BULGARIAN ACADEMY OF SCIENCES
INSTITUTE OF INFORMATION AND COMMUNICATION TECHNOLOGIES
Department: "Information Processes and Decision Support Systems"

Edjola Naka

Optimization Algorithms for Data Management

ABSTRACT

for acquiring Educational and Scientific Degree
"DOCTOR"

In Professional field 4.6. "Informatics and Computer Science",
in Doctoral Program: Informatics

Scientific Supervisor: Prof. Dr. Vassil Georgiev Guliashki

Sofia, 2024

The thesis contains 165 pages, 25 figures, 33 tables, and 287 bibliography sources.

Dissertation Structure

The dissertation consists of an introduction, 3 chapters, conclusions - a summary of the results obtained, thesis contributions, a list of 7 publications on the dissertation.

In Introduction, it is given the motivation of writing the PhD thesis, optimization methods, and algorithms used in data management, feature selection problem, the importance of using metaheuristic optimization algorithms in feature selection, related to predicting Parkinson's.

Chapter 1 presents a detailed and analytical overview of the significance of optimization in data management, and in feature selection problem. It analyzes the theoretical background over the metaheuristic optimization algorithms, Parkinson's, machine learning classification algorithms, including existing proposed methods or algorithms for solving the feature selection problem.\

Chapter 2 provides a detailed information of all the proposed algorithms, methods, and techniques on feature selection using machine learning classifiers for evaluations. It presents a comparative analysis between existing feature selection methods, a novel metaheuristic used in feature selection, two approaches for improving its effectivity, and efficiency, respectively.

Chapter 3 outlines the experimental implementation and validation of the developed algorithms, techniques, and methods based on Parkinson data.

Conclusions presents the conclusions drawn from the suggested algorithms, and methods, limitations, and future work. Thesis contributions, the list of the scientific publications related with the dissertation, and citations are also provided.

Keywords

Data Management, Feature Selection, Metaheuristic, Optimization, Parkinson, Machine Learning, Algorithm

Introduction

There are different techniques for preprocessing the variety of generated data, among them dimensionality reduction and feature selection (FS). The ultimate one is an optimization problem that involves defining binary decision variables to indicate whether the feature is selected, an objective function that evaluates the performance of a model built using the selected features, and constraints that may limit the number of features selected. FS is a critical step in the machine learning pipeline to improve model performance, reduce complexity, and enhance interpretability. Metaheuristic optimization algorithms (MHOAs), a type of

optimization algorithm, have been widely used in recent decades to reduce the number of features and select the most relevant, important, and significant features from diverse datasets. Some key characteristics that make them frequently used are that they are not related to a specific problem, include a stochastic search, offer an iterative improvement of the candidate solution, and involve exploration and exploitation to efficiently navigate the search space in order to reach global optimum. In addition to utilizing single metaheuristics, combinations of them or local improvements on them are largely used. It is important to note that no universal metaheuristic approach can effectively solve all types of optimization problems across all application domains.

This dissertation introduces the binary volleyball premier league (VPL) algorithm first used on FS, where the continuous metaheuristic debuted in 2018. This metaheuristic algorithm simulates the original volleyball game conditions and the volleyball teams' competition in a league. Each team will compete with the others, and in the end of the season, the winning team will represent the best subset of features. Machine Learning (ML) classifiers are usually used for evaluating the quality of the solutions. The primary objective is to investigate, analyse, and improve the VPL to better solve the FS problem focusing on predicting Parkinson’s, and its efficiency and efficacy are investigated and improved to enhance the final achieved optimum and to produce it in a quicker time. Parkinson’s is one of the most important neurological diseases, and a lot of data are generated from whom need to be extracted the most important information. Since MHOAs have demonstrated remarkable efficacy in the FS problem, binary VPL adaptability in Parkinson's disease prediction is being studied as it hasn't been utilized in FS before.

To achieve these objectives, this dissertation comprises seven interconnected studies that utilize binary volleyball premier league algorithm, and improvements, feature selection methods, and machine learning classification algorithms to predict Parkinson's ([15], [16], [17], [18], [19], [20], and [21]). This research intends to show how metaheuristic optimization algorithms and machine learning can be used to get rid of features from the Parkinson's datasets that aren't useful to improve prediction accuracy and reduce the average size of the number of features that are chosen. The dissertation's goal is to create and improve new metaheuristic optimization algorithms for the feature selection problem in Parkinson's prediction using ML classifiers, with a focus on improving the efficiency, effectiveness and execution time of the Parkinson’s prediction algorithms.

1. Data Management, Machine Learning, and Metaheuristic algorithms: A state-of-the-art overview

This chapter describes metaheuristic optimization algorithms, their application in feature selection, the function of machine learning classifiers in feature selection, and provides an overview of Parkinson's disease.

1.1 Data Management and Optimization

1.1.1 Data Management

"Data Management is the development, execution, and supervision of plans, policies, programs, and practices that deliver, control, protect, and enhance the value of data and information assets throughout their lifecycles" [22]. Data management systems are built on data management platforms that, in addition, integrate databases, data warehouses, and big data management systems. Building and storing data in these infrastructures is interrelated with technologies, and fields for analyzing them in order to benefit from them. Machine Learning (ML) is one approach used for these analyses.

1.1.2 Optimization process

A key step in many decision-making and design processes is the optimization phase, which help determine realistic and practical outcomes of management decision-making and design processes. Regarding stochastic algorithms, in general they are divided in two types: heuristic and metaheuristic.

1.1.3 Optimization and data management

Metaheuristics have been largely used in relational databases, data warehouses, and big data. Almost all ML algorithms can be formulated as an optimization problem to find the extremum of an objective function. The existing literature analyzing the hybridization of metaheuristics and ML usually discuss the two approaches: ML is employed to enhance metaheuristics, and the other in which metaheuristics are used to improve the performance of ML techniques.

1.2 Feature Dimensionality

1.2.1 Dimensionality reduction

Dimensionality reduction is the process of transforming high-dimensional data into a low-dimensional space so that the low-dimensional representation retains meaningful features from the original data. Usually, there are two main approaches to dimensionality reduction: feature selection, and feature extraction.

1.2.2 Feature Selection

Feature selection is the process of selecting all relevant features and discarding the redundant and irrelevant ones, to maximize the classification rate of the classifier and diminish the complexity of the original dataset when faced with all the features of the dataset. There are three main methods to address the issue of FS: filter, wrapper, and embedded methods [48]. The wrapper-based approach in FS, includes using different supervised learning algorithms of ML as an approach for testing the fitness of the solutions generated by metaheuristics.

1.2.3 Feature importance.

One approach to dimensionality reduction, feature importance methods, ranks the features of a dataset, meanwhile feature selection reduce the size of the features of a dataset. Various studies have used the cosine method for selecting and reducing the number of features in high-dimensional datasets.

1.3 Parkinson's Disease

Parkinson's is a degenerative condition of the brain associated with motor symptoms as slow movement, tremor, rigidity, and imbalance, and other complications, including cognitive impairment, mental health disorders, sleep disorders, pain, and sensory disturbances. Academics, and not only are searching for the most effective model for predicting Parkinson's using machine learning and optimization algorithms due to the relevance of this disease. All the tests of the thesis were evaluated on ten Parkinson's datasets. Nine of these datasets are publicly available, named shortly D1 ([73], [74]), (D2_S, D2_M, D3_S, D3_M ([75], [76], [77]), D4 ([78], [79]), D5 ([80], [81]), D6 ([82], [83]), D7 ([84], [85]) while one was obtained from the Parkinson's Progression Markers Initiative - D8 [86][1], through a private request made by the author.

1.4 Metaheuristic Optimization Algorithms

1.4.1 Concepts about metaheuristics, taxonomy, and applications

A metaheuristic algorithm seeks to find a near-optimal solution instead of specifically trying to find the exact optimal solution, usually has no rigorous proof of convergence to the optimal solution, and is usually computationally faster than an exhaustive search [90].

[1] Funding for the D8 dataset: PPMI – a public-private partnership – is funded by the Michael J. Fox Foundation for Parkinson's Research and funding partners, including 4D Pharma, Abbvie, AcureX, Allergan, Amathus Therapeutics, Aligning Science Across Parkinson's, AskBio, Avid Radiopharmaceuticals, BIAL, Biogen, Biohaven, BioLegend, BlueRock Therapeutics, Bristol-Myers Squibb, Calico Labs, Celgene, Cerevel Therapeutics, Coave Therapeutics, DaCapo Brainscience, Denali, Edmond J. Safra Foundation, Eli Lilly, Gain Therapeutics, GE HealthCare, Genentech, GSK, Golub Capital, Handl Therapeutics, Insitro, Janssen Neuroscience, Lundbeck, Merck, Meso Scale Discovery, Mission Therapeutics, Neurocrine Biosciences, Pfizer, Piramal, Prevail Therapeutics, Roche, Sanofi, Servier, Sun Pharma Advanced Research Company, Takeda, Teva, UCB, Vanqua Bio, Verily, Voyager Therapeutics, the Weston Family Foundation and Yumanity Therapeutics.

1.4.2 Recently proposed metaheuristics

Approximately 540 new metaheuristics have been developed, with about 385 of them appearing in the last decay, and only in 2022 alone, around 47 ‘novel’ metaheuristics were proposed [12].

1.4.3 Metaheuristic algorithms and feature selection

MHOAs can be used to solve various problems, including the FS problem [91], [7], [72]. A fitness function, also known as an objective function, plays a crucial role in optimization by quantifying the degree to which a particular solution achieves the desired outcome. Metaheuristics initiate the search process by selecting random solutions, with the goal of finding an improved solution in each iteration. Exploration is vital in the initial iterations for discovering the entire search space, whereas exploitation is crucial in the last iterations for locating better solutions. Different discretization and binarization methods can be applied on the continuous metaheuristics [184] in order to be adapted for the feature selection conditions. The two-step binarization technique involves using transfer functions (TF) to transform continuous values within the range of 0 to 1, and then transferring the real number using methods as standard or complement to convert them into binary values, 0 or 1.

1.4.4 Proposed approaches in improving metaheuristics

Researchers have developed various techniques to enhance the performance of metaheuristics for feature selection. Operator modifications, opposition-based learning, chaotic maps, Levy flight, and transfer functions are the most commonly used operators and components to enhance the performance of metaheuristics [13].

1.4.5 Hybrid feature selection methods

Usually, filter, wrapper methods, and metaheuristics are employed in FS in predicting Parkinson’s. Hybridization in the context of metaheuristics refers to combining two or more metaheuristic algorithms to create a new, often more efficient, method.

1.5 Supervised learning algorithms

When predictions are required in many categorization tasks, supervised learning algorithms are frequently used. A class output of a dataset and a list of numeric and non-numeric variables are the two inputs for supervised learning techniques. In our situation, these algorithms are utilized to determine whether a person has Parkinson or not.

1.5.1 k-nearest neighbour

It is a non-parametric, supervised learning classifier, which uses proximity to make classifications or predictions about the grouping of an individual data point. K-NN algorithms

are easy to implement, adapts easily when new training samples are added, and has only a hyper parameter which makes it easier to use [214].

1.5.2 Support vector machines

SVM is a supervised algorithm with the objective to find a hyperplane in an N-dimensional space (N — the number of features) that distinctly classifies the data points.

1.5.3 Random Forest

The Random Forests (RFs) is composed of multiple independent decision trees that are trained independently on a random subset of data.

1.5.4 Performance metrics

The most common metrics used in FS is the fitness function, presented according to best, worst, average, and standard deviation of fitness. Average accuracy, and number of selected features are two other ones.

1.5.5 Hyper-parameter optimization and metaheuristics

Hyper-parameter optimization (HPO) is crucial in machine learning because it aims to find the best set of hyper parameters for a given model and dataset. Metaheuristics can be used as an option for optimizing the parameters of the machine learning algorithms, and has resulted very effective compared to traditional methods.

1.6 Chapter conclusions

This chapter's goal is to look closely at how metaheuristic algorithms, machine learning classification algorithms, hyper-parameter optimization, and performance metrics are used to evaluate FS subsets. The purpose of this is to demonstrate the significance and outcomes of their implementation in enhancing the precision of Parkinson's prediction and other applications. The chapter emphasizes the significance of each feature selection method, but there are still potential algorithms or approaches that could be proposed, which could be more effective and contribute to the FS problem.

❖ The goal of the dissertation is:

To create and develop new metaheuristic optimization algorithms for the feature selection problem and improve them in efficacy, efficiency, and performance time, contributing to Parkinson's prediction using machine learning algorithms. It encompasses the analysis of existing research methodologies and methods for selecting relevant and important features from Parkinson’s data, interconnected with innovative metaheuristic algorithms used in feature selection by improving their exploration and exploitation capabilities, with the aim of

maximizing Parkinson's prediction accuracy. The objective is to provide new algorithms to support forecasting Parkinson's regardless of the input data, as well as to propose a new method for identifying the most important features while reducing the data's dimensionality, maintaining a reasonable machine learning accuracy, and a reasonable execution time. In this regard, the goal of the dissertation was achieved by the following research tasks:

Task 1: To evaluate the "state of the art" of optimization methods, mostly metaheuristics, in data management, and feature selection, for predicting Parkinson's emphasizing their importance in this field.

Task 2: To carry out a comparative analysis of different filter and wrapper methods that uses the heuristic simulated annealing algorithm for hyper-parameter optimization of three machine learning classifiers with the aim of achieving high prediction accuracy for Parkinson's.

Task 3: To propose a new effective metaheuristic algorithm named "Binary Volleyball Premier League" applied for the first time in feature selection for predicting Parkinson's.

Task 4: To enhance the effectiveness of the Binary Volleyball Premier League by incorporating an opposition-based learning technique into its final solution, which will strengthen its search space exploration in favor of increasing the accuracy of Parkinson's prediction.

Task 5: To propose a novel hybrid metaheuristic, named BVPL_BALO, that merges Binary Volleyball Premier League learning phase and Binary Antlion Optimizer phase of generating new solutions, contributing in improving the exploitation of the actual solutions in order to improve the effectiveness of Binary Volleyball Premier League.

Task 6: To incorporate an "occurrence list" procedure into the hybrid metaheuristic Binary Volleyball Premier League and Antlion Optimizer algorithm which reduces its performance time resulting in a significant improvement in its efficiency comparing to Binary Volleyball Premier League.

Task 7: To develop a new advanced method that improves the efficiency of the Binary Volleyball Premier League and Antlion Optimizer algorithms by incorporating a feature ranking algorithm to prioritize reducing the feature dimensionality on high-dimensional datasets before employing the reduced number of features into the hybrid metaheuristic BVPL_BALO.

2. The proposed metaheuristic algorithms and, methods on feature selection Parkinson-based

This chapter introduces novel metaheuristic algorithms, methods, and enhancements used for the first time in FS and Parkinson's prediction. Medical industries, research groups, and non-profit organizations generate a significant amount of data about Parkinson's, leading to an increase in patient data. This has led to an increase in the dimensionality of the provided data, coinciding with the use of methods and algorithms for selecting, reducing and identifying the most important features from Parkinson data. The methodology for solving the proposed tasks on this dissertation follows these sequential steps as presented in Figure 2.1.

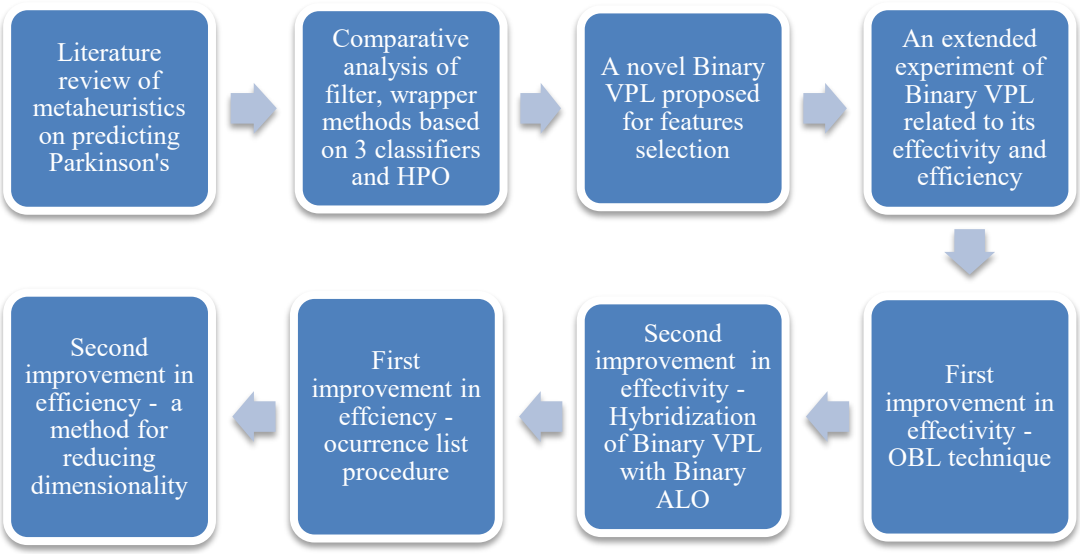

Figure 2.1. The steps of the methodology of the research

2.1 Trends of metaheuristics in feature selection Parkinson-based

Primarily, it was conducted in-depth research on the binary metaheuristics used in FS to predict Parkinson's, fulfilling a part of the requirements of Task 1. This review was necessary to understand the frequency with which metaheuristics were proposed as a solution to select the most identifiable features for Parkinson's prediction and which criteria were applied for this evaluation. A comprehensive search was done on Google Scholar and Research Gate, limiting it to publications published until 2022. The search was implemented using the keywords "metaheuristics algorithms" + "feature selection." There were selected papers that included only Parkinson datasets in their analysis. The collection of datasets under consideration consists of a total seven public datasets, D1, D2_S, D2_M, D3_S, D3_M, D5, and D7. A set of exclusion and inclusion criteria were applied to an overall total of 175 conference papers and journals. The selection process involved choosing papers written in the English language and employing supervised learning algorithms for their applications. Reviews and papers that solely employ the metaheuristic technique for hyper-parameter optimization within machine learning algorithms were excluded. A total of 34 publications were selected from Elsevier,

Springer, IEEE Xplore, MDPI, Hindawi, and Sage. The best proposed metaheuristics together with highlighting the ML algorithm, the performance metrics, fitness function, resampling methods, statistical test, and results related mostly with the accuracy, and number of selected features are analyzed for each source paper.

