul. 2013.
12. M. Gök, "An ensemble of k-nearest neighbours algorithm for detection of Parkinson's disease," International Journal of Systems Science, 46(6), 1108–1112, Apr. 2015.
13. B. E. Sakar et al., "Collection and analysis of a Parkinson speech dataset with multiple types of sound recordings," IEEE Journal of Biomedical and Health Informatics, 17(4), 828–834, Jul. 2013.

14. H.-H. Zhang et al., “Classification of Parkinson’s disease utilizing multi-edit nearest-neighbor and ensemble learning algorithms with speech samples,” BioMedical Engineering OnLine, 15(1), 122, Nov. 2016.
15. V. Abrol, P. Sharma, and A. K. Sao, “Greedy dictionary learning for kernel sparse representation based classifier,” Pattern Recognition Letters, 78, 64–69, Jul. 2016.
16. P. Refaeilzadeh, L. Tang, and H. Liu, “Cross-validation,” in Encyclopedia of Database Systems, Boston, MA, US: Springer, 2009, 532–538.
17. C. O. Sakar et al., “A comparative analysis of speech signal processing algorithms for Parkinson’s disease classification and the use of the tunable Q-factor wavelet transform,” Applied Soft Computing, 74, 255–263, Jan. 2019.
18. J. Cai, J. Luo, S. Wang, and S. Yang, “Feature selection in machine learning: A new perspective,” Neurocomputing, 300, 70–79, Jul. 2018.
19. S. C. Yusta, “Different metaheuristic strategies to solve the feature selection problem,” Pattern Recognition Letters, 30(5), 525–534, Apr. 2009.
20. B. H. Nguyen, B. Xue, and M. Zhang, “A survey on swarm intelligence approaches to feature selection in data mining,” Swarm and Evolutionary Computation, 54, 100663, May 2020.
21. D. Karaboga and B. Basturk, “A powerful and efficient algorithm for numerical function optimization: Artificial bee colony (ABC) algorithm,” Journal of Global Optimization, 39(3), 459–471, Nov. 2007.
22. D. Karaboga, B. Gorkemli, C. Ozturk, and N. Karaboga, “A comprehensive survey: Artificial bee colony (ABC) algorithm and applications,” Artificial Intelligence Review, 42(1), 21–57, Jun. 2014.
23. M. S. Kiran and M. Gunduz, ”XOR-based artificial bee colony algorithm for binary optimization,” Turkish Journal of Electrical Engineering & Computer Sciences, 21, 2307–2328, 2013.
24. M. H. Kashan, N. Nahavandi, and A. H. Kashan, “DisABC: A new artificial bee colony algorithm for binary optimization,” Applied Soft Computing, 12(1), 342–352, Jan. 2012.
25. R.Durgut, ”Improved Binary Artificial Bee Colony Algorithm”,arXiv preprint arXiv:2003.11641.
26. C. J. Santana, M. Macedo, H. Siqueira, A. Gokhale, and C. J. A. Bastos-Filho, “A novel binary artificial bee colony algorithm,” Future Generation Computer Systems, 98, 180–196, Sep. 2019.
27. E. V. Altay and B. Alatas, “Multi-objective association analysis of Parkinson disease with intelligent optimization algorithms,” in 1st International Informatics and Software Engineering Conference (UBMYK), Ankara, Turkey, 2019, 1–6.
28. H. Badem, D. Turkusagi, A. Caliskan, and Z. A. Çil, “Feature selection based on artificial bee colony for Parkinson disease diagnosis,” in 2019 Medical Technologies Congress (TIPTEKNO), İzmir, Turkey, 2019, 1–4.
29. T. Tuncer and S. Dogan, “A novel octopus based Parkinson’s disease and gender recognition method using vowels,” Applied Acoustics, 155, 75–83, Dec. 2019.
30. C. Castro, E. Vargas-Viveros, A. Sanchez, E. Gutierrez-Lopez, and D.-L. Flores, ”Parkinson’s disease classification using artificial neural networks,” in VIII Latin American Conference on Biomedical Engineering and XLII National Conference on Biomedical Engineering, Cancun, Mexico, 2020, 75, 1060–1065.
31. T. Tuncer, S. Dogan, and U. R. Acharya, ”Automated detection of Parkinson’s disease using minimum average maximum tree and singular value decomposition method with vowels”, Biocybernetics and Biomedical Engineering, 40(1), 211–220, Jan. 2020.