Hybrid Filter-Wrapper Feature Selection using Modified Flower Pollination Algorithm

Email: najme.mansouri@gmail.com*

Abstract- A major challenge in machine learning and data science is feature selection. Feature selection involves selecting the optimal (or suboptimal) subset of features to derive useful conclusions from a dataset based on the relevant information contained in those features. The Flower Pollination Algorithm (FPA) is a metaheuristic algorithm developed recently based on flower pollination. In this paper, we propose a new type of binary FPA, called the Filter-Wrapper Modified Binary FPA (FWMBFPA), which aims to improve convergence rate and solution quality by combining filter and wrapper advantages. Using FWMBFPA, the exploration process is directed toward specific search areas by extracting the features of existing solutions. 18 UCI datasets are used to evaluate the performance of the method. FWMBFPA generally performs better than the other algorithms in terms of average classification accuracy. FWMBFPA achieves the highest classification accuracy with the smallest number of selected features when compared to other algorithms when dealing with datasets with a large number of features.

Keywords: Feature selection, Flower Pollination Algorithm, Filter, Wrapper

1. Introduction

A broad range of fields can now access large amounts of data thanks to the advanced tools for collecting data. Data mining and machine learning tasks are greatly affected by data dimensionality [1]. Features increase in number as data dimensions increase. Thus, feature selection is used for selecting the optimal feature subset [2]. Data is reduced in dimensionality through a preprocessing step called Feature Selection (FS), which decreases learning times and eliminates irrelevant or redundant data points. The performance of supervised and unsupervised FS is degraded by redundant and irrelevant features, both of which add complexity [3], [4], [5]. A feature selection process can generally be divided into four steps: generation, evaluation, stopping conditions, and verification. It is crucial to evaluate the subset of features effectively [6]. The first step involves generating a subset of candidate features. In the second step, an evaluation indicator is used to assess the quality of the feature subset. When the process stops at step three, a feature subset meeting the stop criteria is output. FS does not directly involve the last step, but checks that the final feature subset is valid [7].

Researchers have been studying FS methods for decades. The methods are broadly categorized into four types: filter models, wrapper models, embedded models, and hybrid models [8], [9]. It does not require a learning algorithm to evaluate the filter model because it is based on the features' properties. The process is quicker and more efficient. However, since the filter model doesn't take into account the learning algorithm, some irrelevant features could be deleted, while some redundant ones would be retained. Filter approaches generally don't provide as high a classification accuracy as wrappers, so their feature subsets are generally less accurate. The wrapper model uses the classification algorithm for evaluating the results, which increases accuracy. Classification algorithms will need to be learned and verified. A large amount of data has a limited amount of running time, so the algorithm cannot evaluate each combination of features exhaustively. For this reason, heuristic optimization algorithms must be applied to help select feature subsets [10]. By using the original data directly for training, the embedded model constructs a classifier using only the optimal subset of features. These methods, however, take time and require knowledge of background parameters. To balance the algorithm's performance and time, the hybrid model combines the advantages of filter and wrapper models. It makes sense to develop a hybrid-based FS algorithm [11].

It is challenging to find a subset that is (nearly) optimal from the original set. In the past two decades, metaheuristics have proved to be highly reliable solutions to a wide range of optimization problems, including engineering design, machine learning, data mining, scheduling, and production problems [12]. Researchers have investigated metaheuristics in the field of feature selection [13]. It is NP-hard to solve FS with N features since there are 2N solutions to consider. When it comes to FS methods, there are three main search strategies:

- A complete (brute-force) search that generates all possible solutions before selecting the best
- Choosing subsets randomly and hoping to find the best subset
- Random search methods guided by heuristic information.

It is impractical to use complete and random methods with FS when dealing with medium- and large-scale datasets, and a random search becomes more complete when dealing with such datasets. By combining local and global search methods, heuristic search methods produce good (not always best) solutions within a reasonable timeframe [14]. It has become more common to use metaheuristic algorithms to mimic the evolution of living creatures. It is possible to find optimal global solutions using metaheuristic methods. They are more efficient than classical algorithms at solving complex, nonlinear, and indeterminate problems. When new data enters or the environment changes, metaheuristic algorithms don't have to restart. Simple, independent of the problem, flexible, and gradient-free characteristics are some of the advantages of meta-heuristics. Physical phenomena, animal behavior, and evolutionary concepts are common inspirations for meta-heuristics. Additionally, meta-heuristics are independent of the problem's nature, since they use a stochastic approach, which means they don't require derivative information. Unlike mathematical programming, which requires detailed knowledge about the mathematical problem, this program requires no prior knowledge. Because of their independence from the nature of the problem, they are a suitable tool for solving optimization problems without being concerned about the nonlinearity of the search space. Additionally, the algorithms' flexibility allows them to solve virtually any optimization problem without changing their structure significantly. This feature allows them to be a potential candidate for a user-friendly optimizer since they approach the problem as a black box with input and output states. Furthermore, they are mainly based on stochastic operators, unlike mathematical methods, which are deterministic. Therefore, conventional deterministic methods are less likely to lead to local optima. Their independence from the initial guess also enables them to be more flexible. Selecting features based on evolutionary algorithms can reduce the amount of time consumed and make classifications more accurate. Any problem that can be formulated can be solved using them when they are integrated with other optimization techniques. While these algorithms use mathematical formulas to solve problems, they are very fast and accurate [5], [15], [16].

Metaheuristic algorithms have two core concepts: exploration and exploitation. In exploration, the problem space is searched without concern for the results, while in exploitation, the focus is on the results. These capabilities need to be balanced to perform optimally in problem-solving [17], [18]. The exploration phase generally benefits from population-based algorithms. In addition, local search algorithms are typically used during the exploitation phase, since they can condense and find the most suitable solutions close to the original ones [19]. The purpose of this study is to combine the benefits of both a filter method and a wrapper method by using a modified flower pollination optimizer.

This study proposes a hybrid approach to address the crucial challenge of feature selection. This paper makes the following contributions:

- Proposing a novel FWMBFPA that effectively combines the advantages of filter and wrapper methods to improve feature selection performance.
- Developing a modified binary version of the flower pollination algorithm specifically tailored for feature selection problems, enhancing its search capabilities and convergence properties.

- Integrating a two-phase filtering mechanism based on Spearman correlation and relevance to efficiently reduce the dimensionality of the dataset and eliminate redundant and irrelevant features before the wrapper phase.
- Conducting extensive experiments on 18 diverse UCI datasets to demonstrate the superior performance of the proposed FWMBFPA in terms of average classification accuracy and achieving a significantly smaller number of selected features compared to several state-of-the-art metaheuristic algorithms.

The remainder of the article is organized as follows. An overview of flower pollination algorithms is provided in Section 2. Related works are highlighted in Section 3. Section 4 presents the proposed method. A comparison and evaluation of the proposed method is presented in Section 5, and the conclusion and future work are provided in Section 6.

2. Preliminaries

In this section, the concepts used are explained.

2.1. Flower Pollination in Optimization Context: Nature's Inspiration

The majority of plants around the world are flowering plants, where pollination is their primary means of reproduction. During pollination, pollen is transferred from one flower to another by wind, insects, butterflies, bees, birds, and bats. Several evolutionary processes have evolved to ensure pollination by producing nectar to attract pollinators. Additionally, some pollinators and plant species, such as hummingbirds and ornithophilous flowers, contribute to flower constancy in co-evolution. There are two basic types of pollination: biotic and abiotic [20], [21].

- *Biotic pollination:* Pollination is primarily accomplished by biotic pollinators, such as insects, birds, and others. Pollination of flowering plants by this method is used by almost 90% of them. Pollen can travel a long distance as pollinators move at different paces and speeds. It is also possible to consider pollination with such properties to be global pollination. This action can be equated to a global search if pollen is encoded as a solution vector.
- *Abiotic pollination:* In addition to pollination by pollinators, abiotic pollination is also called self-pollination. This form of pollination is estimated to be used by about 10% of floral plants. In local and self-pollinated plants, pollination is usually achieved by wind and diffusion. This type of motion is typically short in distance, making it suitable for use in local searches.
- *Flower constancy:* A partnership between plants and pollinators, such as hummingbirds, can be beneficial for both parties to save energy and achieve success. The result is flower constancy. A flower plant evolves so that pollinators are rewarded with nectar from a fixed set of flower types, while pollinators spend no energy exploring new flower types. To maximize pollinator reproduction by encouraging frequent visits by them [20], [21]. The flower pollination algorithm was developed using the main characteristics of pollination.

2.2. Flower Pollination Algorithm

Based on mimicking flower pollination, Yang [22] proposed the flower pollination algorithm as a metaheuristic optimization algorithm.

To ensure the quality of the search, FPA mixes exploitation and exploration randomly. As a result of FPA, the following idealized principles are followed [22]:

- Rule 1: Through Lévy flight, biotic cross-pollination acts as a global search.
- Rule 2: Local searches are abiotic and self-pollinated.
- Rule 3: The similarity between two flowers can lead to flower constancy.
- Rule 4: Local and global searches are switched randomly $\in [0, 1]$.