2.2 A comparative analysis of filter, and wrapper methods

Since filter and wrapper methods have been very popular and extremely helpful in feature selection problems, an approach is proposed that provides a comparative analysis of different feature selection methods in a voice Parkinson dataset (D1) in order to find an optimal subset with relevant features that gives the highest accuracy. The performance of each feature selection method is evaluated through the accuracy of three popular supervised learning algorithms, and Generalized Simulated Annealing (GSA) is used to improve the accuracy of the hyper-parameters of the classifiers. Figure 2.2 mentions all the methods used.

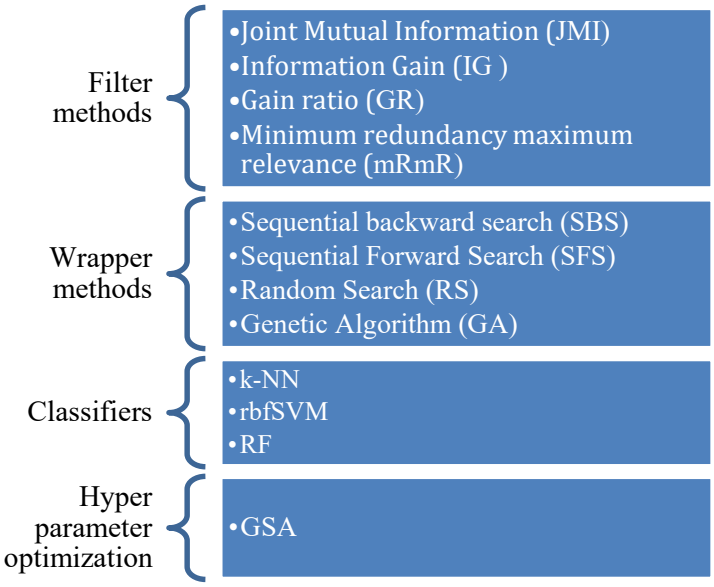

Figure 2.2. The methods used on the comparative analysis

This implementation accomplishes two goals:

- To evaluate and determine the benefits of the filter and wrapper methods in predicting Parkinson and their overall effectiveness.
- The GSA algorithm is employed in hyper-parameter optimization of certain classifiers to highlight the significance of this heuristic approach in improving performance metrics of ML classifiers.

The filter methods do not include a classifier in the process of selecting the optimal subset of features. Figure 2.3 presents the methodology used for filter methods, which primarily involves

generating subsets and then evaluating them using classifier machine learning algorithms. IG and GR are entropy-based filters. These algorithms find weights of attributes basing on their correlation with the class attribute. JMI tries to maximize the mutual information between a subset of selected features and the target variable. The mRmR is a widely used filter method for feature selection that uses mutual information to calculate measures of relevance and redundancy between the different features and the class label [254].

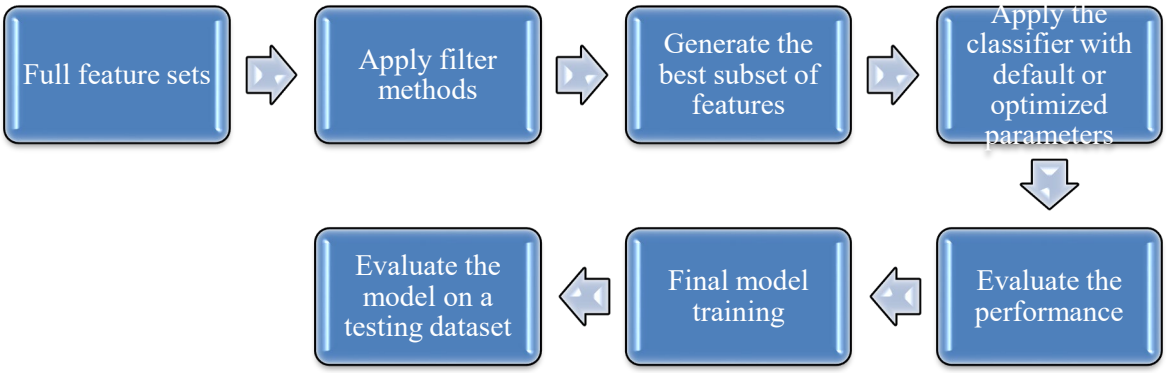

Figure 2.3. The process of applying filter methods in feature selection

Meanwhile, wrapper methods use a learning algorithm as a black box for the feature subset selection. Wrapper methods can require more computational time due to the complexity of each classifier's steps. The chosen wrapper methods are: backward, forward search, and random search. Backward search (SBS) starts with an empty set and adds one by one features from the full set, while forward search (SFS) starts with the full dataset and removes the features one by one, generating in the end the final feature subset [47]. The two schemes cannot guarantee finding the optimal subset; therefore, it can be used a random feature generation using the Random Search (RS) method, with the idea of not stucking to some local minima [47]. This method starts searching for the features randomly, and adding or removing features is done also randomly. Figure 2.4 presents the methodology followed for the wrapper methods when applied for FS.

Moreover, it is also integrated the GA which is an evolutionary metaheuristic, developed by John Holland and his collaborators in the 1960s and 1970s in [255] based on Charles Darwin's theory of natural selection.

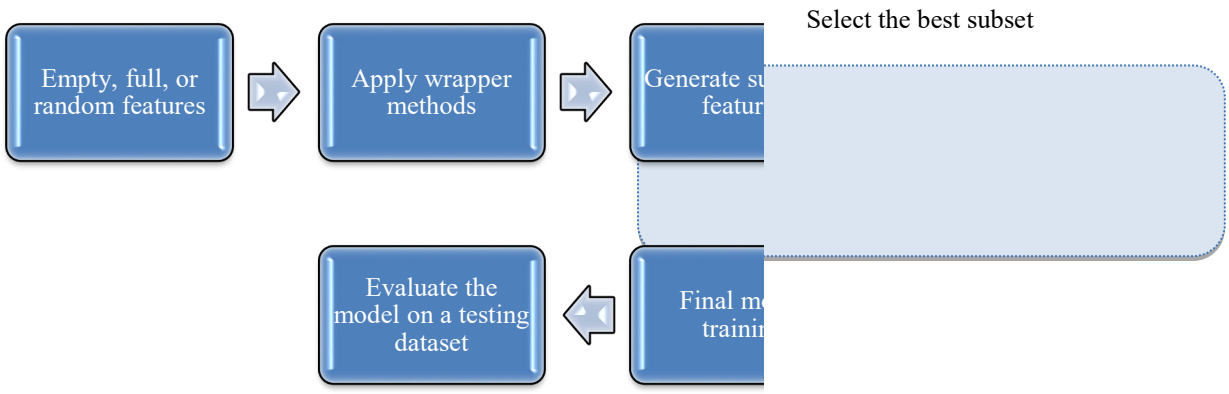

Figure 2.4. The process of applying wrapper methods in feature selection

Simulated Annealing is a physics-based algorithm inspired by the annealing process that happens to particles within a material [257]. Fast Simulated Annealing (FSA) is a semi-local search and consists of occasional long jumps. The cooling schedule of the FSA algorithm is inversely linear in time which is fast compared with the classical simulated annealing (CSA) which is strictly a local search and requires the cooling schedule to be inversely proportional to the logarithmic function of time [258]. Generalized Simulated Annealing (GSA) heuristic algorithm is proposed as an approach that combines the classical SA ("Gaussian visiting distribution") [257] and fast ("Cauchy - Lorentz visiting distribution") simulated annealing [258] and is considered quicker than both of them [259]. The implementation of GSA is used for hyper-parameter tuning, using the mlr package in the R language [261], has two existing functions named *tuneParams()* and *makeTuneControlGenSA()*. The first function optimizes the hyperparameters of the classifier algorithm, while the second function applies GSA.

2.3 A new Binary Volleyball Premier League algorithm in Feature Selection

2.3.1 Mathematical formulation of Volleyball Premier League algorithm

The foundational algorithm of this work is the Volleyball Premier League Algorithm, which Moghdani and Salimifard [262] first created. VPL is considered a human-based algorithm that is inspired by the volleyball league. The composition of players consists of active players, who are those who participate in a game or competition from the initial stages, and passive players, who are substitutes who have the potential to enhance the team's overall performance and are selected by the coach. In VPL, a league represents a population, a team represents a solution, an iteration is a season, a week means the schedule, and the winning team at the end of each season represents the best solution. The VPL Algorithm encompasses 11 distinct steps. The initial phase involves the initialization process, which starts with the utilization of two matrices named formation and substitution with random values. Their dimensions are the team's number of players and the dataset's number of features. The second phase consists of the setting of the

match schedule, which determines the schedule and order of the competitions among the participating teams. In this competitive setting, two teams compete with each other and afterwards, a winner is determined. Probability of winning and power to win the match are calculated when each team plays against each other. Afterward, the losing team is exposed to three strategies: knowledge exchange, repositioning, and substitution. The winning teams should adjust their positions taking into account the best team values, whereas losing teams cannot. During the learning phase, the three first ranked-teams are pointed out, and the lower-ranked teams learn from the teams with better performance. At the end of the season, the best teams are promoted to higher leagues and the worst teams are relegated from the league and will be substituted with new ones. The three concluding stages employed to enhance the efficacy of the proposed solutions are season transfers, promotions, and relegations. In Figure 2.5 it is presented a flowchart where all the steps of VPL are emphasized, and it introduces the idea of programming the VPL in R language.

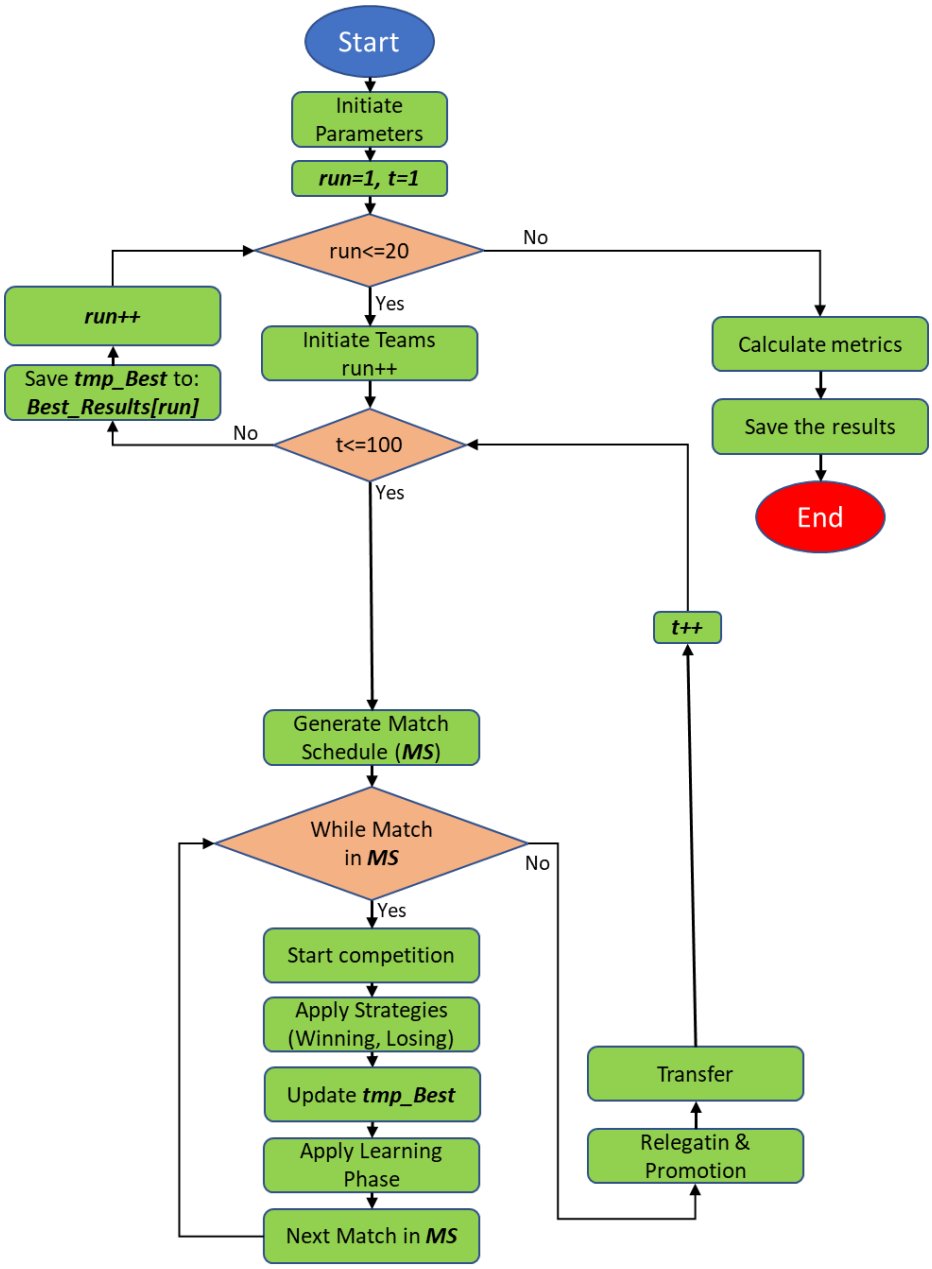

Figure 2.5. The flowchart of volleyball premier league algorithm

The algorithm executes 11 steps for each iteration (season), and stores the best solution (team) for each season (run) in the vector Best_Results[run]. The results of each run are independent from each other.

2.3.2 The proposed Binary Volleyball Premier League algorithm feature selection-based

The thesis primarily relies on the VPL algorithm to propose and assess its effectiveness and advantages in feature selection. Compared to other metaheuristics, VPL has not been widely adopted by researchers. The VPL was selected based on several factors:

- It had not been utilized before for feature selection, and it encompasses numerous ways to enhance the solutions till the optimal outcome is attained.

On the contrary, it possesses certain drawbacks:

- A decreased convergence rate, possibility of trapping in a local optimum, a learning phase which affect the performance of VPL
- The algorithm exhibits increased complexity due to the substantial number of phases and frequent calculations of team costs.

The actual VPL is used for solving optimization problems where the improved teams of each stage start and generate real values. On the other side, FS is known as a binary problem. By combining the original VPL functionality with several additional operators, a new variation of the algorithm has been designed to optimize solutions in a binary space. The VPL teams represent the solution, and it is necessary to binarize these teams to provide the optimal feature combination for maximum accuracy. Each position in the solution can have binary states: “1” or "0." A value of 0 signifies the absence of feature selection, whereas a value of 1 indicates the selection of a feature. all the features. The FS process, when combined with BVPL, consists of the following steps:

1. The initialization phase is the initial step of any FS technique, and it depends on all the original features present in the dataset. The Formation and Substitute matrices divide the total number of features. The players in "Formation" define the maximum number of possible selected features, which is pre-defined by the user. The Substitute matrix includes the other features that are not in the formation in which the VPL phases, namely repositioning, substitution, and knowledge transfer, can interchange with those of the Formation.

2. The second stage is the subset discovery to select candidate subset of feature for evaluation. The binary VPL is proposed for this phase. The two-step binarization is encoded in each new generated team (solution) during all the phases of VPL, and is integrated in BVPL algorithm as in Figure 2.6.

3. The third stage involves evaluating the feature subset generated. An assessment measure will evaluate the subset of features generated by the second stage, identifying their performance. In this instance, the subset of features that have been selected is validated on the test set using this cost function.

```
Cost function (Team) {
---- Cost calculation
Calculate Team Accuracy using k-NN
error=1-Accuracy
alpha=0.99
Cost=alpha*error+(1-alpha) *(length (Selected Features) / dimensions)
Return Cost
```

```
}
```

4. Predetermining the number of iterations reached is one of the stopping criteria's tasks in the fourth stage. Once the stopping criterion is satisfied, the loop will stop.

5. The best achieved result is stored.

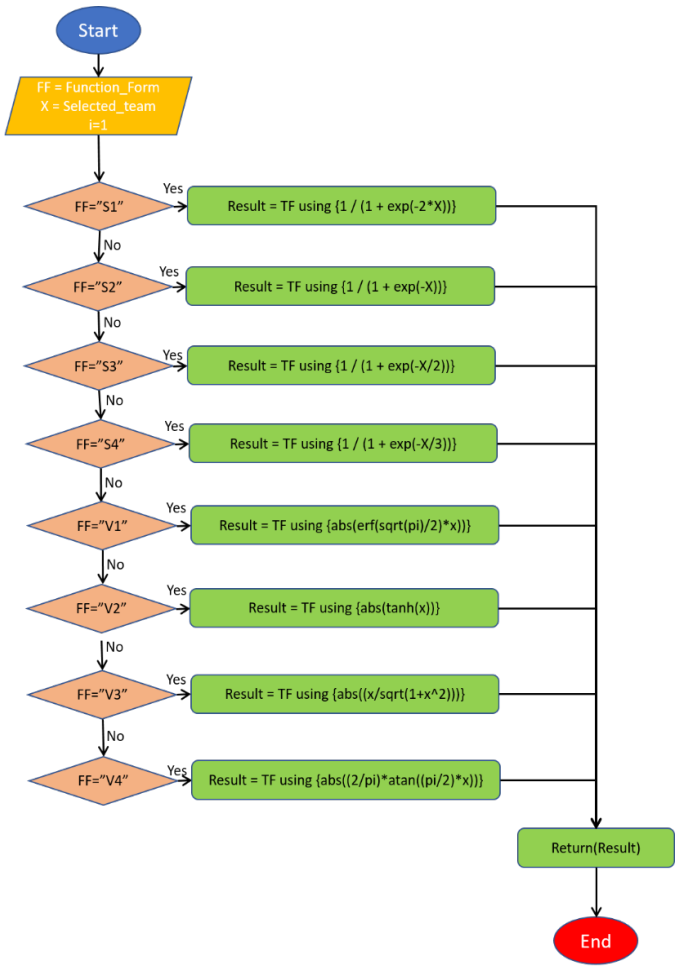

Figure 2.6. The flowchart of two-step binarization

2.4 Improving effectivity of Binary Volleyball Premier League algorithm

In this section, there are proposed two solutions with the goal to predict with a higher accuracy Parkinson's using the binary metaheuristic Volleyball Premier League.

2.4.1 Integration of opposition-based learning in Binary Volleyball Premier League algorithm

The main principle of "opposition-based" learning (OBL) is to evaluate simultaneously the fitness values of the current solution and its corresponding opposite solution, then retain the dominant individual to continue with the next iteration, thus effectively strengthening population diversity. In this connection, an integration of "opposition-based" learning in the binary Volleyball Premier League algorithm is proposed here. OBL is integrated with the aim

of searching for a better solution than that provided by the BVPL, and to explore new other solutions. The mathematical equation for OBL is applied as in Eq. (2.1), where x_{op_sol} is the opposite solution, and x_{act_sol} is the actual solution found better so far.

$$x_{op\_sol} = lower\ boundary + upper\ boundary - x_{act\_sol} \tag{2.1}$$

The idea of using OBL is expressed as in the Figure 2.7.

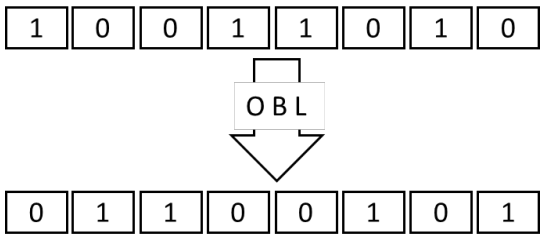

Figure 2.7. Selection of features when using opposition-based learning

Let suppose that the vector of the features evidenced as the best one after applying all the steps of BVPL at the end of iteration is the upper one, where 1 means that the feature is selected, and 0 the feature is not selected. After applying the OBL technique, the vector of features is the same as in the lower part where now the selected features correspond to the 1 value which were not selected in the best solution provided by BVPL.

An OBL technique is implemented in the final phase of BVPL, after concluding season transfer where besides the best solution (team) found so far, it is calculated also the cost of the opposite-based learning solution (here is stored in the variable named Temp). After comparing them, the solution which provides the lower cost, will be retained and will be used as the best solution provided so far in the next solution. The implementation of this idea is shown in Figure 2.8.

Integrating OBL give some advantages:

- If the best optimal solution is chosen from the OBL, it is then incorporated into the subsequent iteration, aiding in the generation of another optimal solution. The influence of the top three ranked teams on the new solution will impact the outcomes of the next iteration, leading to a prediction with higher accuracy.

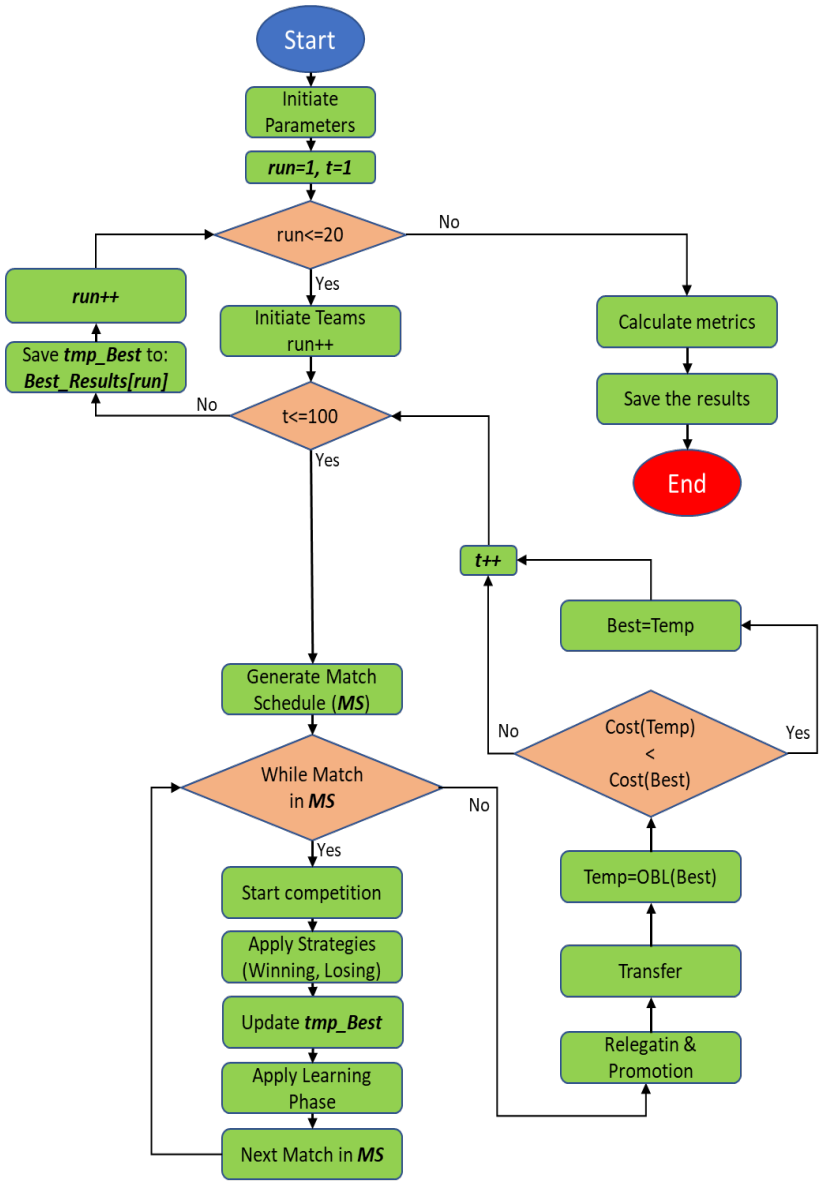

Figure 2.8. The flowchart of opposition-based learning BVPL

- Since BVPL is limited to the number of features selected, the OBL_BVPL solution offers a wider selection of features. This improves its exploration abilities, reducing the risk of remaining at the local optimum.

2.4.2 A hybrid Binary Volleyball Premier League and Antlion Optimizer metaheuristic algorithm

A new hybrid metaheuristic optimization algorithm that combines two different metaheuristics—the ALO metaheuristic and the BVPL metaheuristic algorithms—to solve the problem of FS is a new proposal in the thesis. The aim of this integration is to improve the learning phase of the BVPL algorithm since affects the performance of VPL. Hybridization is employed to prevent becoming trapped in a local optimum giving the option to diversify

solutions. Unlike OBL, which is utilized after an iteration and improves a team (solution) cost, the utilization of antlion improves all the teams using a probability rate specified by the user.

The ALO algorithm was proposed by Mirjalili [267] and mimics the hunting mechanism of ant lions in nature. The binary ALO proposed here is used according to the work of [268], named BALO approach 2, adapted with the presented cost_function, and two-step binarization. ALO has very competitive results in terms of improved exploration, local optima avoidance, exploitation, and convergence [267]. It addresses two problems of BVPL which are the risks to fall in a local optimum, and exploration. These are some reasons for including BALO's solutions in the matrices of teams of BVPL. One strategy mentioned for hybridization was the integrative approach, where one algorithm is considered a subordinate or embedded part of another [12].

2.4.2.1 The mathematical formulation of Ant Lion Optimizer

2.4.2.2 The new hybrid Binary Volleyball Premier League and Antlion Optimizer

The learning phase of VPL creates an extensive searching range for the algorithm. The main advantage of the VPL algorithm comes from the learning phase, making all teams follow the top three teams. However, the learning phase has the largest effect on the performance of the VPL algorithm, and this phase can lead to the VPL getting stuck in an optimal local solution [265]. In order to improve BVPL, and taking into consideration the advantages of BALO, the learning phase of BVPL is improved using the method of generating the best solutions using BALO algorithm. In the event that the BALO learning phase generates a better solution, the team with better fitness will be updated on the team table. So BALO improves the searching area of BVPL. In this approach, the probability of the fitness function f_i is calculated as in Eq. (2.2). Based on the value of $Prob_i$ the current team can update its behavior using the BALO operators, or else the traditional process in BVPL.

$$Prob_i = \frac{f_i}{\Sigma f_i} \tag{2.2}$$

The hybrid metaheuristic BVPL_BALO is presented in a pseudocode form in Algorithm 1.

Algorithm 1. The proposed hybrid metaheuristic BVPL_BALO

```
Input: iteration = 0, parameters, max_iter, nruns
Output: average fitness, standard deviation of fitness,
average of selected features, average accuracy
1. Initialization
2. Create an Occurrence List (first element = all 0 team with Cost 1)
For nruns = 1 to nruns
t =1; maxiter = 100
  While t < maxiter
```

```
         3. Generate a league schedule
         For i=1: (N-1)
      Best team =Select Best team according to Cost function
// ---Two-step binarization is applied each time for converting a continuous
team to a binary one ---//
// --- Cost function is applied each time that the fitness of the team needs
to be calculated -//
             For (each match in schedule table of week i)
4. Apply Competition procedure between team A, and B
            5. Determine winner and loser teams
            6. Apply different strategies for winner
                       and loser teams
            Update Best team
            7. Calculate the probability (Probi) (2.2)
            If (Probi >rand)
                   8. Apply BALO
                 If (Team$fitness>New_team$fitness){
                      Team=New_Team
                 End if
            Else
                   9. Apply learning phase BVPL
            End If
              End For
      i=i+1
         End For
    10. Apply Promotion and relegation process
    11. Apply season transfer process
    t = t + 1
    End While
End For
```

Figure 2.9 illustrates in a flowchart the proposed hybrid metaheuristic BVPL_BALO. In this figure, the hybrid metaheuristic uses the same steps as BVPL until the learning phase. The difference comes in improving the solutions of the learning phase using BALO or BVPL in generating new solutions based on a probability, and an if condition. The metaheuristic BALO is applied when the probability is greater than a generated random number between 0 and 1.

2.5 Improving efficiency of the hybrid Binary Volleyball Premier League and Antlion Optimizer metaheuristic algorithm

A drawback which is observed when using VPL is its largest execution time. The complexity of VPL is dependent on the number of populations or teams: active + passive (2n), number of dimensions of the dataset (dim), the number of iterations (T), fitness function f*i*, and is formulated according to Eq. (2.3) [265].

$$O(VPL) = (O(T(2n^2))) + O(n * dim) * O(f_i) \quad (2.3)$$

The equation shows that as the number of populations and dimensions increases, the complexity of VPL also increases.

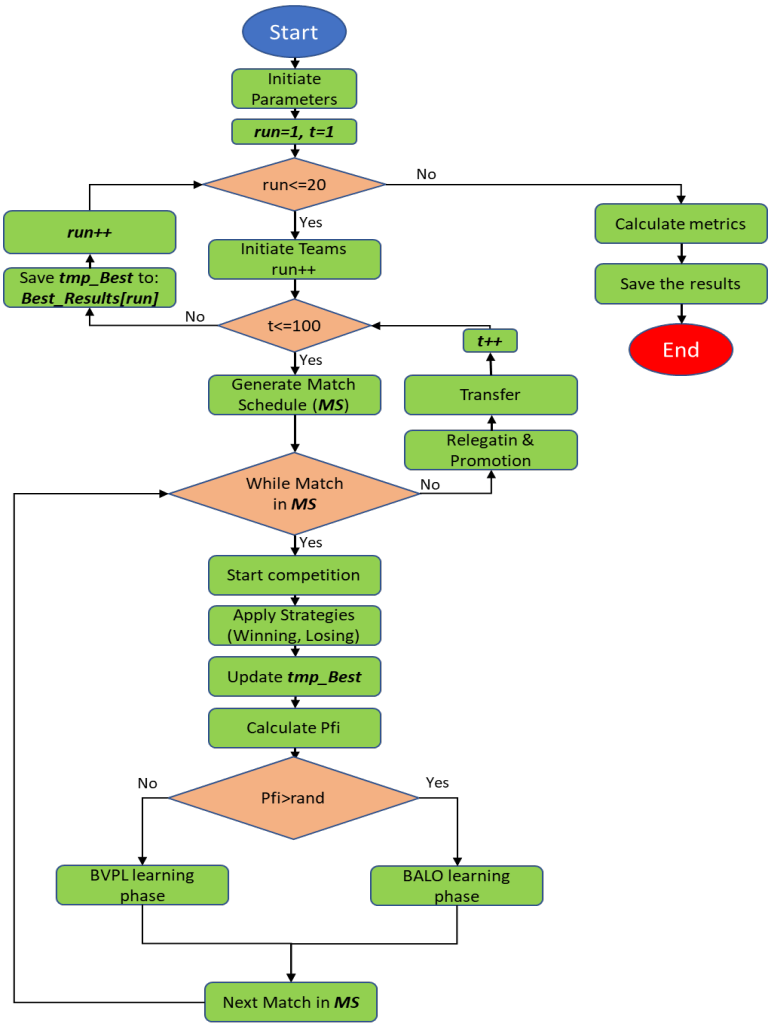

Figure 2.9. Flowchart of BVPL_BALO

Besides it, when the proposed BVPL is applied in feature selection, the execution time is increased when using the cost function because this calculation is affected by the complexity of k-NN classification algorithm, and the number of times the classification algorithm is executed.

2.5.1 Integrating the occurrence list in the cost function

For solving the drawback of the largest execution time required by the hybrid metaheuristic which normally it is inherited by the BVPL metaheuristic algorithm, a new technique is integrated in the proposed hybrid algorithm in order to improve the execution time of BVPL. This approach stores in a list, named here "occurrence list", the binary positions and the fitness of the generated teams from the previous iterations. This new technique restricts the necessity of recalculating the fitness for the exact team, hence allowing the fitness value to be extracted

directly from the "occurrence list". The following pseudocode presents the calculation of the new improved cost function by adding this list.

```
Cost function (Team) {
---- Check if Team is in Occurrence List
If (Team is in Occurrence List)
            Get Cost from Occurrence List
      Else
---- Cost calculation
Calculate Team Accuracy using k-NN
error=1-Accuracy
alpha=0.99
Cost=alpha*error+(1-alpha) *(length (Selected Features) / dimensions)
Add Team and Cost to Occurrence List
End If
Return Cost
}
```

This pseudocode is integrated in the proposed hybrid BVPL_BALO metaheuristic in order to evaluate its effect on reducing the execution time of BVPL_BALO. The results of this implementation will be tested on the high-dimensional dataset D5 together with other metaheuristics.

2.5.2 A new method combining cosine similarity and metaheuristic method

In this subsection, it is presented a new method which has the aim to improve the efficiency of the proposed hybrid metaheuristic, named shortly CS_BVPL_BALO. The proposed method contains two phases for reducing the features in the feature selection process. In the first phase, an algorithm is proposed that will rank the features according to their importance, considering the similarity of each row of the dataset with the average values for all the features of the dataset. The cosine similarity equation will be calculated for measuring this similarity. Each feature of the dataset will be removed one by one, and a recalculation of the similarity between each row and the other average feature row values will be applied using cosine similarity again. In the end, the average difference between the original average values with all the features of the dataset and the average values when each feature is removed is calculated, and the final score is provided. Once ranked, a metaheuristic receives the selected features as input. In summary, the general steps followed of the proposed method are given in Figure 2.10:

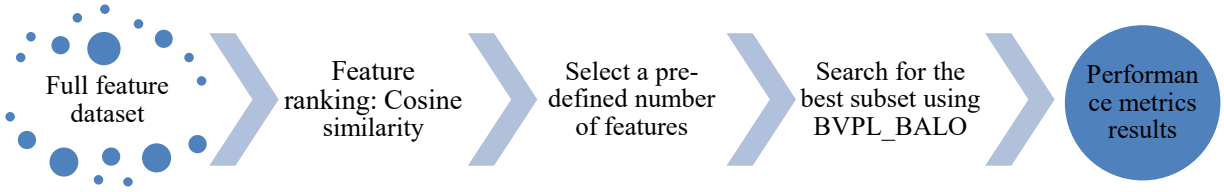

Figure 2.10. The steps of the proposed method CS_BVPL_BALO

Algorithm 2 provides a more detailed description of the method, representing the two phases with a total of 6 sequential steps and the corresponding equations.

Algorithm 2. Proposed method: CS_BVPL_BALO

Phase 1: Features ranking algorithm

Begin

Input: Dataset $P_{rows\ x\ dimensions}$

Step 1. Determine the mean feature value for each column of P. The mean values are stored in a row vector $M_{1\ x\ dimension}$.

Step 2. Calculate the distances between each row of P and row M. The similarity between the dataset and the mean vector, using the cosine similarity is determined by Eq. 2.4. The distances are stored in column vector $H_{rows\ x\ 1}$.

$$similarity = \cos(\theta) = \frac{PxM}{||P||x||M||} \frac{\sum_{i=1}^{n} P_i M_i}{\sqrt{\sum_{i=1}^{n} P_i^2} \sqrt{\sum_{i=1}^{n} M_i^2}} \quad (2.4)$$

Step 3. Generate the first dataset, P1, which is the dataset P without the first feature. Calculate the mean of each feature and store in vector M_1. Calculate the distances between each row of P_1 and row M_1 using Eq. 2.53. Store the distances in vector column H_1. Repeat the process for each feature in the original dataset. Finally, the column vectors H_1, H_2, …, $H_{dimension}$ are generated.

Step 4. Calculate the mean difference between vector H of Step 2 and each other vector $H_{dimension}$ as in Eq. 2.5.

For j = 1 to dim:

$$DIFmean[j] = \sum_{i=1}^{dim} |H_i - Hj_i|/dim \quad (2.5)$$

Step 5. Sort the features according to the vector DIFmean values in decreasing order. The feature that causes the largest distance from the original vector is considered the most important.

Output: A ranked list of features which allows to select a reduced dataset with only the most important features.

Phase 2: Apply BVPL_BALO algorithm

Step 6. A pre-defined percent of features of Step 5 are provided as an input to a selected metaheuristic algorithm, in this case is BVPL_BALO.

Final output: average, and standard deviation of fitness, average accuracy, number of selected features, and execution time.

End

Some advantages of using this method are:

- The combination of the two phases presents a new approach for reducing the dimensionality of the data combining a ranking features method and a FS metaheuristic algorithm.
- Implementation of the first phase, which is *Feature ranking algorithm* guarantee that a desired percent of important and identifiable features can be extracted from a large list of features, and given as an input to a metaheuristic algorithm.
- It would be easy to integrate other metaheuristic algorithms in place of the BVPL_BALO one.
- High-dimensional datasets can benefit from its reduction in dimensionality, but datasets with fewer features can also benefit from its use.

Regarding disadvantages of using this method, some of them are:

- The process of ranking the features using Phase 1 could be longer in time because calculating the similarity using cosine similarity is affected by the dimensions of the input dataset. However, this process is conducted only once, and the ranking features could be accessed anytime.
- This method does not guarantee that an extracted number of features is always effective for each given dataset.
- The proposed method does not guarantee the preservation of the predictive model's accuracy in all the datasets.
- The number of dimensions in the dataset and the complexity of the metaheuristic algorithm influence the complexity of this method.