Initially, a random population is generated, and then the optimal solution is determined by evaluating the data. A new solution can be calculated by determining the pollination type according to a predetermined probability *p* (Rule 4). Assuming *r* is between 0 and 1, global pollination (Rule 1) and flower constancy (Rule 3) can occur as follows if *r* is less than *p* [22], [23], [24]:

$$x_i^{t+1} = x_i^t + \gamma L(x_i^t - g_{best}) \quad (1)$$

Eq. (1) involves x_i^t as a solution, *i* at time *t*, g_{best} as the current best solution, γ as a scaling factor, and *L* as a step size [22], [23], [24]:

$$L(s,c) \sim \frac{\lambda\Gamma(\lambda)\sin(\pi\frac{\lambda}{2})}{\pi} . \frac{c}{s^{1+\lambda}}, \quad (s \gg s_0 > 0) \quad (2)$$

For large steps s > 0, the gamma function $\Gamma(\lambda)$ is valid. The tail amplitude of the distribution is controlled by c, which is 1 in the proposed FPA. According to Rule 2, local pollination (Rule 2) and flower constancy can be expressed as follows [22], [23], [24]:

$$x_i^{t+1} = x_i^t + \varepsilon(x_j^t - x_k^t) \quad (3)$$

The pollens x_j^t and x_k^t come from flowers of the same plant species. As a result, flowers remain constant in a limited area. ε comes from a uniform distribution in [0,1], x_j^t and x_k^t give a local random walk if they are from the same species. The pollination of flowers can take place locally as well as globally. A nearby flower patch or flowers in a neighboring neighborhood are more likely to be pollinated by local flower pollen than those far away. The switch probability (Rule 4) or proximity probability *p* is used to switch between intensive local pollination and common global pollination. To determine the most appropriate parameter range, it is possible to use *p* = 0.5 as an initial value and then do a parametric study to determine the most appropriate parameter range. Algorithm 1 shows the pseudo-code of the FPA [22].

3. Related Works

In classification, feature selection is crucial. Recently, several algorithms have been developed for solving feature selection problems. General optimization problems benefit from swarm algorithms in terms of exploitation and exploration. It is still necessary to improve the accuracy of solution selection, the speed of time consumption, and the finding of global optimums in feature selection problems. To solve these drawbacks, there are many attempts in this direction.

The Salp Swarm Algorithm (SSA) is a bio-inspired algorithm designed to optimize a system using the swarming mechanisms of Salps [25]. Using Salp's swarm algorithm, Hegazy et al. [26] overcame the low convergence rate and avoided getting stuck in a local optimum. Twenty-seven datasets are used to evaluate the performance of CSSA when it is combined with the K-nearest neighbor classifier to solve the feature selection problem.

Using a wrapper approach, Naik et al. [27] identified the relevant subset of features for machine learning tasks. The Binary Bat algorithm is used to select a set of features, and a novel fitness function is implemented using One-pass Generalized Classifier Neural Networks (OGCNN). This fitness function takes into account the entropy of sensitivity, specificity, classifier accuracy, and fraction of selected features. Using four classifiers (Radial Basis Function Neural Networks, Probabilistic Neural Networks, Extreme Learning Machines, and OGCNNs), fitness functions are also compared on six publicly available datasets. Using one-pass classifiers is more efficient from a computational standpoint. Results indicated that OGCNN performed well in most cases when combined with the novel fitness function.

Algorithm 1. Flower Pollination Algorithm Pseudo-Code

Objective min or max **f(x)**, x = (x_1, …, x_d)
Initialize a population of n flowers/pollen with random solutions
Find the best solution g_{best} in the initial population
Define a switch probability p ∈ [0, 1]
While (stopping criterion not satisfied) **do**
    For i = 1: n (all *n* flowers in the population)
        If rand () < p
            Draw a step vector *L* that obeys a Levy distribution
            Global pollination: $x_i^{t+1} = x_i^t + \gamma L(x_i^t - g_{best})$
        else
            Select two random solutions x_j^t and x_k^t
            Local pollination: $x_i^{t+1} = x_i^t + \varepsilon(x_j^t - x_k^t)$
        end if
        Evaluate new solutions
        If new solutions are better, update them in the population
    end for
    Keep the current best solution
end while

Gao et al. [28] presented two algorithms for optimizing binary balance and selecting the best feature subset for classification problems. Equilibrium optimizers (EOs) are optimization algorithms based on physics [15]. In order to estimate dynamic and equilibrium states, it is based on models of controlled volumetric mass balance. First, the BEO-S and BEO-V algorithms map continuous EOs to discrete types. To determine the position of the optimal solution, the position vector (BEO-T) is used. A comparison of the proposed algorithm with other advanced FS algorithms is conducted on 19 well-known UCI datasets. Experimental results proved that BEO-V2 outperforms other state-of-the-art metaheuristic algorithms in terms of performance measures among the proposed binary EO algorithms.

The Grasshopper Optimization Algorithm (GOA) is an algorithm that mimics grasshopper migration and hunting in nature [29]. As a result of the low diversity of agents, this method tends to stagnate or become immature. Using SCGOA, Zhao et al. [30] proposed a new GOA with exploration and exploitation features to improve GOA's ability to handle a wide variety of situations. As a first step, trigonometric substitution is used to disturb people's position vector updates (evolution) to balance the exploration and exploitation stages in the proposed SCGOA. A Cauchy mutation-based strategy increases the diversity of the locust population and prevents stagnation. The Cauchy mutation ensures the diversity of locust populations. A comparison of SCGOA with several well-known meta-heuristic algorithms was conducted using the latest IEEE CEC2017 benchmark functions. The proposed SCGOA is superior to its rivals based on some extensive analysis results. The results of the study demonstrated that SCGOA was superior to some existing algorithms when applied to four engineering design problems based on Cauchy mutations. Several feature selection datasets were also handled using the binary version of Cauchy mutation-based SCGOA. Binary version of GOA outperforms original GOA and other optimization algorithms when it comes to classifying, having fewer errors, and fewer features.

In 2015, Duggan and Olmes [31] developed the Vortex Search Algorithm (VSA), a meta-heuristic algorithm based on the vortex phenomenon. Using chaos theory, Gharehchopogh et al. [32] overcome the entrapment of local optima, obtain the optimal feature set with maximum accuracy and minimum number of features. The proposed method considers various chaotic maps to improve the VSA operators and control both exploration and exploitation. Datasets from 24 UCI standards were used to evaluate this method's performance. This method was also evaluated as a Feature Selection (FS) approach. Based on simulation results, chaotic maps (especially the Tent map) can improve the performance of the VSA. In addition, it was demonstrated that the proposed method provided the best accuracy and the smallest number of features for determining the optimal feature subset. As compared to other algorithms, the proposed method performed better in the real application.

Using the firefly algorithm (FA) previously developed by Bacanin et al. [33], a new feature selection problem was addressed. Compared with the original FA, the proposed method performs much better in limited and practical terms. After validating the method on unconstrained benchmarks, 21 standard datasets from the University of California, Irvine (UCI) were used for feature selection. Furthermore, a new COVID-19 dataset was used in the present study to predict the health of patients, as well as a microcontroller microarray dataset. Based on the results of all practical simulations, we can certify the robustness and efficiency of the proposed algorithm when it comes to convergence, quality of solutions, and classification accuracy. In more detail, the proposed approach outperformed other competitor methods on 13 out of 21 datasets.

Although numerous metaheuristic algorithms have been used to select features, a critical review of existing literature reveals limitations and a research gap that this study aims to fill. The majority of previous studies used wrapper-based approaches (as summarized in Table 1), which, though often providing high accuracy, can be computationally expensive and suffer from scalability issues when dealing with high-dimensional datasets due to the lack of a feature reduction step at the outset. In these methods, feature subsets are evaluated using a learning algorithm, which becomes time-consuming as the number of features increases. Additionally, purely filter-based methods are computationally efficient, but evaluate features independently or based on intrinsic properties, potentially overlooking feature interactions and their influence on specific learning algorithms. Despite the fact that hybrid methods combine filter and wrapper approaches, there is still a need for more efficient and robust hybrid algorithms that can effectively balance the computational speed of filters with the accuracy of wrappers, especially in complex, high-dimensional datasets.

It is therefore necessary to develop a hybrid filter-wrapper feature selection algorithm that not only integrates the strengths of both paradigms but also enhances the optimization engine to ensure efficient exploration and exploitation of the search space. By introducing an initial filtering phase that handles high dimensionality and by integrating modifications to the Flower Pollination Algorithm's search mechanism within the wrapper phase, the FWMBFPA is proposed as a means of bridging this gap.

Table1. Related works on feature selection.

Ref.	Year	FS method	Advantages	Dataset used	Filter/Wrapper
Hegazy et al. [26]	2019	CSSA	Fewer parameters, simpler to implement	27	Wrapper
Naik et al. [27]	2020	OGCNN	Implementation using four classifiers, high accuracy	6	Wrapper
Gao et al. [28]	2020	BEO-S BEO-V	Extensive exploration and exploitation ability to change the solution at random	19	Wrapper
Zhao et al. [30]	2022	SCGOA	Enhance GOA's capability to handle diverse situations, avoid stagnation and laziness	-	Wrapper
Gharehchopogh et al. [32]	2022	VSA	Involving chaotic maps in VSA prevents local optima	24	Wrapper
Bacanin et al. [33]	2023	FA	High convergence speed	22	Wrapper

4. Proposed Method

The Flower Pollination Algorithm (FPA), introduced by Yang [22], is a nature-inspired metaheuristic algorithm that has demonstrated promising performance in solving various optimization problems. FPA is based on the fascinating process of flower pollination, incorporating both global pollination (biotic and cross-pollination via Lévy flight) and local pollination (abiotic and self-pollination). As a result of this inherent duality, FPA is able to balance exploration (searching diverse areas of the search space) and exploitation (refining potential solutions).