2.6 Chapter conclusions

This chapter presents a group of novel metaheuristic algorithms and methods that are employed for the first time to address the feature selection problem with the focus on predicting Parkinson's. The main goal is to find the best set of features which predicts with a higher accuracy Parkinson's on each dataset. This chapter addresses the goals by completing tasks 1–7. Initially, a review examines the classifiers, performance metrics, resampling methods, statistical tests used to compare the metaheuristics, and the optimal combination that yielded the most favorable outcomes. Furthermore, four filter and wrapper methods are utilized to evaluate the performance of the subset of features using a Generalized Simulated Annealing heuristic algorithm to optimize the parameters of the classifiers. From this comparative

analysis, it will be identified the most resultative methods and the effect of Generalized Simulated Annealing in hyperparameter optimization. Next, a highly effective binary Volleyball Premier League algorithm is suggested to be used on feature selection for picking the most significant features in order to predict Parkinson's with a high accuracy and minimal feature size. The comparison with other metaheuristics and performance indicators will confirm its superiority in feature selection. Next, in BVPL, are incorporated two strategies to enhance its effectivity. Firstly, we incorporate an opposition-based learning technique into BVPL to expand its search area, minimize the risk of reaching the local minimum, and enhance the final solution. Secondly, the binary antlion optimizer is used in combination with BVPL to boost the learning phase and improve the exploitation phase of BVPL, proposing a hybrid BVPL_BALO metaheuristic that significantly improves BVPL's effectivity.

Two enhancements are suggested to reduce the execution time of BVPL and to improve its efficiency. In BVPL_BALO, a procedure is implemented that generates an occurrence list which reduce the number of fitness calculations required when the same solution is provided. The second enhancement involves a method that integrates a feature ranking algorithm, and the hybrid metaheuristic BVPL_BALO for ranking the feature according their importance, and applying feature selection.

3. Experimental results and findings

This chapter provide the results of the experiments in order to validate the novel proposed algorithms, methods, and improvements described in each subsection of Chapter 2.

3.1 Statistics of using metaheuristics in feature selection Parkinson-based

This section outlines the summarized information regarding the actual research and trend in using metaheuristics on predicting Parkinson based on machine learning algorithms. Figure 3.1 illustrate the availability of diverse metaheuristics applied in the domain of Parkinson's.

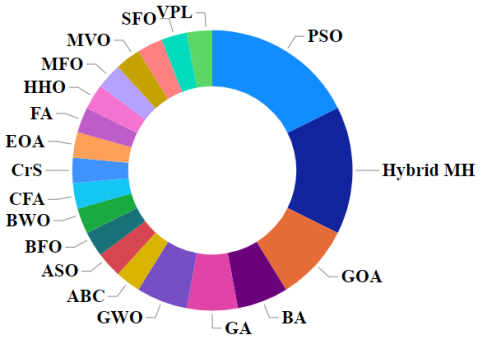

Figure 3.1. Distribution of metaheuristics

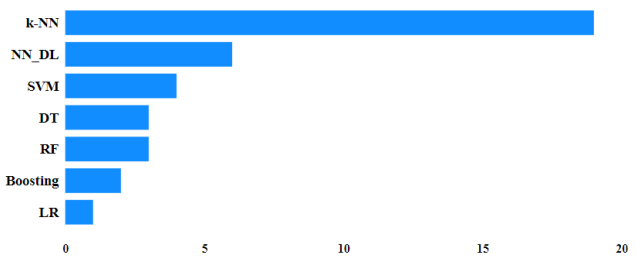

Figure 3.2. The frequency of usage of the SLA algorithms

PSO was the most frequently used method, appearing six times, followed by five instances of hybrid methods and three instances of GOA. Furthermore, sixteen different metaheuristics have been implemented only once. Typically, researchers evaluate the effectiveness of one metaheuristic by comparing it to another, using different datasets and measures.

Figure. 3.2 illustrates the frequency of utilization of each machine learning algorithm in the selected papers. For the wrapper-based approach, nineteen out of thirty-four papers commonly use K-nearest neighbor and its variants, with k = 5 being the most popular. Six publications primarily use neural networks (NN) and deep learning (DL) algorithms. Other frequently used methods are SVM (4 times), RF, and DT in 3 publications. In addition, there are research papers in which metaheuristics, such as GOA and MVO, are used both for evaluating the performance of the selected features and for hyper parameter optimization.

Regarding performance metrics, Figure 3.3 report the usage of average accuracy of the classifier in nearly all of the papers (97.06%). Next in line are the average feature size (76.47%) and computation time (32.35%). Only 23.53% of all publications appear to utilize the F1-score. On the other hand, 14.71% of publications report precision and specificity. The misclassification error rate, negative predictive value, mean squared error, and geometric mean indicate infrequent usage, with a rate of 11.76%.

Because FS is an optimization problem, the predominant objective function typically involves computing the classifier's accuracy or error alongside the number of selected features, which occurred on sixteen occasions. This is the same fitness function applied here on the thesis. Additionally, there are alternative formulas that employ diverse weighting parameter values,

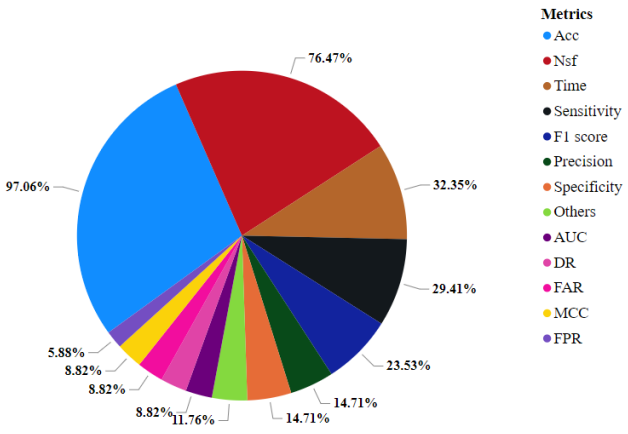

Figure 3.3. The percentage of usage of each metrics

denoted as alpha, and define the optimization problem as either a minimization or a maximization. The variable alpha typically assumes values within the range of 0 to 1. Some of the studies included in the review fail to provide a clear indication of the value of alpha.

Furthermore, some publications state that the classifier's accuracy determines the subset's assessment, but they fail to fully illustrate this for the readers. In addition, various authors have proposed distinct forms of objective functions. To reduce the problem of overfitting in ML, different resampling methods are proposed, such as cross-validation. K-fold CV, where k = 10, is frequently used (14 times). However, resampling methods are not always considered for Parkinson datasets, and sometimes they are not explicitly mentioned in the papers. Regarding the utilization of parametric or non-parametric tests, the Wilcoxon sum-rank test is the most frequently employed (8 instances). Subsequently, the Friedman test and Wilcoxon signed-rank test are utilized twice each.

Public datasets, such as voice, speech, and handwriting, were primarily detected in publications. With 23 publications, the D1 dataset is the most frequently used, followed by the D5 dataset with 10 publications, the D7 dataset six times, the D2 spiral and meander five times, and the D3 spiral and meander only once. The use of gait and handwriting datasets is less common.

For some specific occasions, the combination of the metaheuristic and the supervised learning algorithm gave higher Parkinson predictions. Some of the best results for each dataset are listed in continuity. For the D5 dataset, the best accuracy was given by a combination of the Fuzzy Monarch Butterfly Optimization Algorithm + Levy Flight Cuckoo Search Algorithm + Adaptive FA combined with a fuzzy convolution bi-directional long short-term memory deep learning algorithm. The accuracy of acc = 98.77% was taken using a combination of features (Mel frequency cepstral coefficients features + Wavelet features + Concat (baseline, vocal fold, and time frequency features)). Regarding the D1 dataset, almost all the methods produced accuracy greater than 90%. In particular, GOA + SVM produced an accuracy of 100% for six features of this dataset. Besides that, two average accuracies of 99.62% (average features = 2.15) and 99.19% (average features = 12.9) were generated by using an enhanced black widow optimization algorithm and PSO, respectively. When D2 and D3 datasets are applied, an optimized Crow search algorithm combined with k-NN, DT, and RF shows acc = 100%. For the last dataset, D7, a modified GWO-RF, modified GWO-DT, a modified GOA-RF, and improved Sailfish Optimization -bidirectional gated recurrent unit neural network generated an accuracy of 100%.

3.2 Results of comparative analysis of filter, and wrapper methods

The methodology in the Figure 3.4 presents a unified view of the overall stages which are followed to apply each filter, and wrapper method, the three used classifier algorithms, and how are conducted the evaluations of the final subsets. Firstly, filter, wrapper and GA methods will be applied which will generate their best subsets. After the evaluation of the subsets,

default and optimized parameters of the learning models will be applied in the same classifiers, to compare the new accuracy with the accuracy of the full features and with the default parameters. Then the new parameters will be used to test the new accuracy of each classifier, and to evaluate the difference.

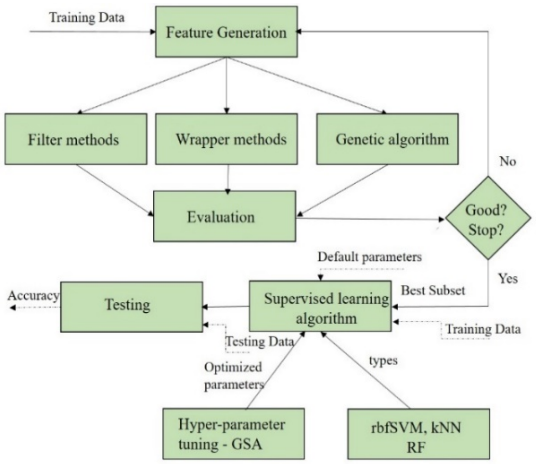

Figure 3.4. The methodology of the comparative analysis

The considered dataset for the analysis is the D1 dataset. All the 22 features of the dataset are numeric. The data were firstly normalized between range 0 and 1 using a min-max normalization. Status variable has two levels: “yes” when patients have Parkinson and “no” otherwise. The positive class is defined the level yes. There are no missing values in the data. All the methods described above are implemented and tested using mlr package in R [261]. The ratio between training and test dataset is always considered 70:30. Cross-validation is the resampling procedure used to evaluate machine learning models on a limited data sample. It is chosen the 10-fold cross-validation repeated 3 times because of the limited size of the dataset.

3.2.1 Results for full features

Initially, were chosen some well-known and largely used classifiers to do a preliminary control of the accuracy that they would classify the dataset. Logistic Regression, Neural networks, and Naïve Bayes were excluded from the analysis because of the low accuracy, specifically 83%, 85%, 68%. Figure 3.5 summarizes the results of the three mentioned classifiers (k-NN, rbfSVM, and RF) in terms of sensitivity, specificity, precision, and accuracy without using GSA (left), and with using GSA (right). K-NN has predicted more instances correctly in the Parkinson data, with an accuracy of 94% for the full features dataset, than two other classifiers. In case of k-NN, rbfSVM, and RF the difference in accuracy is increased 3%, 6%, and 2% respectively when using GSA. The measures when using k-NN are improved totally after using GSA for hyper-parameter tuning.

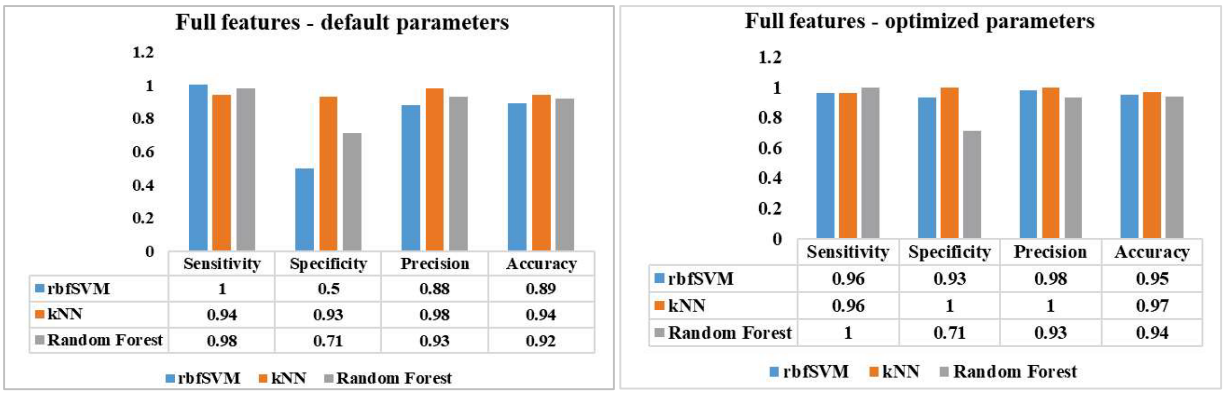

Figure 3.5. Performance measures of the three classifiers with default (left), and optimized parameters (right).

3.2.2 Filter Methods Results for Default and Optimized Parameters

In the filter methods, it is necessary to select a pre-defined number of features, in order to apply these features to the mentioned classifiers. There are selected the 25% of the top scoring features. The results for each filter method in conjunction with each classifier are summarized in Table 3.1. It can be observed that the accuracy of the subset generated by JMI versus the dataset with full features has had an improvement in case of k-NN and rbfSVM, whereas for RF there was an increase in the percent of the correctly classified observations for the healthy people and a slight increase in the precision, but with an unchanged accuracy. The subset generated by JMI when used k-NN shows the best results.

Table 3.1. Performance measures for the filter methods

	Without using GSA					Using GSA			
k-NN	Sensitivity	Specificity	Precision	Accuracy	k-NN	Sensitivity	Specificity	Precision	Accuracy
FF	0.94	0.93	0.98	0.94	FF	0.96	1	1	0.97
IG	0.82	0.71	0.91	0.8	IG	0.84	0.86	0.96	0.85
GR	0.82	0.79	0.93	0.82	GR	0.92	0.86	0.96	0.91
JMI	0.98	1	1	0.98	JMI	0.98	1	1	0.98
mRmR	0.9	0.93	0.98	0.91	mRmR	0.96	0.79	0.94	0.92
rbfSVM	**Sensitivity**	**Specificity**	**Precision**	**Accuracy**	**rbfSVM**	**Sensitivity**	**Specificity**	**Precision**	**Accuracy**
FF	1	0.5	0.88	0.89	FF	0.96	0.93	0.98	0.95
IG	0.96	0.5	0.88	0.86	IG	0.92	0.71	0.92	0.88
GR	1	0.43	0.86	0.88	GR	0.84	0.86	0.96	0.85
JMI	1	0.64	0.91	0.92	JMI	1	0.93	0.98	0.98
mRmR	1	0.57	0.89	0.91	mRmR	0.98	0.79	0.94	0.94
RF	**Sensitivity**	**Specificity**	**Precision**	**Accuracy**	**RF**	**Sensitivity**	**Specificity**	**Precision**	**Accuracy**
FF	0.98	0.71	0.93	0.92	FF	1	0.71	0.93	0.94
IG	0.96	0.71	0.92	0.91	IG	0.86	0.57	0.88	0.8
GR	0.96	0.71	0.92	0.91	GR	0.9	0.64	0.9	0.85
JMI	0.96	0.79	0.94	0.92	JMI	0.98	0.86	0.96	0.95
mRmR	0.98	0.71	0.93	0.92	mRmR	0.98	0.86	0.96	0.95

The main idea is to compare how does GSA affects the performance of each classifier for each feature selection methods. After it is used GSA, JMI method gives an improvement in the accuracy compared with the full features dataset for each classifier. It is seen that GSA doesn't affect and improve the accuracy of JMI filter method but it increases with 5%, 9% and 1 % the accuracies of IG, GR, and mRmR for k-NN algorithm. The subset generated from JMI and mRmR has an increased accuracy when used GSA in case of RF. In totally, among the filter methods, regardless of the classifier, the subset of JMI gives the best accuracy after using optimization with GSA.

3.2.3 Wrapper Methods Results for Default and Optimized Parameters

Regarding the wrapper methods, for forward, and backward search, it is used parameter alpha which shows the minimal required value of improvement difference for a forward/adding step. In this case, the value alpha is equal to 0.02. About RS and GA, the argument chosen was number of iterations, and the computation were executed for 100 iterations/500 iterations respectively. Following is presented Table 3.2 with the performance measures when it is applied or not the GSA algorithm after each subset created by wrapper methods, and GA.

Table 3.2. Performance measures for the wrapper methods

	Without using GSA					Using GSA			
k-NN	Sensitivity	Specificity	Precision	Accuracy	k-NN	Sensitivity	Specificity	Precision	Accuracy
FF	0.94	0.93	0.98	0.94	FF	0.96	1	1	0.97
SFS	0.92	0.86	0.96	0.91	SFS	0.88	0.79	0.94	0.86
SBS	0.94	0.93	0.98	0.94	SBS	0.94	1	1	0.95
RS	0.94	0.93	0.98	0.94	RS	0.94	1	1	0.95
GA	0.94	1	1	0.95	GA	0.94	1	1	0.95
rbfSVM	**Sensitivity**	**Specificity**	**Precision**	**Accuracy**	**rbfSVM**	**Sensitivity**	**Specificity**	**Precision**	**Accuracy**
FF	1	0.5	0.88	0.89	FF	0.96	0.93	0.98	0.95
SFS	0.96	0.57	0.89	0.88	SFS	0.92	0.57	0.89	0.85
SBS	1	0.57	0.89	0.91	SBS	0.98	0.79	0.94	0.94
RS	1	0.5	0.88	0.89	RS	0.96	0.93	0.98	0.95
GA	1	0.57	0.89	0.91	GA	0.98	0.79	0.94	0.94
RF	**Sensitivity**	**Specificity**	**Precision**	**Accuracy**	**RF**	**Sensitivity**	**Specificity**	**Precision**	**Accuracy**
FF	0.98	0.71	0.93	0.92	FF	1	0.71	0.93	0.94
SFS	0.96	0.86	0.96	0.94	SFS	0.94	0.86	0.96	0.92
SBS	0.98	0.71	0.93	0.92	SBS	0.98	0.71	0.93	0.92
RS	0.98	0.79	0.94	0.94	RS	0.98	0.79	0.94	0.94
GA	0.94	0.93	0.98	0.94	GA	0.92	0.93	0.98	0.92

Performing GA for generating a new subset of features came about with an improvement on all the measures when GSA is not used. In regards to Random Forest, SFS, RS and GA gives the same accuracy but GA has also a better precision. The combination GA with k-NN classifier achieves the best accuracy. In the right of Table 3.2 is summarized how the performance of each classifier changes with each subset generated by each wrapper method and GA when is

applied hyper-parameter tuning. There is no difference when optimizing the parameters of the k-NN algorithm for the subset of GA comparing with the default parameters of the learner, and there is a slight increase for SBS and RS in accuracy. The only decrease of measures is with SFS filter method (difference in acc = 5 %). The classifier rbfSVM has an increase in accuracy with 3%, 6%, 3% respectively for SBS, RS and GA, with default and optimized parameters, whereas for SFS there is a decrease with 3 %. When comparing wrapper methods for RF, we can state that the accuracies of the SBS and RS has not been changing whereas for GA and SFS has decreased with 2%. In overall, none of the subsets generated by wrapper methods or GA find a subset with a high accuracy than the dataset with the full features, but SBS, RS, GA with k-NN and RS with rbfSVM generates the higher accuracies after the optimization of learning parameters.

From all the experiments, it is concluded that in most of the cases, GSA increased the accuracy of the subsets generated. GSA helps in achieving better performance measures for each of the three classifiers for the full features dataset. From the three of them, k-NN has the highest performance with an accuracy 97% against the others. In regards with the filter methods, for the default parameters of the k-NN learner algorithm, the subset generated by JMI achieves the best results (acc = 98%). The optimal parameters defined by GSA achieves the best results with JMI + k-NN classifier, and also JMI combined with rbfSVM. There are some differences between wrapper methods and GA, when are used default and optimized parameters. Without using GSA, the combination GA and k-NN achieves the best results (acc = 95%), and additionally for rbfSVM and RF, GA gives better results than the full features dataset, respectively, acc = 91 % and acc = 94%. After using GSA, GA, RS and SBS with k-NN and RS with rbfSVM generates the higher accuracies with acc = 95%. When comparing the classifiers, k-NN and rbfSVM gives an improvement in measures for three from the four methods, in wrapper methods.

3.3 Results from Binary Volleyball Premier League algorithm in Feature Selection

In this paragraph are developed two experiments for testing the compatibility of BVPL in the feature selection problem. In the first one, BVPL is compared with a developed binary PSO metaheuristic algorithm based on one Parkinson's dataset, using the machine learning classifier k-NN as part of the evaluation of the fitness function. In the second experiment, there were enlarged the domain of the metaheuristics whom BVPL is compared, and also 9 other Parkinson datasets were used using again the k-NN classifier.

3.3.1 Experiment 1

In this first experiment, S2, and V4 transfer functions and respectively the standard and complement methods are applied as part of the two-step binarization steps. The metaheuristics

BVPL and BPSO are used as a search method to investigate the features region as well as to minimize the fitness function. The prediction is based on the D1 dataset. The general, and the parameters of both metaheuristics are presented in Table 3.3:

Table 3.3. Experiment 1 parameters

Parameter	Value
Iterations	100
Runs	30
alpha	0.99
No. particles_BPSO	10
ω_{min} – minimum inertia weight	0.4
ω_{max} – maximum inertia weight	0.9
V_{max} – maximum speed	6
c_1, c_2 – cognitive, and social factor	2
δ_{pr}	0.15
δ_{st}	0.5
No. of players	8
No. of teams in a league_BVPL	10

Table 3.4 presents the best minimum (Min_Fit), maximum fitness (Max_Fit), the average number of features (Avg_Feat), average (Avg_Best), and standard deviation (SD_Best) of best solutions achieved in all runs for the four algorithms: BPSO_V (BPSO V-shaped), BPSO_S (B-PSO S-shaped), BVPL_V (BVPL V-shaped), and BVPL_S (BVPL_S S-shaped). The best minimum is achieved from both the BVPL_S and the BVPL_V algorithms, and it is observed that the same optimum value is obtained. Both BVPL variants choose a small number of features compared to BPSO.

Table 3.4. Summary of the metrics in 30 runs

Algorithm	Evaluated metrics				
	Min_Fit	***Max_Fit***	***Avg_Feat***	***Avg_Best***	***SD_Best***
BPSO_V	0.00227	0.005	8.7	0.00395	0.00072
BPSO_S	0.0027	0.005	7.7	0.00351	0.00055
BVPL_V	**0.00091**	**0.00272**	2.73	**0.00135**	**0.00054**
BVPL_S	**0.00091**	**0.00172**	2.73	**0.00182**	**0.00289**

The results from this preliminary experiment shows some promising solutions on using BVPL for generating an optimal subset of features for different input datasets. These results served as indices to extent the calculations in a larger number of datasets, and in other popular metaheuristics.

3.3.2 Experiment 2

In this subsection, a more detailed comparison of BVPL for its efficiency and effectivity compared to other MHOAs its analyzed, and concluded. This experiment includes the 10

datasets related to PD, summarized in Table 3.5. The number in the brackets on the third column shows the final number of columns selected after removing columns or rows with missing values, or ID columns.

Table 3.5. Summarized information about PD datasets

Dataset name	ID	Dimension	Class
Parkinson	D1	195x23 (23)	2
HandPD spiral	D2_S	368x16 (13)	2
HandPD meander	D2_M	368x16 (13)	2
NewHandPD spiral	D3_S	264x16 (13)	2
NewHandPD meander	D3_M	264x16 (13)	2
Early biomarkers of PD based on natural connected speech	D4	130x65 (27)	3
Parkinson's Disease Classification speech-based	D5	756x754 (754)	2
Replicated acoustic features Parkinson	D6	240x48 (46)	2
Parkinson dataset with Multiple Types of Sound Recordings	D7	1040x29 (27)	2
Gait Data Arm Swing	D8	148x58 (55)	2

In Table 3.6 are presented the name of the metaheuristics, references about each algorithm, and the parameters of them.

Table 3.6. Parameters for the experiment 2

Algorithm	Reference	Parameters
General	-	nRuns = 20; maxiter = 100, population = 6; alpha_cost = 0.99, k=5-fold
BVPL	-	fall_rate=0.15, transport_rate = 0.5, β=2, b from β to 0
ACO	[270]	τ =1, eta= 1, α = 1, β= 0.1, ρ = 0.2.
ABC	[271]	Acceleration Coefficient a =1
ALO	[268]	-
ASO	[272]	V_{max} = 6, ε = 0.001, Depth weight α = 50, multiplier weight β = 0.2
BA	[273]	Loudness A = 0.25, pulse rate r = 0.1, Q_{min}=0, Q_{max}=2
DE	[274]	Crossover probability CR = 0.9
DF	[275]	D_{max}= 6
FA	[276]	Light Absorption Coefficient γ = 1, Attraction Coefficient β_0 = 2, Mutation Coefficient α = 0.2, Mutation Coefficient Damping Ratio alpha_damp = 0.98
GWO	[277]	α linearly decreases from 2 to 0, C1, C2, and C3 are random numbers
HHO	[251]	β = 1.5
MFO	[278]	a linearly decreases from -1 to -2
PSO	[279]	Cognitive factor C1 =2, Social factor C2 = 2, W_{max} = 0.9, W_{min} = 0.4, V_{max} =

		6
SSA	[280]	C2, C3 = random number]0,1[
TGA	[281]	Number of trees in first group N_1= 3, Number of trees in second group N_2 = 5, Number of trees in fourth group N_4 = 3, Tree reduction rate τ = 0.8, Parameter controls nearest tree λ = 0.5.
WOA	[282]	a decreases linearly from 2 to 0, a_2 linearly decreases from -1 to -2, r_1, r_2, p are random numbers in interval (0,1), b =1
EOA	[283]	Thres = 0.5, V = 1, a_1 =2, a_2=1, GP = 0.5
GA	[284]	Crossover rate CR = 0.8, mutation rate MR = 0.3
SCA	[285]	r_1, decreases linearly from α to 0, α= 2, r_2, r_3, r_4, are random numbers,
TLBO	[286]	-
GOA	[287]	c_{max}=1, c_{min}==0.00004

The selected algorithms are: Ant Colony Optimizer (ACO), Artificial Bee Colony (ABC), ALO, Atom Search Optimization (ASO), Bat Algorithm (BA), DE, Dragon Fly (DF), Firefly Algorithm (FA), Grey Wolf Optimization (GWO), Harris Hawk Optimization (HHO), Moth Flame Optimization (MFO), PSO, Salp Swarm (SSA), Tree Growth Algorithm (TGA), Whale Optimization Algorithm (WOA), Equilibrium Optimizer Algorithm (EOA), GA, Sine-Cosine Algorithm (SCA), Teaching Learning-Based Optimization (TLBO), and Grasshopper Optimization Algorithm (GOA) algorithms. All the metaheuristics codes have been programmed in R language, and adapted for FS from the author. The two-step binarization method has not been applied to GA, DE, and ACO as they provide themselves binary outcomes. As part fitness function for evaluations, it was used again k-NN classifier accuracy with a Euclidean distance metric and k-neighbor = 5 to measure the quality of the subset of solutions. To avoid overfitting, k-fold cross-validation with k-fold = 5 was used.

3.3.2.1 Results from the S-shaped and V-shaped Transfer Function

This subsection presents the optimal outcomes attained by BVPL utilizing eight transfer functions for each dataset. The objective was to identify the most prominent TF for the BVPL based on metrics as: average and standard deviation of fitness, average accuracy, and the average number of selected features. The criteria of selection of the best TF were according two conditions. First one is the minimum average fitness achieved for each dataset, and secondly when the average fitness is equal (for example D4), the subsequent criterion considered is the maximum average accuracy. After the evaluations, the successful transfer functions for each dataset are as follows: D1 (V3), D2_S (V3), D2_M (S2), D3_S (S3), D3_M

(S3), D4 (S2), D5 (S3), D6 (V4), D7 (S2), and D8 (V3). The selected TF are utilized for the subsequent experiments in the other metaheuristics.

3.3.2.2 Comparison of Binary Volleyball Premier League and metaheuristics

The results from the metrics for the MHOAs are presented in Tables 3.7 – 3.11. The optimal outcomes are shown through the utilization of both italics and bold formatting. In these tables, the short names are referred to the metrics as average fitness (f_{avg}), the standard deviation of the fitness (f_{sd}), average accuracy (acc_{avg}), and the average number of features ($feat_{avg}$).

In reference to D1 (Table 3.7), it can be shown that ACO outperforms BVPL in all metrics, with the exception of the average number of features. BVPL is among the third-best algorithms after ACO and GA. According to the metrics data presented in Table 3.7 for the D2_S dataset, it is evident that the BVPL algorithm ranks as the second most effective approach, surpassed only by the ACO Algorithm. The ACO algorithm demonstrates superior performance in terms of average fitness and accuracy. BVPL algorithm exhibits a high level of rivalry in terms of accuracy when compared to TGA, WOA, and GA.

Table 3.7. The results of the metrics for D1(left) and D2_S (right)

MHOA	f_{avg}	f_{sd}	acc_{avg}	$feat_{avg}$	f_{avg}	f_{sd}	acc_{avg}	$feat_{avg}$
BVPL	0.05338	0.01874	0.94723	***2.5***	0.06385	0.01782	0.93724	2.05
ACO	***0.04030***	0.02004	***0.96379***	9.95	***0.05837***	0.01429	***0.94495***	4.75
ABC	0.11009	0.02708	0.89052	2.95	0.12738	0.04765	0.87294	2.05
ALO	0.05949	0.02450	0.94310	6.75	0.08581	0.04358	0.91606	2.8
ASO	0.06754	0.02601	0.93621	9.65	0.07619	0.017780	0.92569	3.15
BA	0.07099	0.03129	0.93362	11.6	0.09661	0.02855	0.90734	5.85
DE	0.05733	0.02612	0.94741	11.6	0.07337	0.01636	0.93119	6.3
DF	0.08042	0.02547	0.92414	11.7	0.08423	0.02352	0.92064	6.8
FA	0.10744	0.03267	0.89310	3.55	0.12738	0.05275	0.87294	***1.9***
GWO	0.05761	0.02221	0.94741	12.2	0.07440	0.01922	0.93028	6.45
HHO	0.06503	0.02657	0.93707	6	0.08095	0.02206	0.92156	3.95
MFO	0.07499	0.02574	0.92586	3.5	0.09121	0.04623	0.90963	2.1
PSO	0.07726	0.02155	0.92759	12.25	0.08802	0.02536	0.91606	5.9
SSA	0.10635	0.03638	0.89483	3.05	0.17589	0.04722	0.82431	2.05
TGA	0.05684	0.02242	0.94828	12.4	0.06544	0.01638	0.93853	5.5
WOA	0.05721	0.02530	0.94483	5.7	0.06641	0.01579	0.93532	2.85
EOA	0.08080	0.03317	0.91983	9.7	0.11483	0.05546	0.88578	6.05
GA	0.04904	0.01659	0.95517	10.25	0.06713	0.01844	0.93670	5.35
SCA	0.07658	0.01947	0.92414	2.7	0.09888	0.03708	0.90180	2.1
TLBO	0.09030	0.02952	0.91035	3.4	0.12556	0.05023	0.87477	***1.9***
GOA	0.09256	0.02409	0.90862	4.6	0.10967	0.04740	0.89174	3

In relation to the D2_M dataset, as illustrated in Table 3.8, it can be established that BVPL gives the greatest average fitness, along with average accuracy and the selected features. The ACO remains highly competitive. According to the data presented in Table 3.8 (D3_S), BVPL

Algorithm exhibits substantially better outcomes in terms of average fitness and accuracy compared to ACO. Their results are better than those of the other MHOAs.

Table 3.8. The results of the metrics for D2_M (left) and D3_S (right)

MHOA	f_{avg}	f_{sd}	acc_{avg}	$feat_{avg}$	f_{avg}	f_{sd}	acc_{avg}	$feat_{avg}$
BVPL	***0.05716***	0.01763	***0.94433***	***2.45***	***0.14256***	0.03281	***0.85924***	***3.85***
ACO	0.05977	0.02083	0.94358	4.85	0.14590	0.03043	0.85823	6.8
ABC	0.10143	0.03725	0.90138	4.65	0.20743	0.02453	0.79557	5.4
ALO	0.07313	0.01890	0.93211	7.15	0.16586	0.03436	0.83987	8.6
ASO	0.06607	0.02793	0.93532	***2.45***	0.17314	0.03374	0.82848	4
BA	0.08402	0.02626	0.92064	6.55	0.17735	0.02515	0.82595	6.05
DE	0.07990	0.02709	0.92523	7.05	0.16929	0.03239	0.83481	6.9
DF	0.06998	0.01959	0.93440	6.05	0.18040	0.03475	0.82279	5.95
FA	0.11725	0.04133	0.88578	5	0.22355	0.04155	0.77911	5.85
GWO	0.07577	0.02597	0.92890	6.45	0.15396	0.03695	0.85063	7.3
HHO	0.07436	0.02302	0.92982	5.85	0.15467	0.03317	0.84873	5.9
MFO	0.11023	0.05208	0.89312	5.3	0.17004	0.03198	0.83354	6.3
PSO	0.08836	0.03218	0.91606	6.3	0.20972	0.04757	0.79367	6.55
SSA	0.15769	0.04766	0.84312	4.35	0.29933	0.06355	0.70063	5.5
TGA	0.06899	0.01860	0.93486	5.4	0.15542	0.03411	0.84937	7.55
WOA	0.06738	0.01862	0.93716	6.2	0.15609	0.02941	0.84747	6.1
EOA	0.08653	0.04672	0.91697	5.4	0.17861	0.04447	0.82468	4.45
GA	0.06915	0.02361	0.93486	5.6	0.15684	0.03629	0.84684	6.25
SCA	0.10139	0.04792	0.90184	4.45	0.17618	0.03727	0.82722	5.8
TLBO	0.08632	0.02717	0.91697	4.95	0.18345	0.04716	0.81962	5.85
GOA	0.10469	0.04931	0.89817	4.65	0.18884	0.04317	0.81392s	5.55

According to the findings presented in Table 3.9 of D3_M, the SCA method demonstrates superior performance in terms of accuracy and fitness. According to the ranking, ACO is considered the second most favorable alternative, followed by BVPL. The results from the D4 dataset, which are shown in the right of Table 3.9, show that BVPL produce better results than the other MHOAs on all of the criteria that have been looked at. ACO is the second-best one, and the others are far away from this result.

Table 3.9. The results of the metrics for D3_M (left) and D4 (right)

MHOA	f_{avg}	f_{sd}	acc_{avg}	$feat_{avg}$	f_{avg}	f_{sd}	acc_{avg}	$feat_{avg}$
BVPL	0.13584	0.02114	0.86557	***3.3***	***0.34154***	0.03427	***0.65643***	3.65
ACO	0.12869	0.02737	0.87532	6.45	0.37252	0.04392	0.62821	12
ABC	0.18324	0.03017	0.81962	5.45	0.481	0.05230	0.51795	9.55
ALO	0.16381	0.02369	0.84177	8.85	0.42017	0.03796	0.58333	19.7
ASO	0.16077	0.03347	0.84051	3.45	0.43267	0.04637	0.56410	***2.95***
BA	0.18696	0.04335	0.81646	6.3	0.43515	0.05336	0.56539	12.7
DE	0.16645	0.03761	0.83797	7.25	0.42367	0.05521	0.57821	15.85
DF	0.16327	0.02266	0.84051	6.45	0.43883	0.04639	0.56154	12.35
FA	0.22121	0.05310	0.78101	5.3	0.49485	0.04527	0.50384	9.5
GWO	0.15166	0.03789	0.85317	7.55	0.416	0.04499	0.58590	15.7
HHO	0.14385	0.02511	0.86013	6.45	0.40635	0.04903	0.59359	10.4

MFO	0.15421	0.03190	0.84937	6.1	0.41321	0.03936	0.58590	8.45
PSO	0.19677	0.03463	0.80696	6.8	0.49506	0.05337	0.50513	13.35
SSA	0.25526	0.06306	0.74494	4.7	0.46537	0.05410	0.53205	10.15
TGA	0.14953	0.02683	0.85443	6.5	0.40173	0.03892	0.6	14.9
WOA	0.14573	0.02331	0.85823	6.45	0.39673	0.03614	0.60513	15.1
EOA	0.16766	0.02287	0.83544	5.2	0.40215	0.04229	0.59744	10.9
GA	0.14707	0.02602	0.85633	5.8	0.43825	0.04543	0.56154	10.85
SCA	***0.11151***	*0.04781*	***0.89220***	5.25	0.40267	0.04265	0.59744	11.4
TLBO	0.18228	0.04361	0.82025	5.2	0.43529	0.03697	0.56410	9.75
GOA	0.17113	0.03209	0.83165	5.35	0.43435	0.04199	0.56410	7.3

The results from the seventh dataset, D5 (Table 3.10), provide further confirmation that BVPL outperforms the other MHOAs. In the D6 dataset, as presented in the right of Table 3.10, it can be observed that BVPL presents greater performance in terms of average fitness. However, ACO demonstrates higher average accuracy. ALO reveals strong concurrence with ACO in terms of metrics.