In feature selection, the goal is to find a subset of features that maximizes classification accuracy while minimizing the number of selected features. This NP-hard problem is well suited to metaheuristic algorithms. The ability of FPA to balance global and local search makes it an ideal candidate for navigating the complex and high-dimensional binary search space of feature selection. With its structure of updating based on the best solution and random interactions on the local level, it provides a solid foundation for adapting and modifying. Due to FPA's demonstrated effectiveness in optimization and its intrinsic mechanisms for exploration and exploitation, which are crucial for effectively searching the feature subset space, FPA was chosen for the wrapper phase. As a result, its structure allows for targeted modifications, such as the Modified FPA (MFPA) within our proposed FWMBFPA, that further enhance its performance. The framework of the proposed method is presented in this section, which is divided into two phases.

4.1. Filter phase

In order to handle high-dimensional data efficiently, the initial feature set needs to be reduced during this filter phase. This process begins with identifying and handling redundant features.

4.1.1. Redundancy computation

In the initial step of the filter phase, we address feature redundancy in order to reduce dimensionality. In order to accomplish this, the correlation between features is computed and analyzed.

In redundancy, two or more attributes are dependent on each other. MI measures how much a feature depends on a Subset (S) of features. Features that set symmetry, non-linearity, non-negativeness, and non-decreasing properties are observable as features are added. This measure does not indicate which features of S are redundant. A Markov blanket and total correlation are both useful over time measures that reduce redundancy. To assess numerical characteristics and subject matter knowledge, data-driven correlation analysis is useful. Data-driven methodologies can be used to calculate correlation coefficients between two features quickly. A highly correlated trait must have a correlation coefficient that exceeds a certain threshold to be eligible to calculate the Spearman correlation coefficient. Spearman correlation coefficients were used to estimate the correlation between two features. Correlation analysis uses Eq. (10) as an alternative stimulus condition. In addition to linear correlations, SCC can also measure nonlinear correlations. The SCC measures the degree to which two features are closely related. SCC values increase with stronger correlations. Feature f_i and feature f_j should not have an SCC greater than k_1 when there is a high correlation between them [34], [35], [36].

$$corr(f_i, f_j) = |SCC(f_i, f_j)|, n \geq k_1 \tag{4}$$

4.1.2. Relevance computation

Generally, an attribute is relevant if it provides information on a class tag attribute alone (C) or if it provides information when combined with another variable.

Weakly associated features, highly associated features, and unrelated features have been used to define associations. When a feature is strongly related to C, it cannot be replaced with another feature without removing its information. A weakly associated feature provides information about C, but can be replaced by another without losing any information. It is possible to lose information about C when you remove irrelevant features from it. In Table 2, the relevance levels of feature f_i are shown [34], [35], [36].

Table 2. Levels of relevance for feature f_i.

Relevance level	Condition	Probabilistic approach	Mutual information approach
Strongly relevant	$\nexists$	$p(C \mid f_i, \neg f_i) \neq p(C \mid \neg f_i)$	$I(f_i; C \mid f_i) \succ 0$
Weakly relevant	$\exists S \subset \neg f_i$	$p(C \mid f_i, \neg f_i) \neq p(C \mid \neg f_i) \ \wedge$ $p(C \mid f_i, S) \neq p(C \mid S)$	$I(f_i; C \mid f_i) \succ 0$ $\wedge$ $I(f_i; C \mid S) \succ 0$
Irrelevant	$\exists S \subset \neg f_i$	$p(C \mid f_i, S) \neq p(C \mid S)$	$I(f_i; C \mid S) \succ 0$

4.2. Wrapper phase

In the wrapper phase, the optimal feature subset is selected using a modified optimization algorithm, described in this subsection. In the wrapper phase, an evolutionary search approach is used to select the optimal subset based on the reduced and more relevant feature set obtained during the filter phase. The search is powered by a modified version of the flower pollination algorithm.

4.2.1. Modified Flower Pollination Algorithm

The selection of the optimal feature subset is based on an effective optimization algorithm, as described above in the wrapper phase description. For this essential search, the study uses an MFPA, described in this subsection.

Based on the clonal selection principle, the proposed MFPA modifies the standard FPA. According to experimental results, random walks produce faster convergent solutions than Levy flights in local pollination. Therefore, we replaced Levy flights with random walks. Using random uniform distributions in [0, 1], random walks are generated. A high-affinity solution is cloned proportional to its affinity before local pollination can be applied. To modify the local pollination, γ_2 was introduced as a step-size scaling factor. In a preliminary parametric study, it was found that $\gamma_2 = 3$ is effective for all test cases. At iteration *t*, the MFPA selects the top 14 solutions from a population *Pop* and clones each solution proportionally based on its fitness. Cloned solutions are likely to be exploited, so to avoid getting stuck in local minima, the algorithm checks a value not greater than 10e^6 for 100 successive iterations. In such a case, the entire population Pop is replaced by a new randomly generated one, while keeping the best solution g_{best}; this greatly increases exploration. The pseudo-code of the modified flower pollination algorithm can be seen in Algorithm 2.

4.2.2 Fitness function

In general, FS aims to minimize feature selection while maximizing classification accuracy. There is a conflict between these two objectives. It is possible to combine these two objectives into one objective problem by utilizing Eq. (5).

$$Fitness = \omega(1 - Accuracy) + (1 - \omega) \times F \tag{5}$$

A ratio *F* is computed by dividing the number of features selected by the original dimension of the dataset. The classification error rate of the selected subset of features is *(1-Accuracy).* Weight (ω) is represented by the values 0 and 1.

4.2.3. Filter-Wrapper Modified Binary Flower Pollination Algorithm (FWMBFPA)

In Eq. (6), the binary version of the algorithm is converted using sigmoid functions. As a consequence, FS can only be solved with binary values between 0 and 1. There is a binary vector for every solution, where 1 indicates that the corresponding feature has been selected, and 0 indicates that it has not been selected.

$$T(x) = \frac{1}{1 + e^{-x}} \tag{6}$$

Algorithm 2. Modified Flower Pollination Algorithm Pseudo-Code

Objective function $f(x)$, where $x = (x_1, \ldots, x_D)$ is a binary vector of dimension D.
Initialize a population of n flowers/pollen with random solutions
Find the best solution g_{best} in the initial population
Define a switch probability $p \in [0, 1]$
While (stopping criterion not satisfied) do
    If rand () < p
        For i = 1: n (all *n* flowers in the population)
        Draw a step length vector *L* from the Lévy distribution.
        Global pollination: $x_i^{t+1} = x_i^t + \gamma L(x_i^t - g_{best})$
        End for
    Else
        Identify the best m solutions from the current population P
        Solutions are cloned proportional to their fitness
        For each solution in clone population
            Obtain a random value $\boldsymbol{\varepsilon}$ uniformly distributed between 0 and 1
            Randomly select two distinct solutions j and k from the population P
            Local pollination: $x_i^{t+1} = x_i^t + \varepsilon(x_j^t - x_k^t)$
        End for
    End if
    Select the best n solutions from the combined pool to form the new population
    Update the current population P with the new population P_{new}
    Identify the best solution g_{best} in the current population P
    If g_{best} doesn't improve in 100 iterations by more than 10^-6, keep g_{best} and Replace the population with a new, randomly generated binary solutions
End while
Print g_{best}

The flowchart in Fig. 1 illustrates the proposed method's overall flow. Figure 1 shows two phases of the proposed method: filter and wrapper. Filtering is achieved using a combination of two filter methods, so that Spearman correlation between features is calculated first, a correlation limit of 0.8 is applied, and overly correlated features are discarded. By measuring the correlation between the category feature and other features, irrelevant features that were unrelated to the category feature were eliminated. Wrappers are provided with a bunch of normal, non-redundant features after the filter phase. The wrapping phase involved selecting an optimal set of features with maximum accuracy, based on the features of the modified flower pollination algorithm, which avoids the local optimum.

5. Experimental Results and Discussion

The purpose of this section is to present the experimental setup and datasets that were used to evaluate the proposed FWMBFPA. Several experiments were conducted to evaluate and compare the performance of the proposed FWMBFPA with existing methods. These experiments are described in more detail below.

5.1. Experiment setup

The purpose of this subsection is to provide specific details of the experimental setup used to conduct the evaluation. It includes the datasets, the data splitting strategy, as well as the configuration of the classifier. With the FS method, a subset of the entire dataset was selected to evaluate KNN classifier performance. According to [37], K=5 is the recommended value for KNN classifiers.