Table 3.10. The results of the metrics for D5 (left) and D6 (right)

MHOA	**f_{avg}**	**f_{sd}**	**acc_{avg}**	**$feat_{avg}$**	**f_{avg}**	**f_{sd}**	**acc_{avg}**	**$feat_{avg}$**
BVPL	***0.08522***	0.01227	***0.91593***	149.8	***0.10983***	0.02630	0.89040	5.95
ACO	0.09819	0.01492	0.90553	351.7	0.11227	0.01997	***0.89097***	19.7
ABC	0.12731	0.01741	0.87589	331.6	0.16207	0.02344	0.83819	8.2
ALO	0.10986	0.01564	0.89712	605.4	0.11520	0.01930	0.8875	16.9
ASO	0.10473	0.01292	0.89602	***134.6***	0.15249	0.02125	0.84792	8.7
BA	0.09687	0.01310	0.90708	367.7	0.15248	0.02947	0.85069	21
DE	0.10996	0.01631	0.89513	462.3	0.13757	0.02791	0.86667	25.1
DF	0.11513	0.01214	0.88872	373.3	0.17192	0.02570	0.83125	21.9
FA	0.12204	0.01559	0.88120	332.9	0.16833	0.02649	0.83194	8.8
GWO	0.09813	0.01367	0.90774	511.9	0.13015	0.02325	0.87431	25.7
HHO	0.10991	0.01650	0.89381	359.9	0.13889	0.02635	0.86181	9.4
MFO	0.10492	0.01664	0.89845	330	0.12981	0.01735	0.87083	8.7
PSO	0.12457	0.01712	0.87920	374.7	0.16834	0.02626	0.83472	21.2
SSA	0.11879	0.01712	0.88252	336.6	0.15796	0.02849	0.84306	8.2
TGA	0.10641	0.01481	0.89867	459.3	0.13511	0.01844	0.86944	26.4
WOA	0.10327	0.01346	0.90155	436.9	0.13913	0.02421	0.86042	4.25
EOA	0.10829	0.01556	0.89513	295.1	0.12934	0.01938	0.87083	16.45
GA	0.10979	0.01378	0.89381	350.5	0.13516	0.02037	0.86806	20.4
SCA	0.10739	0.01431	0.89624	351.3	0.12829	0.02698	0.87153	***3.4***
TLBO	0.11524	0.01484	0.88805	331.9	0.14520	0.02107	0.85486	6.8
GOA	0.11153	0.01648	0.89071	251.2	0.15037	0.02646	0.85070	11.5

Table 3.11 represents the results for the D7, and D8 dataset. It can be observed that BVPL indicates weak efficacy, while ACO yields superior outcomes in terms of average fitness and accuracy. In the last dataset (Table 3.11), it can be observed that BVPL implies better performance across all measures, with the exception of the number of features, where SCA

displays the most positive outcomes. For the same dataset, ACO and ALO indicate a considerable level of similarity.

Table 3.11. The results of the metrics for D7 (left) and D8 (right)

MHOA	f_{avg}	f_{sd}	acc_{avg}	$feat_{avg}$	f_{avg}	f_{sd}	acc_{avg}	$feat_{avg}$
BVPL	0.32260	0.01530	0.67564	***3.85***	***0.15493***	0.03445	***0.84429***	4.2
ACO	***0.29192***	0.01316	***0.71026***	13.5	0.16540	0.04240	0.8375	24.6
ABC	0.33930	0.01734	0.66122	10	0.24681	0.05044	0.75227	8.1
ALO	0.30902	0.01684	0.69615	21.35	0.16177	0.05892	0.83977	17.1
ASO	0.31488	0.01947	0.68478	7.3	0.22156	0.05046	0.77727	5.7
BA	0.31183	0.02068	0.69022	13.4	0.21408	0.05738	0.78864	26.1
DE	0.31067	0.01824	0.69311	17.8	0.20351	0.05208	0.8	29.8
DF	0.31903	0.01359	0.68301	13.6	0.25707	0.05711	0.74546	27.4
FA	0.34018	0.01799	0.66010	9.6	0.24349	0.05011	0.75568	8.7
GWO	0.30059	0.01698	0.70272	16.4	0.19273	0.04837	0.81136	32.3
HHO	0.30341	0.01564	0.69792	11.3	0.21054	0.05312	0.78864	6.95
MFO	0.31164	0.01737	0.68958	11.3	0.17367	0.04461	0.82614	8.4
PSO	0.32716	0.02074	0.67468	13.3	0.25237	0.05098	0.75	26.3
SSA	0.33484	0.02285	0.66555	9.9	0.24825	0.05944	0.75114	8.3
TGA	0.30115	0.00596	0.70401	21.1	0.21772	0.01143	0.78523	27.5
WOA	0.29696	0.01591	0.70657	16.8	0.18185	0.04795	0.81705	3.9
EOA	0.31606	0.01780	0.68510	9.7	0.18704	0.04309	0.8125	21.4
GA	0.31019	0.01461	0.69183	13.3	0.20466	0.04393	0.79773	23.8
SCA	0.31166	0.01995	0.68942	12.2	0.18071	0.04426	0.81818	***3.35***
TLBO	0.31445	0.01843	0.68638	10.3	0.22857	0.03077	0.77046	7.1
GOA	0.32323	0.01769	0.67772	10.9	0.22477	0.04204	0.775	10.9

It can be observed that BVPL produces a smaller number of features compared to the other methods, particularly in the cases of D1, D2_M, D3_S, D3_M, and D7.

3.3.2.3 Convergences curves and statistical difference

The convergence curves can visually illustrate the variations in the performance of all the MHOAs across different criteria. Figures 3.6 and 3.7 illustrate the convergence curves that correspond to the average fitness observed throughout each iteration. Each graph shown represents a distinct dataset. It can be observed that among the three datasets, namely D2_M, D4, and D5, the BVPL algorithm exhibits a quicker convergence speed compared to the other metaheuristics. Moreover, in the cases of D2_S, D3_M, D3_S, D6, and D8, the level of competitiveness of BVPL is notably high. Additionally, it is observed that BVPL in D8, D6, and D3_S exhibit a faster rate of convergence compared to the other MHAs, after the 75th iteration

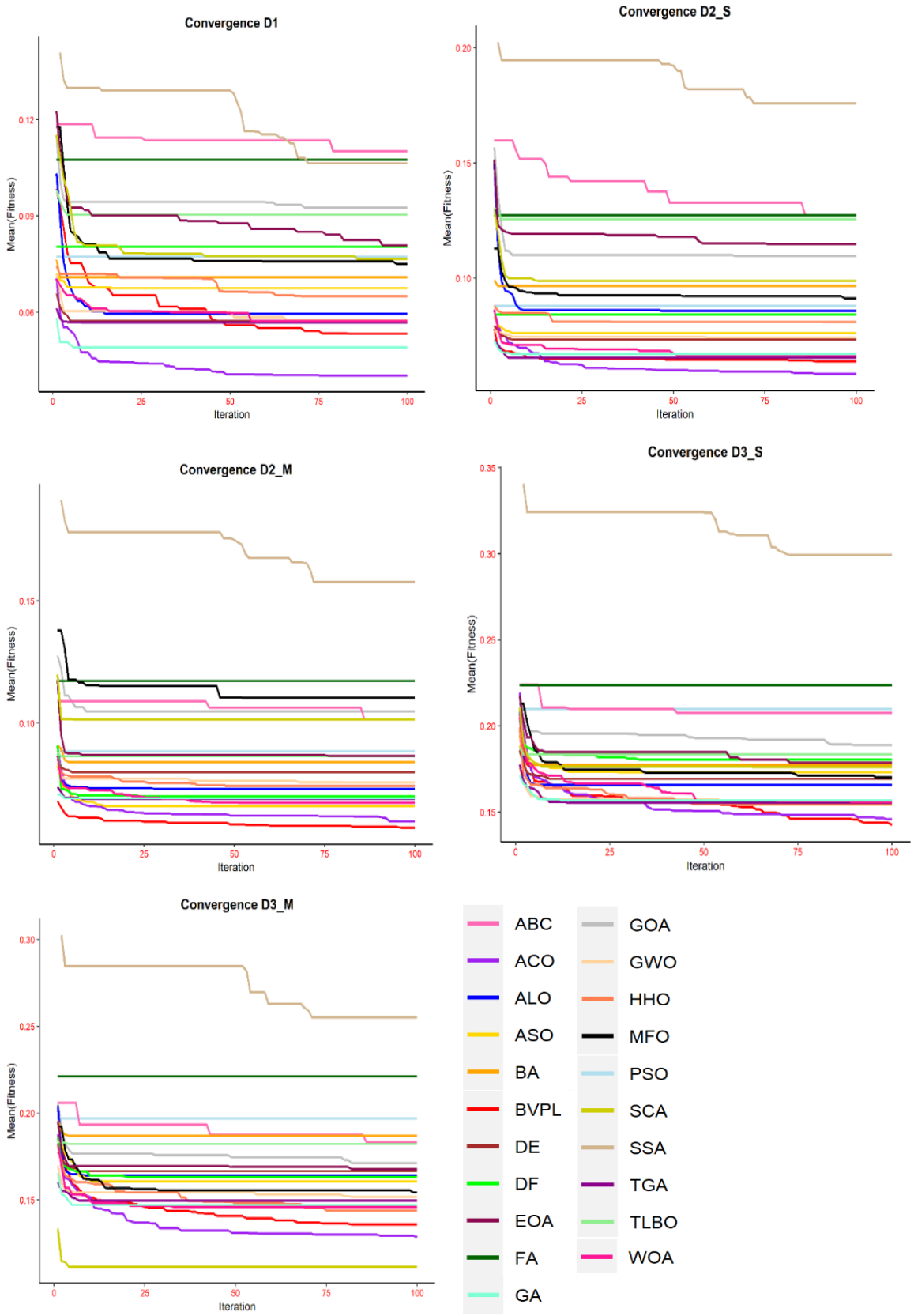

Figure 3.6. The convergence curves of BVPL versus the other MHOAs for the first five datasets

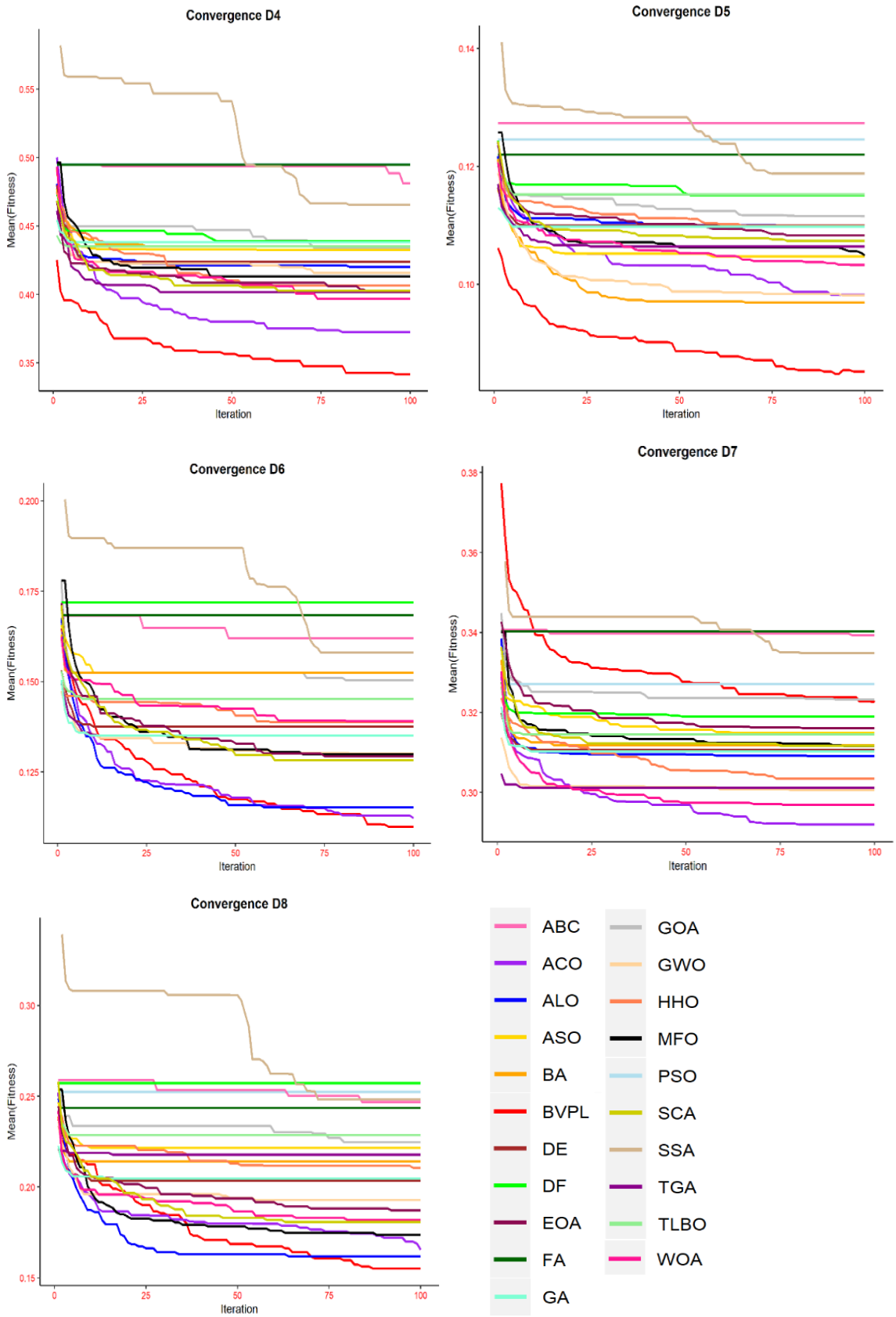

Figure 3.7. The convergence curves of BVPL versus the other MHOAs for the last five datasets

Statistical tests are utilized to assess the significance of the solutions provided by BVPL_BALO in comparison to those produced by other MHOAs. Here the significance is

measured combining the parametric t-test and the non-parametric Wilcoxon sum-rank test. The independent t-test is employed when the variances are homogenous and the fitness values are normally distributed, otherwise the Wilcoxon sum-rank test is utilized. The null hypothesis being assessed is that there is no difference in means (medians) in terms of optimum fitness between the hybrid metaheuristic and the other MHOAs. The proposed algorithm's fitness differs substantially from the compared methods if the p-value is less than 0.05. These tests have been conducted to examine the difference in average fitness between BVPL and the remaining 20 MHOAs. If the p-value is less than 0.05, it can be concluded that there is a significant difference in the average fitness of BVPL when compared to the other algorithm. The performance of BVPL surpasses that of 17 MHOAs, exhibiting superior results in over 50% of the datasets. The performance of BVPL does not appear to be superior to ACO, except in the case of four specific datasets. Additionally, when considering the performance of WOA and GA, they are found to be equally superior to BVPL.

To summarize, the metaheuristic BVPL in feature selection problem, has provided a higher accuracy in predicting PD, in 10 different datasets, compared with a large list of metaheuristics. BVPL outperforms ACO in fitness and accuracy across five datasets, while ACO outperforms in one. SCA outperforms ACO in one dataset, with the lowest values across all datasets. The BVPL algorithm demonstrates an acceptable speed of convergence and effectiveness in searching across a wide range of datasets, consistently ranking among the top three among other MHOAs, and superior in three of them.

3.4 Results on improving the effectivity of Binary Volleyball Premier League

Therefore, in this paragraph are presented the results from applying OBL into BVPL, and the new hybrid metaheuristic BVPL_BALO which are used to improve on the effectivity of BVPL in predicting Parkinson. The datasets, and experiment settings are the same as in the experiment 2.

3.4.1 Results for Opposition-based learning Binary Volleyball Premier League algorithm

Table 3.12 provides a summary of the four metrics, which will compare for BVPL against OBL_BVPL.

Table 3.12. Results BVPL against OBL_BVPL

Algorithm	**BVPL**				**OBL_BVPL**			
Dataset	$\mathbf{f_{avg}}$	$\mathbf{f_{sd}}$	$\mathbf{acc_{avg}}$	$\mathbf{feat_{avg}}$	$\mathbf{f_{avg}}$	$\mathbf{f_{sd}}$	$\mathbf{acc_{avg}}$	$\mathbf{feat_{avg}}$
D1	0.05338	0.01874	0.94723	2.5	0.01848	0.00539	0.98274	3.05
D2_S	0.06385	0.01782	0.93724	2.05	0.05601	0.00747	0.94654	3.7
D2_M	0.05716	0.01763	0.94433	2.45	0.05082	0.00864	0.95216	4.15
D3_S	0.14256	0.03281	0.85924	3.85	0.10058	0.01898	0.90241	4.75
D3_M	0.13584	0.02114	0.86557	3.3	0.09235	0.00385	0.91013	4.05

D4	0.34154	0.03427	0.65643	3.65	0.30865	0.01973	0.68971	3.8
D5	0.08522	0.01227	0.91593	149.8	0.08392	0.01209	0.91728	152.3
D6	0.10983	0.02630	0.89040	5.95	0.08794	0.01235	0.91249	5.85
D7	0.32260	0.01530	0.67564	3.85	0.28404	0.01429	0.71523	5.5
D8	0.15493	0.03445	0.84429	4.2	0.11701	0.02745	0.88298	6.3

The suggested technique exhibits significant enhancements in terms of average fitness and accuracy across all datasets. Incorporating the opposing approach leads to a considerable increase in accuracy and a decrease in fitness. The observed improvement in accuracy ranges from 0.135% in D5 to 4.456% for D3_M. In relation to efficacy, this technique has demonstrated major relevance in the prediction of Parkinson's disease. Opposition-based learning can enhance the prediction of Parkinson above 90% in 7 out of 10 datasets.

3.4.2 Results from Binary Volleyball Premier League and Antlion Optimizer metaheuristic algorithm

In this subsection, the main goal is to apply and validate the hybrid metaheuristic algorithm BVPL_BALO in the same 10 Parkinson datasets, and to compare with some prominent metaheuristics which provided the better results in section 3.3.2.