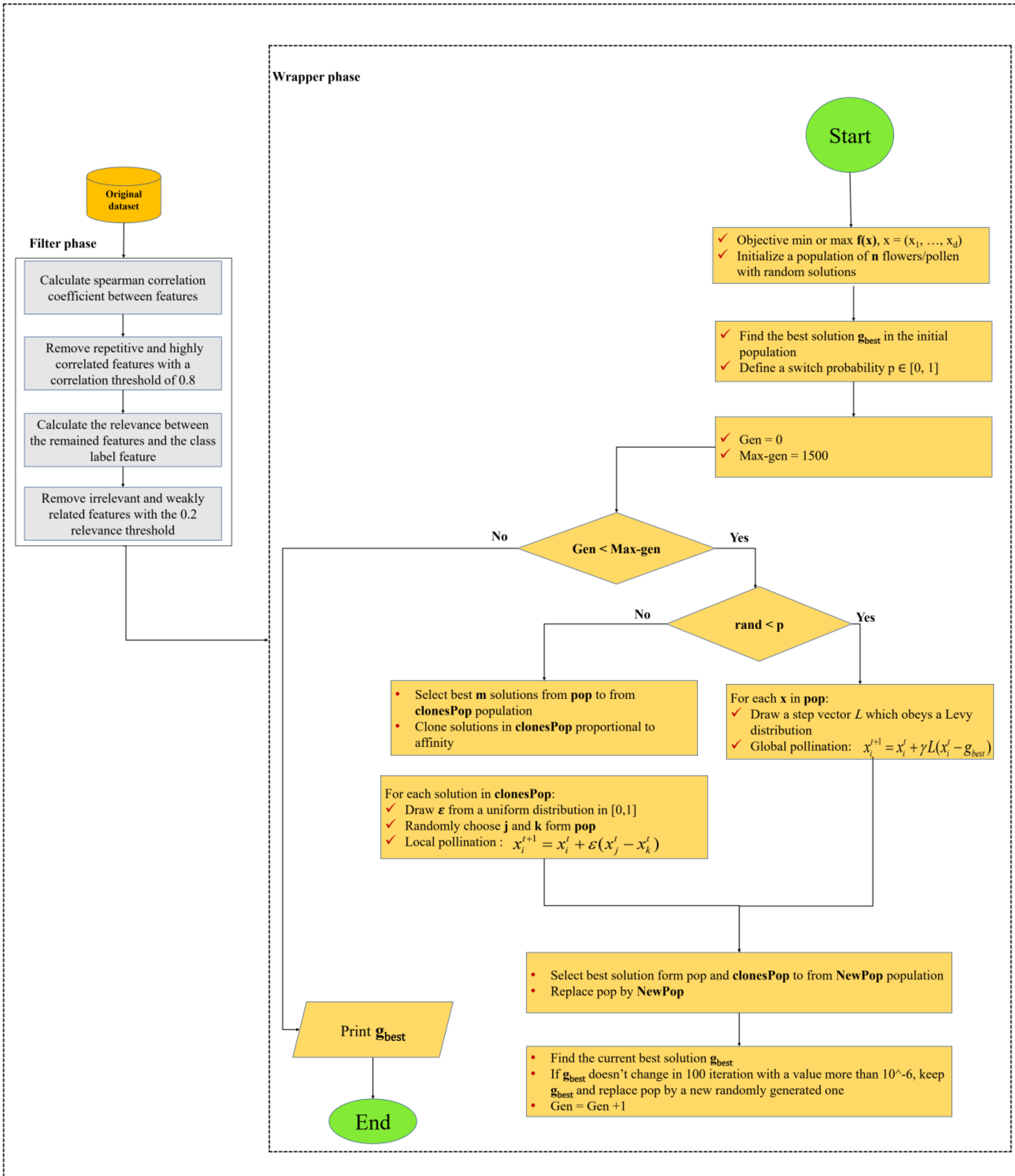


Fig. 1. Flowchart of proposed FWMBFPA.

A training dataset contains 80% of cases, while a test dataset contains 20%. The proposed method is implemented using Python3 and Matplotlib. There was 18 original UCI datasets evaluated in Table 3. Feature counts can be seen before filtering each dataset. Filtering is part of the detection phase, which detects redundancy among features, thus excluding duplicates, and ignoring features that are unrelated to category features. Additionally, the proposed method costs less to compute and can select a more accurate subset of features than existing methods. After applying the filter phase, Table 3 shows the dataset dimensions. The effect of the filter becomes more apparent on datasets with higher dimensions and more features, as shown in this table.

5.2. Evaluation of FWMBFPA and FPA

According to Table 4, BFPA and FWMBFPA were compared for classification accuracy and number of selected features across 18 datasets. As shown in Table 4, FPA had classification accuracy greater than 95% in 10 out of 18 data sets (55.55%), whereas FWMBFPA had classification accuracy greater than 95% in 12 out of 18 data sets (66.66%). Further, FWMBFPA selects fewer features in 17 data sets than BFPA and achieves the same number of features in only one data set (Zoo). To improve the performance of BFPA, the modified method that uses both the filtering and wrapping advantages has been modified and combined with the wrapper advantages. It is shown in Fig. 2 that FWMBFPA is both more accurate and has a lower mean number of selected features than BFPA.

Table 3. Description of datasets before and after the filter phase.

Dataset	No. of sample	No. of features before filter	No. of features after filter	No. of class	Domain
Algerian_forest_fires	244	14	8	2	Life
BreastCancer	698	11	9	2	Life
BreastEW	568	31	15	2	Life
CongressEW	434	17	15	2	Social
HeartEW	270	14	10	2	Life
Ionosphere	351	35	13	2	Physical
lung-cancer	32	57	28	2	Life
Lymphography	148	19	12	4	Life
M-of-n	1000	14	7	2	Life
Pd-speach	756	755	82	2	Life
penglung	73	326	148	7	Life
sobar-72	72	20	16	2	Physical
Sonar	208	61	35	2	Life
SpectEW	267	23	14	2	Physical
Vote	300	17	15	2	Social
Wholesale customers data	440	8	4	2	Business
Wine	178	14	12	3	Physical
Zoo	101	17	10	2	Life

Table 4. Classification accuracy of BFPA and FWMBFPA with selected features.

	Dataset	BFPA		FWBMFPA	
		Accuracy	No. of Features	Accuracy	No. of features
1	Algerian forest fires	98.64	3	100	2
2	Breast cancer	100	3	100	2
3	BreastEW	96.49	14	97.07	2
4	CongressEW	96.94	7	99.23	4
5	HeartEW	83.95	5	91.35	3
6	Ionosphere	91.50	15	97.16	3
7	lung-cancer	100	23	100	6
8	Lymphography	91.11	9	93.33	5
9	M-of-n	93	10	100	6
10	Pd-speech-feature	77.53	353	84.14	14
11	penglung	95.45	149	100	20
12	Sobar72	100	4	100	3
13	Sonar	90.47	30	92.06	9
14	SpectEW	88.88	10	92.59	6
15	Vote	97.77	8	98.88	2
16	Wholesale customers data	94.69	4	94.69	1

17	Wine	98.14	6	100	2
18	Zoo	96.77	5	96.77	5

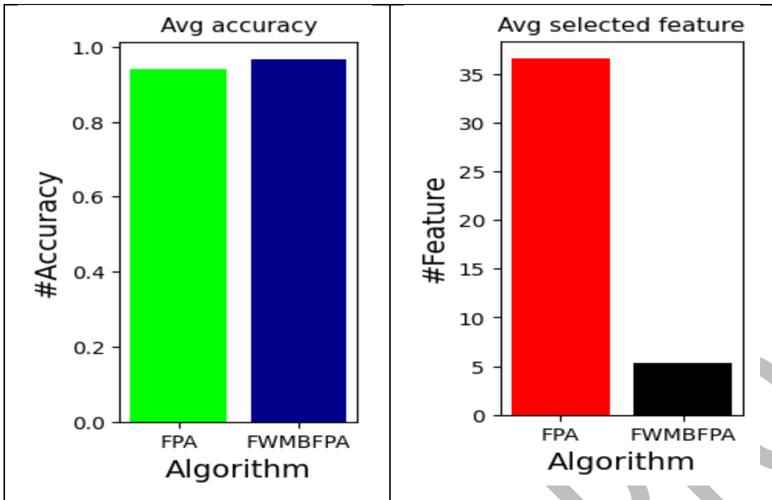


Fig. 2. Average accuracy and selected features obtained by FWMBFPA and BFPA.

5.3. Comparison and discussion

In this section, the results of comparing the proposed method with BFPA and 9 Binary optimization methods including Whale Optimization Algorithm (WOA), Time-Varying Salp Swarm Algorithm (TVSSA), Two-phase Mutation Gray Wolf Optimizer (TMGWO), Sine Cosine Algorithm (SCA), Jaya Algorithm (JA), Differential Evolution Algorithm (DEA), Cuckoo Search Algorithm (CSA), Bat Optimization Algorithm (BAT) and Bare Bone Particle Swarm Optimization (BBPSO) are presented.

5.3.1. Convergence rate of fitness value

Tables 5, 6, and 7 show the worst, average, and best fitness values obtained from FWMBFPA and other methods. Based on Table 5, FWMBFPA has the lowest fitness value out of 10 datasets out of 18 (55.55%), and has the worst fitness value in 8 of the 18 datasets. Bold number in all Tables shows the best performance. After the proposed method, BWOA and TVSSA have a better fitness value in the two datasets. In the entire dataset, BFPA, TMGWO, and BJA have the worst performance and worst fitness values. Table 6 shows that the proposed FWMBFPA has a significant advantage over other methods in terms of average fitness value (83.33%) in 15 datasets. Thereafter, only BCSO, BBPSO, and TMGWO had optimal fitness values in equal than one dataset, whereas BFPA did not have an optimal fitness value in any dataset. The best fitness value in 12 data sets was obtained by FWMBFPA in Table 7 (66.66%) when compared to other methods. The best fitness values in three datasets are obtained by BBAT and BBPSO, followed by TMGWO, BCSA, BDE, and BSCA, but BJA only in one dataset achieves the best value, while other methods like BFPA are not able to achieve the best value. Compared to the results derived from these three tables, FWMBFPA significantly improves BFPA's performance and is superior to other methods, and has a higher convergence rate. Figure 3 presents the fitness values obtained using FWMBFPA and other methods for a clearer understanding of the content. To improve clarity and identification, a circular legend highlights the proposed method, shown in blue.

5.3.2. Evaluation of Classification accuracy and selected features

The accuracy of classification and the number of selected features of FWMBFPA, BFPA, and nine other methods are compared in this section. If the classification accuracy of a method is higher compared to other methods, it means that the method in question performed with better accuracy and had fewer errors. Among the 18 datasets analyzed in Table 8, the FWMBFPA method had the best classification accuracy in 14 datasets (77.77%) and performed better than other methods, followed by BSCA and TMGWO in 8 datasets (44.44%). Based on this Table, with better performance in two datasets, BFPA and BWOA perform the worst in terms of classification accuracy.

Table 5. The worst fitness value achieved by FWMBFPA and other methods.

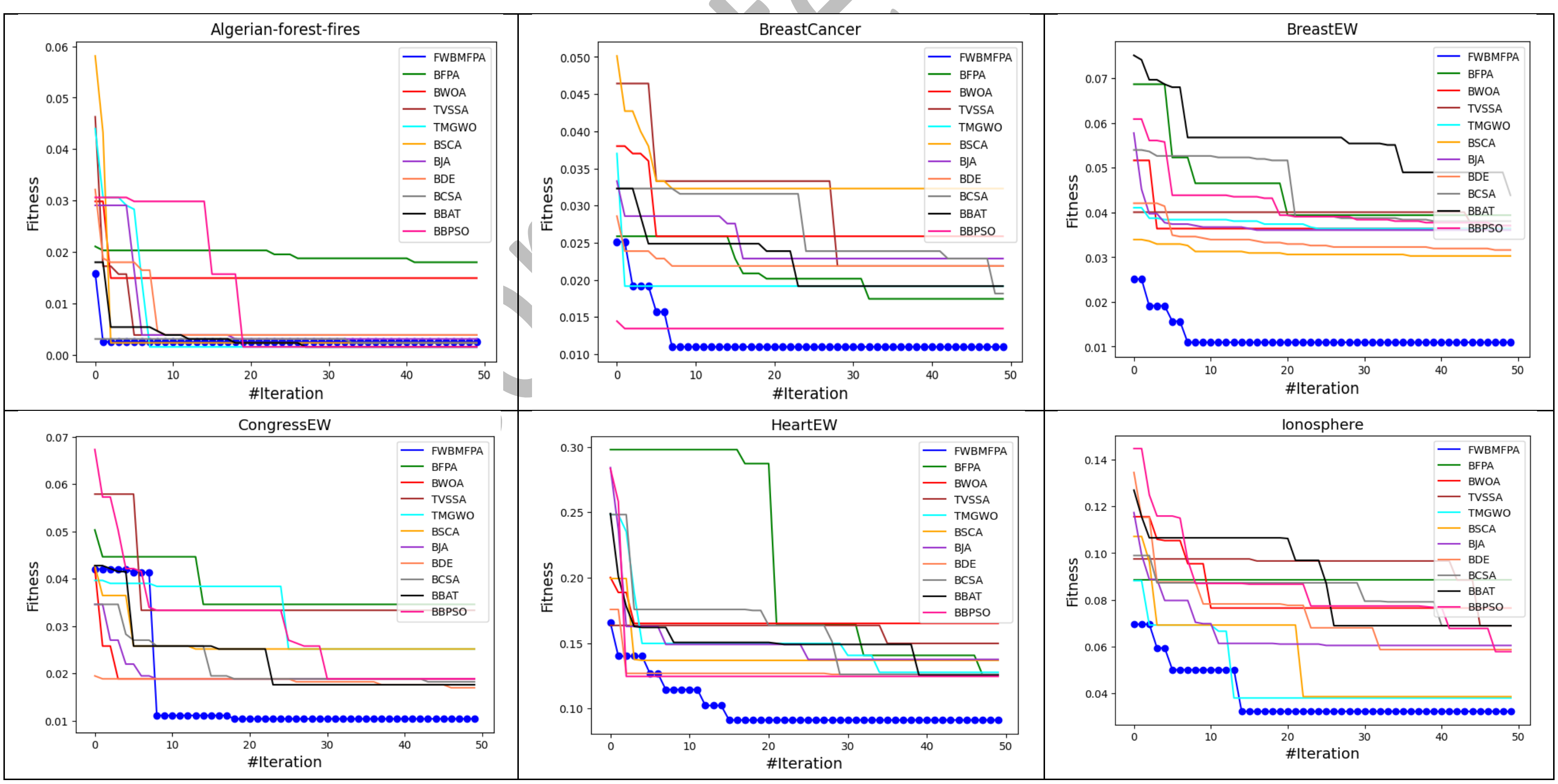
Dataset	FWBMFPA	BFPA	BWOA	TVSSA	TMGWO	BSCA	BJA	BDE	BCSA	BBAT	BBPSO
Algerian forest fires	**0.0158**	0.0210	0.0298	0.0462	0.0439	0.0581	0.0290	0.0321	0.0030	0.0179	0.0306
Breast cancer	0.0251	0.0258	0.0380	0.0464	0.0370	0.0501	0.0332	0.0285	0.0322	0.0322	**0.0144**
BreastEW	0.0520	0.0686	0.0516	0.0400	0.0410	**0.0339**	0.0577	0.0420	0.0539	0.0751	0.0608
CongressEW	0.0420	0.0503	0.0421	0.0579	0.0396	0.0427	0.0346	**0.0194**	0.0346	0.0427	0.0673
HeartEW	0.1655	0.2979	0.2001	**0.1635**	0.2482	0.1994	0.2841	0.1757	0.2482	0.2490	0.2834
Ionosphere	**0.0695**	0.0884	0.1156	0.0975	0.0881	0.1071	0.1173	0.1345	0.0989	0.1270	0.1448
lung-cancer	**0.1026**	0.2026	0.1052	0.2026	0.2040	0.2031	0.3018	0.2028	0.3016	0.2024	0.3009
Lymphography	**0.0723**	0.1601	0.0907	0.2024	0.1573	0.2018	0.1364	0.1358	0.1798	0.1347	0.1810
M-of-n	**0.0100**	0.1414	0.1795	0.1457	0.1894	0.1851	0.1612	0.1899	0.1670	0.1792	0.0960
Pd-speech-feature	**0.1701**	0.2272	0.3101	0.2664	0.2886	0.2621	0.2229	0.2403	0.2321	0.2228	0.2362
Penglung	0.0948	0.0948	**0.0497**	0.1849	0.0944	0.0050	0.1401	0.0946	**0.0497**	0.0499	0.0502
Sobar72	**0.0046**	0.0486	0.0073	0.0513	0.0936	0.0492	0.0497	0.0497	0.0497	0.0497	0.0047
Sonar	0.1484	0.2106	**0.1146**	0.1305	0.1300	0.1465	0.2082	0.1935	0.1312	0.1459	0.1628
SpectEW	**0.1146**	0.1874	0.1512	0.1154	0.1629	0.1507	0.1666	0.1403	0.1385	0.1620	0.1629
Vote	0.0401	0.0386	0.0606	**0.0276**	0.0697	0.0367	0.0496	0.0593	0.0483	0.0392	0.0600
Wholesale Costomers data	**0.0474**	0.0867	0.0642	0.0807	0.0717	0.0703	0.0596	0.1032	0.0792	0.0910	0.0657
Wine	**0.0054**	0.0596	0.0412	0.0802	0.1520	0.0412	0.0596	0.0412	0.0405	0.1512	0.1489
Zoo	0.0694	0.0719	0.1327	0.0701	0.0707	0.0375	0.1008	0.1327	0.0381	**0.0056**	0.0350

Table 6. The average fitness value achieved by FWMBFPA and other methods.

Dataset	FWBMFPA	BFPA	BWOA	TVSSA	TMGWO	BSCA	BJA	BDE	BCSA	BBAT	BBPSO
Algerian forest fires	**0.0027**	0.0193	0.0155	0.0057	0.0054	0.0042	0.0061	0.0063	0.0028	0.0031	0.0112
Breast cancer	**0.0122**	0.0209	0.0269	0.0295	0.0194	0.0333	0.0247	0.0221	0.0273	0.0221	0.0134
BreastEW	**0.0122**	0.0448	0.0373	0.0397	0.0343	0.0310	0.0370	0.0337	0.0444	0.0560	0.0417
CongressEW	**0.0155**	0.0375	0.0196	0.0362	0.0319	0.0265	0.0199	0.0183	0.0217	0.0229	0.0293
HeartEW	**0.1009**	0.2101	0.1666	0.1593	0.1483	0.1405	0.1490	0.1284	0.1567	0.1491	0.1303
Ionosphere	**0.0387**	0.0884	0.0822	0.0936	0.0467	0.0540	0.0658	0.0730	0.0827	0.0879	0.0849
lung-cancer	**0.0229**	0.1313	0.1038	0.1666	0.1091	0.1029	0.0829	0.0479	0.1130	0.0484	0.0708
Lymphography	**0.0708**	0.1362	0.0907	0.1679	0.1001	0.0990	0.0945	0.1124	0.1176	0.1247	0.1226
M-of-n	**0.0100**	0.0789	0.0782	0.1055	0.0420	0.0316	0.0146	0.0414	0.0361	0.0379	0.0102
Pd-speech-feature	0.1603	0.2271	0.2111	0.2118	**0.1538**	0.1817	0.2141	0.2301	0.2315	0.2267	0.2316
Penglung	**0.0143**	0.0750	0.0479	0.1173	0.0486	0.0014	0.0184	0.0448	0.0395	0.0493	0.0488
Sobar72	**0.0022**	0.0178	0.0027	0.0217	0.0240	0.0178	0.0078	0.0116	0.0297	0.0189	0.0035
Sonar	0.0935	0.1118	0.0854	0.1176	0.0793	0.0817	0.0789	0.1067	**0.0706**	0.1056	0.0896
SpectEW	**0.0899**	0.1400	0.1141	0.1098	0.0933	0.1067	0.1141	0.0931	0.1136	0.1110	0.1056
Vote	**0.0202**	0.0274	0.0368	0.0262	0.0333	0.0286	0.0244	0.0383	0.0323	0.0245	0.0277
Wholesale Costomers data	**0.0474**	0.0867	0.0499	0.0770	0.0525	0.0505	0.0527	0.0670	0.0553	0.0688	0.0555
Wine	**0.0048**	0.0538	0.0210	0.0633	0.0295	0.0215	0.0239	0.0228	0.0236	0.0358	0.0253
Zoo	0.0387	0.0519	0.0419	0.0405	0.0191	0.0070	0.0115	0.0330	0.0152	0.0055	**0.0144**