3.4.2.1 Comparison of the hybrid metaheuristic vs other metaheuristics

Table 3.13 presents results of the performance metrics related to average fitness (avg(fit)), standard deviation of the fitness (sd(fit)), average accuracy (avg(acc)), and average of selected features (avg(feat)).

Table 3.13. The results of the metrics for the hybrid against other metaheuristics

Dataset	MHOA	Avg(fit)	Sd(fit)	Avg(acc)	Avg(feat)
D1	BVPL	0.053379	0.018743	0.947230	**2.5**
	BALO	0.059486	0.024495	0.943103	6.75
	BVPL_BALO	**0.014506**	**0.007155**	**0.987115**	3.85
	BACO	0.040299	0.020041	0.963793	9.95
	BSCA	0.076581	0.019466	0.924138	2.7
D2_S	BVPL	0.063845	0.017816	0.937235	**2.05**
	BALO	0.085814	0.043584	0.916055	2.8
	BVPL_BALO	**0.049251**	**0.008463**	**0.954587**	5.15
	BACO	0.058370	0.014289	0.944954	4.75
	BSCA	0.098881	0.037083	0.901803	2.1
D2_M	BVPL	0.057158	0.017634	0.944327	**2.45**
	BALO	0.073128	0.018895	0.932110	7.15
	BVPL_BALO	**0.040951**	**0.011221**	**0.962718**	4.85
	BACO	0.059774	0.020828	0.943578	4.85
	BSCA	0.101392	0.047918	0.901835	4.45
D3_S	BVPL	0.142560	0.032805	0.859241	**3.85**
	BALO	0.165859	0.034362	0.839873	8.6
	BVPL_BALO	**0.080607**	**0.000689**	**0.924051**	6.5
	BACO	0.145896	0.030426	0.858228	6.8
	BSCA	0.176182	0.037267	0.827215	5.8

D3_M	BVPL	0.135835	0.021138	0.865570	**3.3**
	BALO	0.163812	0.023694	0.841772	8.85
	BVPL_BALO	**0.071963**	**0.006467**	**0.931519**	5
	BACO	0.128687	0.027367	0.875316	6.45
	BSCA	0.111512	0.047812	0.892201	5.25
D4	BVPL	0.341538	0.034274	0.656430	**3.65**
	BALO	0.420173	0.037956	0.583333	19.7
	BVPL_BALO	**0.3095**	**0.026907**	**0.689763**	6.15
	BACO	0.372519	0.043923	0.628205	12
	BSCA	0.402673	0.042648	0.597436	11.4
D5	BVPL	**0.085218**	0.012270	**0.915930**	**149.75**
	BALO	0.109861	0.015638	0.897124	605.35
	BVPL_BALO	0.087538	**0.011656**	0.915503	292.65
	BACO	0.098193	0.014918	0.905530	351.7
	BSCA	0.107392	0.014310	0.896239	351.25
D6	BVPL	0.109825	0.026297	0.890401	5.95
	BALO	0.115197	0.019299	0.8875	16.9
	BVPL_BALO	**0.075396**	**0.018124**	**0.925683**	8.2
	BACO	0.112271	0.019972	0.890972	19.7
	BSCA	0.128288	0.026980	0.871528	**3.4**
D7	BVPL	0.322596	0.015304	0.675641	**3.85**
	BALO	0.309019	0.016842	0.696154	21.35
	BVPL_BALO	**0.269404**	**0.012301**	**0.735897**	20.65
	BACO	0.291923	0.013163	0.710256	13.45
	BSCA	0.311663	0.019952	0.689423	12.2
D8	BVPL	0.154931	0.034449	0.844290	4.2
	BALO	0.161773	0.058918	0.839773	17.1
	BVPL_BALO	**0.078278**	**0.030837**	**0.922727**	9.6
	BACO	0.165403	0.042396	0.8375	24.6
	BSCA	0.180713	0.044262	0.818182	**3.35**

[a.] The underlined and bold values show the better metrics

The data reveals that the hybrid approach exhibits superior performance compared to the other methods across 90% of the datasets, as seen by its higher values for maximum, average, and standard deviation of fitness, as well as average accuracy. In relation to the average number of features, it is seen that the hybrid approach exhibits an increase in the number of features, in contrast to BVPL, which yields a lower feature ratio in 80% of the datasets. The utilization of BVPL_BALO in the D5 dataset does not produce any noticeable improvement. The final accuracy obtained from the hybrid exceeds 90% in most of the datasets, which is an adequate result in predicting PD. However, for datasets D4 and D7, the accuracies remain somewhat low, despite the hybrid approach managing to increase them by approximately 3.33% and 6.03% respectively from BVPL. Additionally, it has been shown that the BACO algorithm outperforms BVPL in terms of accuracy for almost half of the datasets.

3.4.2.2 Convergence curves

For a complete idea of how fitness changes in each iteration of each run for the MHOAs, the respective convergence curves for the 5 metaheuristics are presented. Figure 3.8, and 3.9 shows the graphical illustration of the convergence curves for the five metaheuristics together.

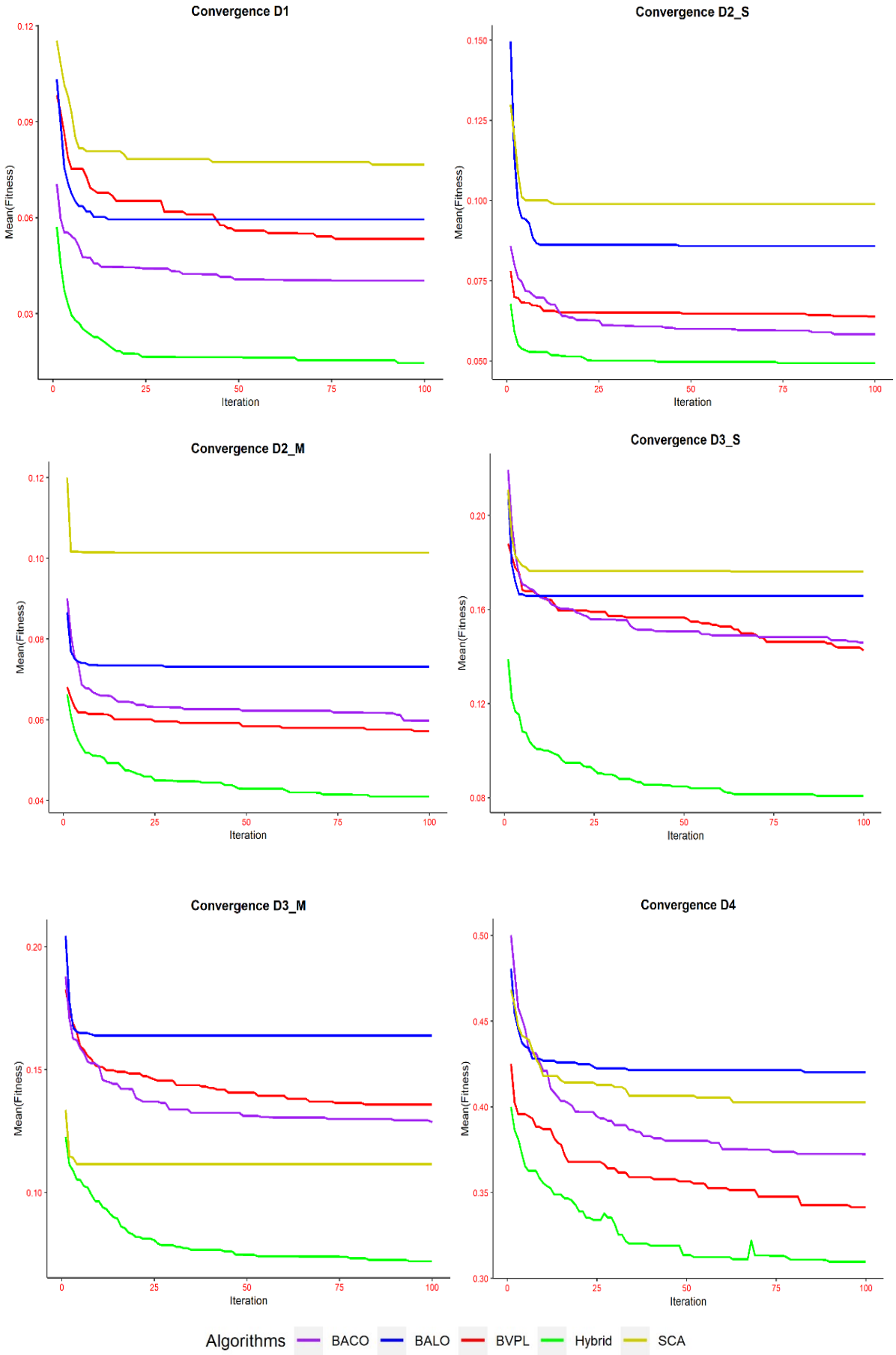

Figure 3.8. The convergence curves of the metaheuristics for the first 6 Parkinson's datasets

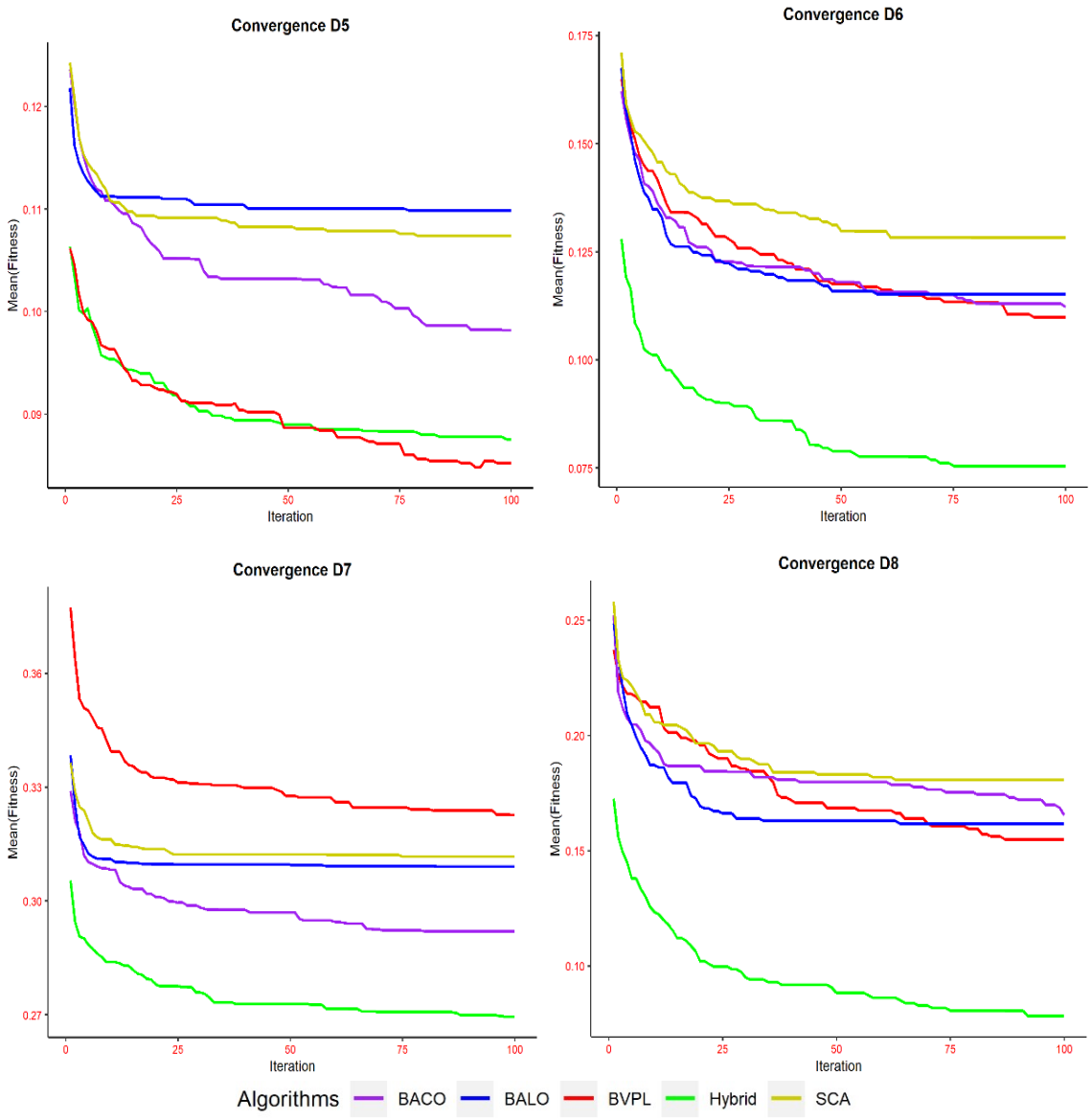

Figure 3.9. The convergence curves for the other Parkinson's datasets

Based on the data presented in the charts, it can be observed that the hybrid shows consistently lower fitness values in each iteration, except for the D5 dataset. This means that the algorithm has shown efficacy since the first iterations. Regarding the BVPL, it has provided the best convergence for one dataset and is the second best for 5 datasets out of 10. All the experiments show that the hybrid BVPL_BALO converges faster to the optimum for nine datasets.

3.4.2.3 Statistical tests results

The provided p-values generated by the Wilcoxon sum-rank test or t-test for the comparison between the average fitness of BVPL_BALO and four other MHOA are calculated. The only case where BVPL_BALO doesn't show a significant difference with BVPL is for the D5 dataset, and is highlighted (p-value >0.05).

The newly proposed metaheuristic demonstrates superior performance compared to all other existing MHOAs across 90% of the datasets. The experimental results show that BVPL_BALO is more competitive in terms of converging faster to the optimum, and in predicting with a higher accuracy the PD. The convergence plots and statistical tests both support these results.

3.5 Results on improving the efficiency of the hybrid Binary Volleyball Premier League and Antlion Optimizer metaheuristic algorithm

3.5.1 Results after integrating the occurrence list in the cost function

In this subsection, is evaluated the effect that has integrating the occurrence list in the hybrid metaheuristic BVPL_BALO. A comparative analysis was conducted to assess the execution times of BVPL_BALO vs the other metaheuristics. The aim was to evaluate the reduction in time achieved by BVPL and to determine the magnitude of variations between the other algorithms. The D5 dataset has been incorporated for this test due to its higher number of dimensions. The final results are shown in Table 3.14.

Table 3.14. Execution time for D5 dataset

MHOAs	Time
BVPL	7.4888 days
BALO	2.8644 hours
BVPL_BALO	2.5463 days
BSCA	1.3866 hours
BACO	1.4925 hours

It is observed that the execution time of BVPL is significantly reduced when BVPL_BALO is employed. The other metaheuristics offer more promising computational times in relation to BVPL when used in this particular dataset. The proposed improvement in the hybrid technique leads to a decrease of execution time of approximately 2.94 times; yet, this decrease remains unsatisfactory.

3.5.2 Results after using cosine similarity and the hybrid metaheuristic algorithm

The interpretation of the results is given with three different comparisons. Initially, the proposed technique is implemented on the high-dimensional Parkinson dataset, D5. Then it is applied on the other datasets in order to test the effect of the method, and finally an observation of the similarity of the features between the proposed method, and when applying only the hybrid metaheuristic algorithm.

3.5.2.1 Results for D5 dataset

In the proposed method, the phase 1 concludes with ranking the features according to their importance, and were selected the top 2.5%, 5%, 10%, 15%, 20%, 25%, 30%, and 50% of

features for extraction. The selection process for the percentage was driven by the goal of minimizing the execution time of BVPL_BALO. To achieve this, a brute force method was employed to identify the optimal percentage of features that would produce comparable results when using the entire dataset. The measurement results for each feature include the execution time, average (avg), standard deviation (sd) of fitness (fit), average accuracy (acc), and average number of selected features (feat). The new method compares metrics from the dataset with all features (D5_hybrid100%) to those from the dataset with fewer features. The size of rows and features in the dataset influences the initial ranking time, which is approximately 5 days. However, you only need to perform this procedure once, and you can choose different percentages without repeating the ranking. The results are shown in Table 3.15.

Table 3.15. The results of the metrics for each percent of selected features

Dataset	Performance metrics				
	Time(h)	***Avg(fit)***	***Sd(fit)***	***Avg(acc)***	***Avg(feat)***
D5_2.5%	1.2244 h	0.2149	0.0093	0.7852	4.05
D5_5%	6.3734 h	0.1637	0.0106	0.8374	10.1
D5_10%	9.7011 h	0.1325	0.0104	0.8685	17.8
D5_15%	12.2934 h	0.1204	0.0106	0.8805	23.55
D5_20%	13.0052 h	0.1141	0.0128	0.8870	32.65
D5_25%	16.7052 h	0.1027	0.0140	0.8985	40
D5_30%	17.8223 h	0.1024	0.0107	0.8987	48.4
D5_50%	**26.7905 h**	**0.0863**	**0.0129**	**0.9150**	**81.35**
D5_hybrid100%	61.1112 h	0.0875	0.0117	0.9155	292.65

The calculations clearly show that extracting 50% of the features yields metrics that are exactly the same as when the BVPL_BALO uses the full feature (original) dataset as input. At the same time, the execution time was reduced by 56.16% and the number of selected features minimized by 72.2%.

3.5.2.2 Results for the other datasets

The identical suggested method was implemented on the remaining datasets, but with a focus on only 50% of the most significant features, since it provided better results for the D5 dataset. This allows for a comparison between this subset of features (referred to as D$_50%) and the complete set of features (D$_Hybrid100%). Table 3.16 summarizes the experiment's findings.

Table 3.16. The metrics for 50% and 100% of the features

Dataset	Performance metrics				
	Time(m)	***Avg(fit)***	***Sd(fit)***	***Avg(acc)***	***Avg(feat)***
D1_50% D1_Hybrid100%	11.7123 63.924	0.0251 0.0145	0.0093 0.0072	0.9793 0.9871	5.05 3.85
D2_S50% D2_S_Hybrid100%	2.9290 11.1710	0.0851 0.0493	0 0.0085	0.9174 0.9546	2 5.15
D2_M50%	3.2653	0.0836	0.0012	0.9229	4.4

D2_M_Hybrid100%	11.0229	0.0410	0.0112	0.9627	4.85
D3_S50% D3_S_Hybrid100%	3.8261 16.1578	0.1804 0.0806	0 0.0007	0.8228 0.9241	3 6.5
D3_M50% D3_M_Hybrid100%	3.3871 12.2148	0.2289 0.0720	0 0.0065	0.7721 0.9315	2 5
D4_50% D4_Hybrid100%	41.0500 279.594	0.3407 0.3095	0.0307 0.0269	0.6590 0.6898	4.05 6.15
D6_50% D6_Hybrid100%	136.719 415.478	0.1011 0.0754	0.0152 0.0181	0.8997 0.9257	4 8.2
D7_50% D7_Hybrid100%	20.8601 247.531	0.2923 0.2694	0.0128 0.0123	0.7090 0.7359	5.4 20.65
D8_50% D8_Hybrid100%	253.810 425.641	0.1030 0.0783	0.0180 0.0308	0.8977 0.9227	4.7 9.6

In the D1 dataset, the results provided by the extracted 50% of the features are very similar to the case of 100% of the features. The average accuracy decreases by 0.8%, the number of features increases by approximately 31.17%, and the time decreases by 81.68%. The D2_S dataset exhibits a decrease in average accuracy of 3.9%, a decrease in the number of features of about 61.17%, and a decrease in time of 73.78%. The D2_M dataset exhibits a decrease in average accuracy of 4.13%, a reduction in features of about 9.28%, and a decrease in time of 70.38%.

The D3_S dataset shows a 10.96% decrease in average accuracy, a 53.85% decline in the number of features, and a 76.32% decrease in time. In the D3_M dataset, the average accuracy has decreased by 17.11%, the number of features has fallen by 60%, and the time has decreased by 72.27%. Both of these datasets experience a significant decrease in the number of features and execution time, but their accuracy is greatly impacted.

The D4 dataset shows a decrease in average accuracy by 4.47%, a reduction of features by 34.15%, and a decrease in time by 85.32%. Within the D6 dataset, there is a decrease in accuracy by 2.81%, a reduction in the number of features by 51.22%, and a decrease in time by 67.09%. The D7 dataset shows a 3.66% decrease in average accuracy, a 73.85% reduction in features, and a 91.57% decrease in time. The D8 dataset shows a 2.71% decrease in average accuracy, a 51.04% drop in the number of features, and a 40.37% decrease in time.

The proposed strategy significantly reduces the execution time in all datasets, with a maximum impact of 4.47% on the accuracy of the selected number of features for seven datasets. A change of approximately 10.96% and 17.11% significantly influences the accuracy in the remaining two datasets, while improving the execution time and selected features. An explanation could be that some datasets require a larger number of features in order to achieve a satisfactory level of accuracy. Generally, the results show that choosing 50% of the features is not always a guarantee that the results will be the same as for the full feature dataset.

3.5.2.3 Results on similarity

Lastly, at this experiment, 50% of the ordered features extracted using cosine similarity are compared to the top 50% of the most selected features when the hybrid metaheuristic is applied to the whole feature dataset. The objective is to examine whether there is a convergence of features between both methods. Table 3.17 contains the results of this comparison. The first column stores the similarity of the features, the second stores the dimensions of the datasets, and the third one compares the accuracy difference between BVPL_BALO with 100% of the features vs 50% of the features.

Table 3.17. The similarity for each dataset

Dataset	Similarity		
	% Similarity	***Dimensions***	***Difference in accuracy***
D1	45.5%	23	0.8%
D2_S	50%	13	3.9%
D2_M	33.3%	13	4.13%
D3_S	50%	13	10.96%
D3_M	33.3%	13	17.11%
D4	53.8%	27	4.47%
D5	52.9%	754	0.05%
D6	48.9%	46	2.81%
D7	53.8%	27	3.66%
D8	48.1%	55	2.71%

This comparison aims to illustrate how accuracy changes with the number of dimensions and the similarity of features. The majority of the datasets have a similarity range of 45.5 % to 53.8% among their features from the two comparisons, except for two datasets that have a similarity of 33.3%. There is no evident association between dimensions or similarities that affects the difference in accuracy. This approach appears to be useful in certain datasets, such as D1 and D5 dataset.

The proposed method offers significant advantages in terms of reducing the execution time, and number of features. However, there is a trade-off in terms of accuracy. The similarity approach does not guarantee that the features will be identical, or nearly identical, in the case of using only the hybrid metaheuristic as a feature selection method. The user can alter the percentage of extracted features to determine the number of crucial features that are important to the dataset.

The suggested method can be advantageous in scenarios where prioritizing execution time is more crucial than maintaining relatively high accuracy. These two approaches are not interchangeable but provide two perspectives for choosing the most significant features based on two criteria: execution time and accuracy. The proposed method can be applied on other

low-to-high dimensional datasets. The first phase of ranking the features could also be combined with other metaheuristics besides BVPL_BALO.

3.6 Chapter conclusions

In this chapter, the novel metaheuristics algorithms, and methods used in the FS problem for forecasting Parkinson presented on Chapter 2, are applied and validated. The experiments were performed in a PC with an Intel (R) Core (TM) i5-8365U CPU @ 1.60 GHz and 1.90 GHz, 16 GB of RAM. All the codes are written and executed in RStudio environment.

- First, a literature review was conducted about the use of metaheuristics in predicting PD based on seven public datasets. The results indicate the frequent use of the particle swarm optimization algorithm, the k-nearest neighbor classifier, the accuracy metric, and the D1 dataset. 10-fold cross-validation and the Wilcoxon sum-rank test are the most frequently used. The structure of publications reveals that even with identical data, variations in metaheuristics, classifiers, hyper-parameter optimization, performance indicators, and fitness evaluation can yield seemingly superior results.
- Second, it is proposed a comparative assessment employing three distinct filter methods, three wrapper methods, and GA to identify the most crucial features for PD prediction in D1 dataset. This comparative analysis also included the GSA to optimize the parameters of each of the three classifiers. Using GSA or not, Joint Mutual Information and k-NN provided the best accuracy (98%). The combination of GA and k-NN achieves the best results without using GSA (acc = 95%). Moreover, k-NN gives a better prediction of accuracy = 97%, sensitivity = 96%, specificity 100%, and precision 100% when using GSA and k-NN for the dataset with the full features.
- Next, a binary VPL metaheuristic algorithm for FS is proposed for the first-time using two-step binarization (S-shaped and V-shaped functions, complement, and standard method). Initially, it is applied to the same Parkinson dataset, D1, and the results are compared with the binary PSO. This model generates very low fitness values for minimum, maximum, average, and standard deviation, as well as a lower number of features. This is a first attempt to evaluate the suitability of binary VPL in FS. Additional experiments must be provided.
- The next section presents a detailed comparative analysis to estimate the effectiveness and efficiency of BVPL. This analysis includes 20 MHOAs, 10 Parkinson's datasets, 5 S-shaped and 5 V-shaped TF, an analysis of convergence, and a statistically measured change in fitness. This second experiment demonstrated BVPL's strong competitiveness

with most of the MHOAs, with binary ACO being the most competitive with him. BVPL's convergence speed increases with the number of iterations, and in three datasets, it converges more quickly than the others. In four datasets, it predicts Parkinson's with an accuracy greater than 90%. The experiment revealed the need to improve the accuracy of BVPL's Parkinson prediction and address its lengthy execution time, particularly for the high-dimensional Parkinson dataset D5 (754 features).

- There are two proposed improvements to the BVPL's effectivity in FS. The first enhancement to BVPL involves integrating OBL solutions as a technique to search for a better solution than the one BVPL ultimately provides. In this way, incorporating OBL improved the accuracy of predicting Parkinson above 90% in 7 out of 10 datasets. The next improvement is integrating a binary ALO into the learning phase of BVPL in order to select the better solution provided by each of them. The conditions of the experiment have not changed, which confirms that the hybrid BVPL_BALO performs better than all other MHOAs (binary VPL, ALO, ACO, and SCA) in 90% of the datasets. The experimental results demonstrate that BVPL_BALO is more competitive in terms of converging faster to the optimum and predicting the Parkinson with higher accuracy. The convergence plots and statistical tests both support these results.
- Apart from effectivity, there are also two proposed improvements to the efficiency of the BVPL on FS related to execution time. Most BVPL phases include the calculation of team costs, which leads to an increase in BVPL execution time, particularly in high-dimensional datasets. Therefore, the proposed hybrid metaheuristic BVPL_BALO first integrates it into an occurrence list, storing the teams and their corresponding cost function values. The proposed improvement in the hybrid technique leads to a decrease in execution time of approximately 2.94 times compared to BVPL for D5 dataset. The second one concentrates on enhancing the BVPL_BALO's execution time in the 10 Parkinson datasets, primarily targeting the high-dimensional dataset D5. It employs a method that integrates cosine similarity to assess the significance of each feature in the input dataset and subsequently ranks them. After that, in the second phase, the user pre-defines a percentage of selected features and gives them as an input to the hybrid BVPL_BALO, which will select from them the most important features. This solution offers significant advantages in terms of reducing the execution time for future computations, with a range of improvement from 40.37% to a maximum of 91.57%. Additionally, it reduces the number of features within a range of 9.28% to 73.85%. However, there is a trade-off in terms of accuracy, ranging from a minimal decrease of

0.05% to a significant decrease of 17.11%. Moreover, when using cosine similarity, the ranking of the top 50% significant features generally produces an average similarity range of approximately 47%, as opposed to solely utilizing BVPL_BALO in the full features dataset.

4. Conclusions - a summary of the results obtained

Conclusions of the thesis

This thesis has examined, analyzed, and suggested novel algorithms, techniques, and methods to address the feature selection issue based on metaheuristic optimization algorithms, with a specific emphasis on predicting Parkinson's. The thesis provides a concise overview of the significance of employing metaheuristic optimization techniques for feature selection, integrating them with machine learning, and extending this field with novel optimization strategies.

Firstly, the dissertation surveys and analyzes the trend of using metaheuristics for feature selection based on Parkinson's, with the results confirming the popularity of combining metaheuristic optimization algorithms with machine learning algorithms. Moreover, a comparative analysis is conducted in order to assess the importance and suitability of integrating filter and wrapper methods, including the genetic algorithm, together with integrating the heuristic generalized simulated annealing for hyper parameter optimization. The results confirmed that each feature selection method has its importance in feature selection, and choosing them is dependent on different conditions. Hyper parameter optimization is very helpful in improving the accuracy of predicting Parkinson’s.

Improved strategies for solving the feature selection problem focusing only on metaheuristics are proposed using a metaheuristic optimization algorithm called the "binary Volleyball Premier League Optimization” algorithm. The "binary Volleyball Premier League Optimization" algorithm, not being proposed before in feature selection, has undergone a redesign to effectively tackle this binary problem. The binary VPL has given very good results in accuracy for predicting Parkinson, a high convergence speed to find the global optimum, and better results in fitness and the number of selected features than the majority of the other metaheuristics on most of the datasets.

Two improvements are proposed for enhancing the accuracy, the optimal solution, convergence, and exploitation of the binary volleyball Premier League algorithm, in other words its effectivity. The first one is the integration of an opposite-based learning technique, which contributes to achieving a better optimum, faster convergence on the optimum, and a

predictive capability of more than 90% for Parkinson's in 70% of the datasets. The second improvement on effectivity for binary Volleyball Premier League, is a new hybrid metaheuristic algorithm which employs the Antlion Optimizer algorithm in the binary volleyball Premier League in order to generate a hybrid BVPL_BALO, which enhances the learning phase of the binary VPL. ALO has very competitive results in terms of improved exploration, local optima avoidance, exploitation, and convergence. As a result, BALO improves binary VPL's final solution. The experimental results show that BVPL_BALO is more competitive in terms of converging faster to the optimum and predicting Parkinson with higher accuracy.

This thesis proposes two improvements using BVPL_BALO for improving its efficiency. The first is to store the results of each generated team's fitness in a list called the occurrence list which help in not doing a recalculation when the same solution is found. The second method involves two phases. Firstly, it ranks the features based on their significance, using cosine similarity as a measure for the distance between the dataset's rows after removing each feature individually. Next, the hybrid metaheuristic is applied to a defined number of features for feature selection. Both approaches improve the execution time by a considerable amount.

Limitations of the study

This thesis explores potential enhancements to the VPL metaheuristic algorithm first employed in FS. While this thesis work shows progress in using metaheuristics for feature selection, it did not fully overcome some limitations.

- A notable shortcoming of binary VPL is the extended duration of execution, requiring around 7.4888 days for a high-dimensional dataset (754x756). This thesis introduces two notable enhancements that significantly decrease the time required for the task, specifically 61.1112 hours and 26.7905 hours. The computer's processor and CPU time primarily influence this longer duration. Thus, enhancing the outcomes can be achieved by utilizing a more robust computer and implementing parallelization techniques for the binary VPL or other combinations of feature selection methods.
- The binary VPL algorithm required a large execution time; therefore, the number of independent runs was 20, and the number of iterations was 100, but larger values could enforce the stability of the final solutions.
- Moreover, for the reasons mentioned in the previous paragraph, only one classifier, k-nearest neighbor, is used for evaluating the quality of the final solutions, and in the future, other supervised learning ML algorithms could be used for evaluating the effectiveness of the binary VPL.

- In terms of metrics, the fitness function we use solely relies on the accuracy metric, which is influenced by imbalanced datasets. Other metrics, like F-score, can be integrated in the future in the fitness function, or other suggested fitness functions could be integrated.

Future research

The implementation of the binary VPL and other modifications have produced positive and helpful outcomes in the feature selection problem. There has been a significant advancement in forecasting Parkinson's integrating metaheuristics for selecting the optimal subset of features. There are some directions that could be followed for future research:

- The proposed binary VPL, hybrid BVPL_BALO, and other improvements of it includes a lot of random generated numbers, and solutions as chaotic maps, and others can improve more the generated solutions, and the proposed features, as it is shown in the work of [264].
- The proposed metaheuristic algorithm, BVPL, and other improvements could be used in other data related with Parkinson to identify important features with higher accuracy.
- The proposed methods and algorithms are tested only on Parkinson's data but it is not limited its application on other fields. Various sectors such as banking, healthcare, genetics, climatology, marketing, e-commerce, and network traffic, which collect substantial amounts of data, can be used to demonstrate the efficacy of this model.
- Even the innovative methods, and algorithms reduce the execution time, especially for datasets with medium to high dimensions, it is critical to carefully consider the limitations of BVPL that result from the large number of steps in VPL and fitness evaluations required in most of the phases of BVPL which is strongly affected by the number of features of the input dataset. Hence, it is advisable to explore alternative combinations of feature selection, and feature importance methods, in conjunction with BVPL to reduce the largest execution time.

Thesis contributions

1. Analysis of the wide usage of metaheuristic optimization algorithms in feature selection for data processing combined with machine learning methods, with a special focus on predicting Parkinson’s.

2. A comparative analysis for evaluating different feature selection methods (filter and wrapper) on predicting Parkinson's evaluating the subsets using three classification machine learning algorithms, and considering optimizing their parameters by a generalized Simulated Annealing heuristic algorithm.
3. Proposed novel and effective Binary Volleyball Premier League algorithm in feature selection which predicts with a higher accuracy Parkinson's, and a faster convergence speed compared to most other metaheuristic optimization algorithms.
4. Proposed integration of an "opposition-based learning" technique in the Binary Volleyball Premier League algorithm that improves its exploration abilities, and effectivity in predicting Parkinson's with a higher accuracy.
5. A new proposed hybrid metaheuristic of Binary Volleyball Premier League algorithm and Antlion Optimizer algorithm which aims to search for new optimal solution, and to improve the exploitation of Binary Volleyball Premier League algorithm considering BALO advantages. The hybrid metaheuristic improves the predictability of Parkinson's, and contributes in a more effective Binary Volleyball Premier League metaheuristic algorithm.
6. Proposed procedure to decrease the execution time of the proposed hybrid metaheuristic Binary Volleyball Premier League algorithm and Antlion Optimizer named "occurrence list" that improves its efficiency by avoiding redundant calculations of the fitness function.
7. Proposed efficient method to reduce the dimensionality of the data and to select the most relevant features by incorporating two algorithms: a feature ranking one based on cosine similarity, and the hybrid metaheuristic Binary Volleyball Premier League algorithm and Antlion optimizer algorithm in a most efficient time.

List of publications related to the thesis

1. Naka, E. K., Guliashki V. G. (2021), "Optimization Techniques in Data Management: A Survey", 7th International Conference on Computing and Data Engineering (ICCDE2021) (ACM Digital Library), January 15-17, 2021, Phuket, Thailand, pp. 8-13, ISBN:9781450388450; doi: 10.1145/3456172.3456214.
2. Naka, K. E., Guliashki V. G., Marinova G. I., (2021) "A Comparative Analysis of Different Feature Selection Methods on Parkinson Data", 2021 15th International Conference on Advanced Technologies, Systems and Services in Telecommunications

(TELSIKS 2021), (IEEE, Scopus), October 20-22, 2021, Niš, Serbia, pp. 366-371, doi:10.1109/TELSIKS52058.2021.9606398.

3. Naka E., Guliashki V., (2022), "B-VPL: "A Binary Volleyball Premier League optimization algorithm for Feature Selection", 2022 29-th International Conference on Systems, Signals and Image Processing (IWSSIP 2022), (IEEE, Scopus), June 01 - 03, 2022, Sofia, Bulgaria, pp. 1-4, doi: 10.1109/IWSSIP55020.2022.9854424.
4. Naka K. E., "Review of Metaheuristic Algorithms in Feature Selection based on Parkinson Disease", 2023 24th International Conference on Control Systems and Computer Science (CSCS), (IEEE, Scopus), Bucharest, Romania, 24-26 May 2023, pp. 221-228, doi: 10.1109/CSCS59211.2023.00042.
5. Naka E., "A Competitive Parkinson-Based Binary Volleyball Premier League Metaheuristic Algorithm for Feature Selection" Cybernetics Information Technologies, vol.23, no. 4 (Nov 2023), SJR 2022 (0.46) Q2, IF 2022 (1.2), Print ISSN: 1311-9702; Online ISSN: 1314-4081, doi:10.2478/cait-2023-0038
6. Naka E., "An efficient hybrid volleyball premier league and antlion metaheuristic algorithm for feature selection", 2023, 8th IEEE International Conference "Big Data, Knowledge and Control Systems Engineering" - BdKCSE'2023, (IEEE, Scopus), 02-03.11.2023, Sofia, Bulgaria, pp. 1-8, doi: 10.1109/BdKCSE59280.2023.10339732
7. Naka E., "A feature importance method based on cosine similarity and metaheuristic algorithm," in *IEEE International Conference on Artificial Intelligence in Engineering and Technology*, Kota Kinabalu, Malaysia, 2024 (accepted to be presented on 26-28 August 2024) Status: to appear, Indexed in: IEEE Xplore.