Table 7. The best fitness value achieved by FWMBFPA and other methods.

Dataset	FWBMFPA	BFPA	BWOA	TVSSA	TMGWO	BSCA	BJA	BDE	BCSA	BBAT	BBPSO
Algerian forest fires	0.0025	0.0179	0.0149	0.0038	0.0015	0.0023	0.0030	0.0038	0.0023	**0.0015**	**0.0015**
Breast cancer	**0.0109**	0.0174	0.0258	0.0218	0.0191	0.0322	0.0228	0.0218	0.0181	0.0191	0.0134
BreastEW	**0.0303**	0.0394	0.0364	0.0370	0.0364	**0.0303**	0.0360	0.0316	0.0380	0.0438	0.0370
CongressEW	**0.0104**	0.0346	0.0188	0.0333	0.0251	0.0251	0.0188	0.0169	0.0182	0.0176	0.0188
HeartEW	**0.0911**	0.1260	0.1650	0.1497	0.1276	0.1367	0.1375	0.1260	0.1260	0.1252	0.1245
Ionosphere	**0.0321**	0.0884	0.0764	0.0689	0.0379	0.0385	0.0604	0.0586	0.0689	0.0689	0.0578
lung-cancer	**0.0026**	0.1036	0.1034	0.1031	0.1002	0.0012	0.0033	0.0035	0.0035	0.0051	0.0033
Lymphography	**0.0705**	0.0930	0.0907	0.1144	0.0907	0.0918	0.0918	0.0913	0.0913	0.0913	0.1133
M-of-n	0.0100	0.0358	0.0325	0.0729	**0.0046**	**0.0046**	**0.0046**	**0.0046**	**0.0046**	**0.0046**	**0.0046**
Sobar72	**0.0020**	0.0057	0.0021	0.0042	0.0021	0.0036	0.0026	0.0042	0.0036	0.0047	0.0031
Sonar	0.0838	0.0967	0.0822	0.0984	0.0665	0.0519	0.0511	**0.0514**	0.0360	0.0989	0.0820
SpectEW	**0.0779**	0.1145	0.1027	0.1014	0.0891	0.1000	0.0928	0.0905	0.1023	0.1009	0.1000
Vote	0.0145	0.0270	0.0361	0.0251	0.0276	0.0238	0.0232	0.0251	0.0245	**0.0141**	0.0232
Wholesale Costomers data	**0.0474**	0.0867	0.0492	0.0746	0.0492	0.0492	0.0521	0.0642	0.0521	0.0671	0.0553
Wine	**0.0036**	0.0229	0.0206	0.0428	0.0046	0.0206	0.0229	0.0214	0.0214	0.0214	0.0198
Zoo	0.0374	0.0381	0.0388	0.0356	0.0037	0.0050	0.0037	0.0050	**0.0031**	0.0050	**0.0031**
Pd-speech-feature	0.1587	0.2271	0.2009	0.2099	**0.1199**	0.1664	0.2134	0.2182	0.2312	0.2225	0.2306
Penglung	**0.0006**	0.0499	0.0464	0.0488	0.0463	0.0007	0.0048	0.0038	0.0045	0.0490	0.0479



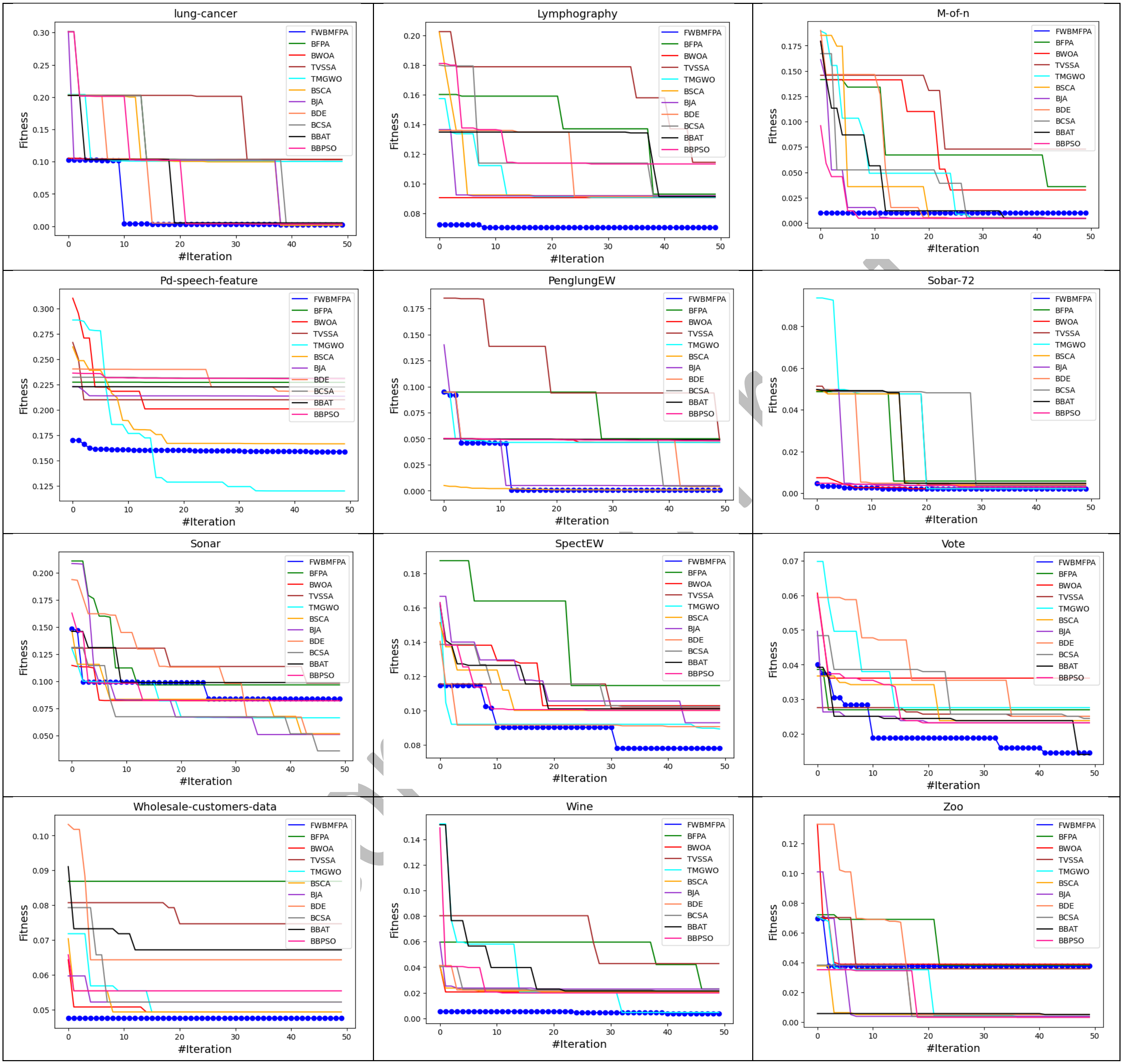


Fig. 3. FWMBFPA fitness values compared to other methods with KNN classifier.

Based on each method, Table 9 shows the number of features selected. FWMBFPA achieved the lowest number of selected features in all 18 datasets (100%), which is outstanding compared to other methods. Other methods have a percent superiority of less than 17% in the selected feature, and BFPA has selected the fewest features in only one data set, showing how much FWMBFPA has affected BFPA. Therefore, FWMBFPA has superior performance in both classification accuracy and number of selected features, while BFPA has improved its performance. As can be seen in Fig. 4, FWMBFPA selects features with an average accuracy, and other methods select features with an average feature selection. In the diagram, it is apparent that fewer features have been selected more accurately, improving BFPA's performance.

Table 8. Classification accuracy obtained by FWMBFPA and other methods.

Dataset	FWBMFPA	BFPA	BWOA	TVSSA	TMGWO	BSCA	BJA	BDE	BCSA	BBAT	BBPSO
Algerian forest fires	**1**	0.9864	0.9864	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**
Breast cancer	**0.9952**	0.9809	0.9714	0.9809	0.9904	0.9857	0.9857	0.9809	0.9857	0.9857	0.9904
BreastEW	**0.9707**	0.9649	0.9649	0.9649	**0.9707**	**0.9707**	0.9649	0.9649	0.9649	0.9649	0.9649
CongressEW	**0.9923**	0.9694	0.9847	0.9694	0.9770	0.9770	0.9847	0.9874	0.9847	0.9847	0.9847
HeartEW	**0.9135**	0.8395	0.8395	0.8518	0.8765	0.8641	0.8641	0.8765	0.8765	0.8765	0.8765
Ionosphere	**0.9716**	0.9150	0.9150	0.9339	0.9622	0.9622	0.9433	0.9433	0.9433	0.9339	0.9622
lung-cancer	**1**	**1**	0.9000	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**
Lymphography	**0.9333**	0.9111	0.9111	0.8888	0.9111	0.9111	0.9111	0.9111	0.9111	0.9111	0.8888
M-of-n	**1**	0.9300	0.9733	0.9333	**1**	**1**	**1**	**1**	**1**	**1**	**1**
Pd-speech-feature	0.8414	0.7753	0.7973	0.8193	**0.8810**	0.8325	0.7885	0.7841	0.7709	0.7797	0.7709
penglung	**1**	0.9545	0.9545	0.9545	0.9545	**1**	**1**	**1**	**1**	0.9545	0.9545
Sobar72	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**
Sonar	0.9206	0.9047	0.9206	0.9047	0.9206	0.9365	0.9523	0.9523	**0.9682**	0.9523	0.9206
SpectEW	**0.9259**	0.8888	0.9012	0.8888	0.9135	0.9012	0.9135	0.9135	0.9012	0.9012	0.9012
Vote	**0.9888**	0.9777	0.9666	0.9777	0.9777	0.9777	0.9777	0.9777	0.9777	**0.9888**	0.9777
Wholesale customers data	0.9469	0.9469	**0.9545**	0.9318	0.9545	**0.9545**	**0.9545**	0.9393	0.9545	0.9393	**0.9545**
Wine	**1**	0.9814	0.9814	0.9814	**1**	0.9814	0.9814	0.9814	0.9814	0.9814	0.9814
Zoo	0.9677	0.9677	0.9677	0.9677	**1**	**1**	**1**	**1**	**1**	**1**	**1**

Table 9. The number of selected features by FWMBFPA and other methods.

Dataset	FWBMFPA	BFPA	BWOA	TVSSA	TMGWO	BSCA	BJA	BDE	BCSA	BBAT	BBPSO
Algerian forest fires	**2**	3	3	3	2	3	**2**	2	3	2	**2**
Breast cancer	4	4	**3**	**3**	4	**3**	6	**3**	4	5	4
BreastEW	**2**	14	5	7	4	4	4	5	9	11	7
CongressEW	**4**	7	6	5	8	6	6	3	5	**4**	5
HeartEW	**3**	5	8	4	7	**3**	**3**	4	5	4	**3**
Ionosphere	**3**	15	4	12	5	4	15	9	16	12	8
lung-cancer	**6**	23	9	22	38	7	16	16	20	24	15
Lymphography	**5**	9	**5**	7	**5**	7	7	6	10	6	**5**
M-of-n	**6**	10	8	9	**6**	**6**	**6**	**6**	**6**	**6**	**6**
Pd-speech	**14**	353	22	74	165	57	307	357	339	337	265
penglung	**20**	149	48	116	25	25	159	125	147	131	96
Sobar72	**3**	4	4	5	4	4	5	4	4	4	4
Sonar	**9**	30	22	25	14	10	29	19	28	27	17
SpectEW	**4**	10	11	7	5	5	10	11	10	7	5
Vote	**2**	8	4	4	**2**	**2**	**2**	3	4	5	**2**
Wholesale customers data	**1**	4	3	4	3	3	3	3	3	2	5
Wine	**2**	6	3	3	6	3	4	4	4	4	3
Zoo	**5**	**5**	6	6	6	8	6	7	**5**	6	**5**

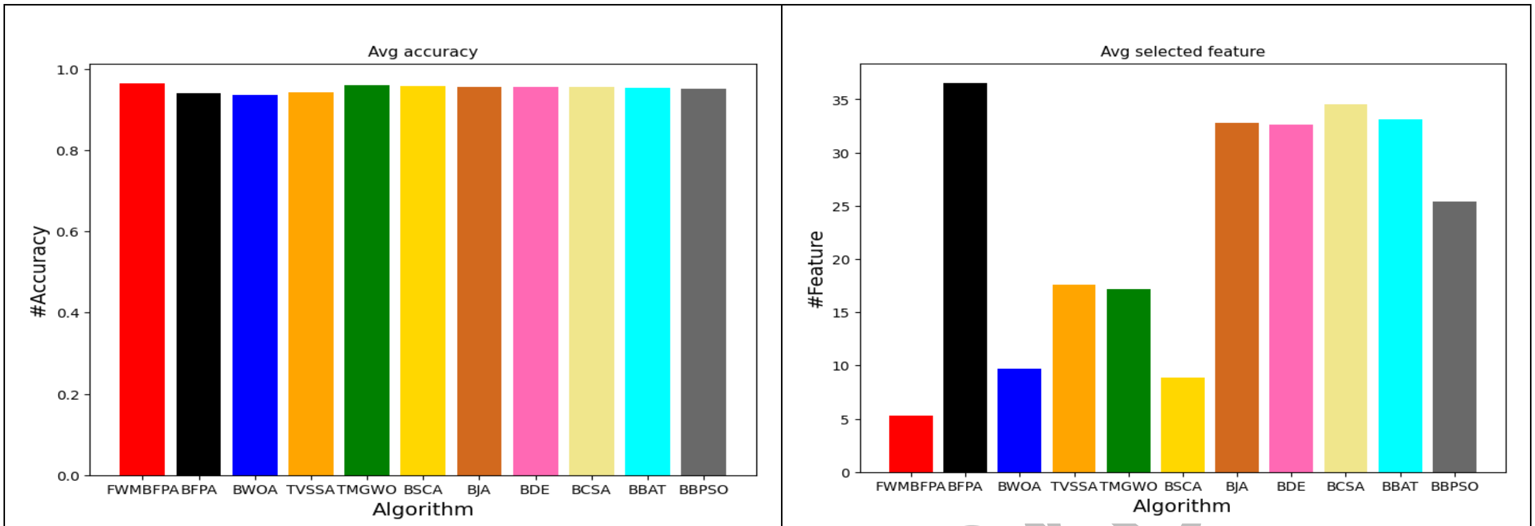


Fig. 4. Average accuracy and average selected features by FWMBFPA and the other methods.

5.3.3. Statistical Analysis

In order to rigorously assess the statistical significance of the observed performance differences between the compared feature selection algorithms across the 18 datasets, a non-parametric statistical analysis was carried out. A Friedman test was performed independently on the classification accuracy results (Table 8) and the number of selected features results (Table 9), which are suitable non-parametric alternatives to ANOVA for comparing multiple groups over multiple test conditions. The Friedman test yielded a statistic of 65.0219 with a corresponding p-value of 0. Since the p-value ($p<0.05$), which is less than the significance level $\alpha=0.05$, the null hypothesis is rejected. There is a statistically significant difference in classification accuracy among the algorithms evaluated. For the number of selected features, the Friedman test yielded a statistic of 61.8933 with a p-value of 0. This p-value is also less than $\alpha=0.05$, leading to the rejection of the null hypothesis and confirming a statistically significant difference in the number of features selected by the algorithms.

Since the Friedman test showed significant differences for both evaluation metrics, the Nemenyi post-hoc test was conducted to determine which specific pairs of algorithms performed statistically significantly differently. Nemenyi tests compare the average ranks of algorithms across datasets. In Table 10, the average ranks for classification accuracy as well as the number of selected features are sorted by the average rank for accuracy. The Nemenyi test determines statistically significant differences between algorithms when their average rank differences exceed the Critical Difference (CD). The calculated CD value for comparing 11 algorithms over 18 datasets at a significance level of $\alpha=0.05$ is approximately $CD\approx3.345$.

Table 10 shows that the proposed FWMBFPA achieved the lowest average rank for classification accuracy (3.3889) and the lowest average rank for the number of selected features (1.8889). For classification accuracy, FWMBFPA's average rank (3.3889) is lower than all other algorithms. Comparing FWMBFPA to other algorithms using the CD (3.345):

- FWMBFPA's average rank is significantly lower than algorithms whose average rank is greater than 3.3889+3.345=6.7339. Based on Table 10, FWMBFPA shows a statistically significantly higher accuracy compared to BWOA (8.0556), TVSSA (8.2500), and BFPA (8.6111), as their average ranks are greater than 6.7339.
- For the remaining algorithms (TMGWO, BSCA, BJA, BDE, BCSA, BBAT, BBPSO), the difference in average rank compared to FWMBFPA is less than or equal to the CD, indicating no

statistically significant difference in accuracy compared to FWMBFPA at the 0.05 significance level. However, FWMBFPA holds the best average rank among all.

For the number of selected features, FWMBFPA obtained an outstanding average rank of 1.8889. Comparing FWMBFPA's rank to others using the CD (3.345):

- FWMBFPA's average rank (1.8889) is significantly lower than all other algorithms, whose average rank is greater than 1.8889+3.345=5.2339. Based on Table 10, FWMBFPA selects a statistically significantly lower number of features compared to TMGWO (5.3889), BWOA (5.6111), BDE (5.6667), TVSSA (6.7222), BJA (6.8333), BCSA (7.3889), BBAT (6.5000), and BFPA (8.8889).
- The difference in average rank between FWMBFPA (1.8889) and BSCA (4.2222) is $|1.8889-4.2222|=2.3333$, which is less than the CD (3.345).
- The difference in average rank between FWMBFPA (1.8889) and BBPSO (4.4444) is $|1.8889-4.4444|=2.5555$, which is less than the CD (3.345). Therefore, there is no statistically significant difference in the number of selected features between FWMBFPA and BSCA or BBPSO, although FWMBFPA still has the lowest average rank.

Based on the statistical analysis, the proposed FWMBFPA achieved not only the best average rankings for classification accuracy and feature selection but also demonstrated statistically significant superiority in accuracy over several algorithms as well as a statistically significant ability to select fewer features than most of the algorithms evaluated. Statistical evidence supports the effectiveness of the proposed method based on these results.

Figures 5 and 6 provide a visual representation of these comparisons.

Table 10. Average Ranks of Algorithms on 18 Datasets (Lower Rank is Better).

Algorithm	Avg. Rank (Accuracy)	Rank (Accuracy)	Avg. Rank (Features)
FWMBFPA	3.3889	1	1.8889
TMGWO	4.5278	2	5.3889
BSCA	5.0833	3	4.2222
BJA	5.1944	4	6.8333
BCSA	5.3611	5	7.3889
BDE	5.4167	6	5.6667
BBAT	5.8333	7	6.5000
BBPSO	6.2778	8	4.4444
BWOA	8.0556	9	5.6111
TVSSA	8.2500	10	6.7222
BFPA	8.6111	11	8.8889
Critical Difference (CD) for Nemenyi Test (α=0.05α=0.05)	≈ 3.345		

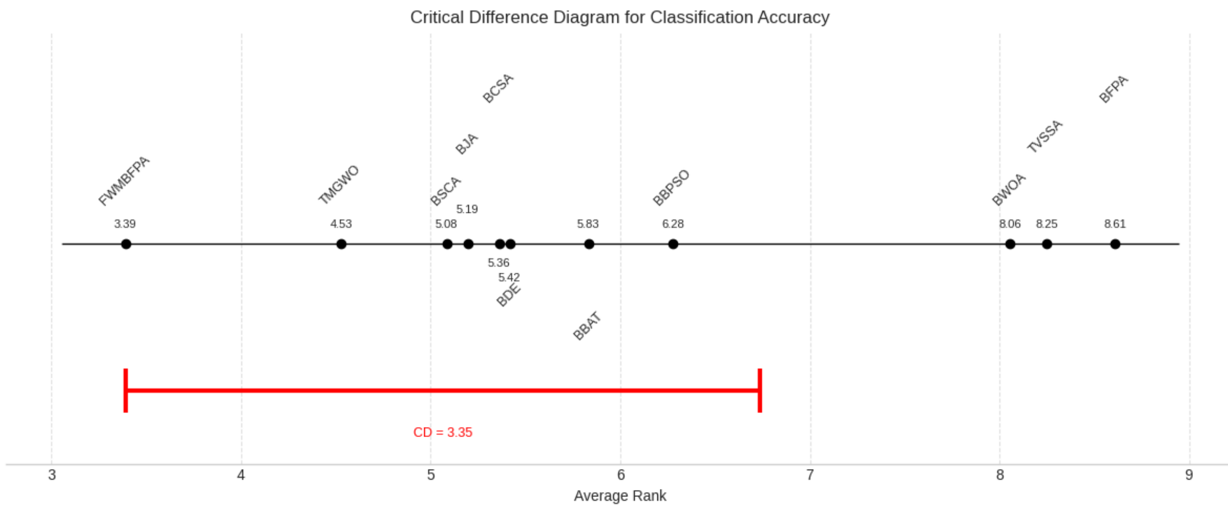


Fig. 5. CD diagram showing the average ranks for Classification Accuracy. The red bar indicates the Nemenyi Critical Difference (α=0.05).

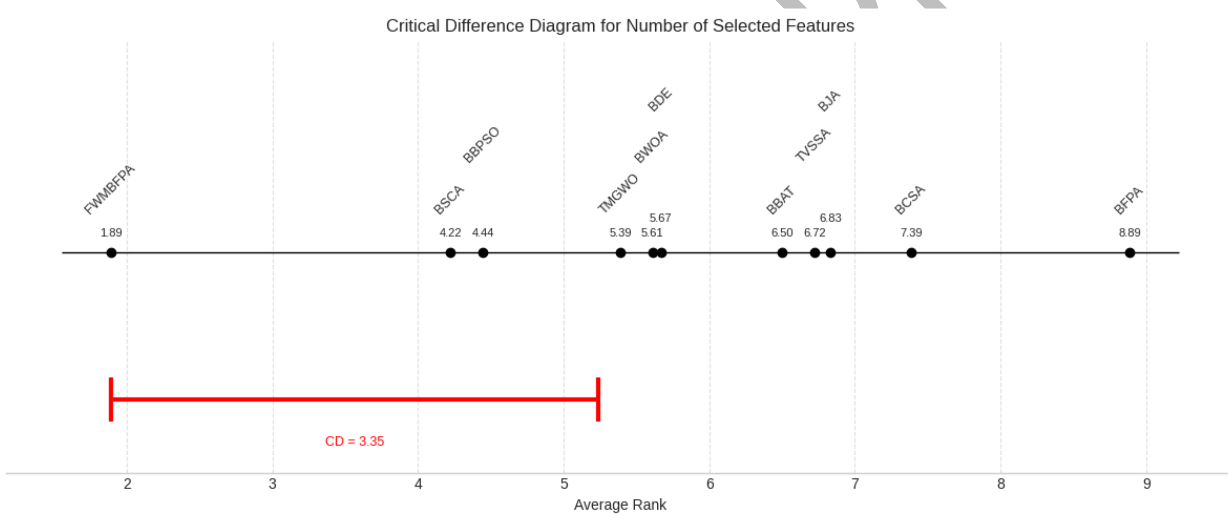


Fig. 6. CD diagram showing the average ranks for the Number of Selected Features. The red bar indicates the Nemenyi Critical Difference (α=0.05).

5.3.4. Time Complexity Analysis

This section analyzes the theoretical time complexity of the proposed FWMBFPA and compares it with the general complexity of wrapper-based metaheuristic algorithms. The time complexity of a typical population-based metaheuristic is primarily determined by the number of iterations (T), the population size (P), and the cost of evaluating the fitness function for each solution in every iteration. In feature selection using a classifier like KNN, the fitness evaluation for a subset of F features on N samples has a complexity of approximately O(N×F). Thus, a standard wrapper-based metaheuristic operating on the original D features has a theoretical complexity of O(T×P×N×D). The proposed FWMBFPA, being a hybrid approach, includes an initial filter phase. This phase involves calculating pairwise Spearman correlations among D features, which takes about $O(D^2 \times N)$, and assessing feature relevance, which is O(D×N). This filter phase is performed only once. The subsequent wrapper phase then operates on the reduced set of D′ features. The fitness evaluation in this phase has a complexity of O(N×D′). Therefore, the complexity of the wrapper phase is O(T×P×N×D′). The total theoretical complexity of FWMBFPA is the sum of the complexities of the two phases: $O(D^2 \times N + T \times P \times N \times D')$.

Table 11 summarizes this comparison. Theoretically, FWMBFPA introduces an initial cost ($O(D^2 \times N)$) that is absent in pure wrapper methods. However, for datasets with a large number of original features (D) where the filter phase effectively reduces the feature space to a much smaller number of features ($D' \ll D$), the cost per iteration in the wrapper phase $O(N \times D')$ becomes significantly lower than $O(N \times D)$. If the total number of fitness evaluations ($T \times P$) is sufficiently large, the cumulative savings in the wrapper phase can potentially outweigh the initial filter cost, making FWMBFPA theoretically more efficient for high-dimensional problems with high redundancy/irrelevance. This aligns with our experimental results showing significant feature reduction on such datasets.

This analysis provides an asymptotic upper bound. Several variables can influence the empirical runtime, such as implementation details, hardware specifications, and constant factors hidden in the Big O notation. For high-dimensional feature selection problems, the theoretical comparison highlights the structural advantage of incorporating a filter phase.

Table 11. Time complexity analysis.

Algorithm Type	**Representative Algorithms**	**Theoretical Time Complexity**
Wrapper-based Metaheuristics	BFPA, BWOA, TVSSA, TMGWO, BSCA, BJA, BDE, BCSA, BBAT, BBPSO	$O(T \times P \times N \times D)$
Hybrid Filter-Wrapper (Proposed)	FWMBFPA	$O(D^2 \times N + T \times P \times N \times D')$

5.3.5. Discussion on Observed Performance and Potential Limitations

The experimental results presented in Tables 4, 8, and 9 show that the proposed FWMBFPA exhibits significant improvements in performance compared to BFPA and several other metaheuristic algorithms, especially in achieving higher classification accuracy with a substantially reduced number of selected features on many datasets. In particular, datasets like 'Pd-speech-feature' and 'penglung' show dramatic feature reductions and significant improvements in accuracy.

This magnitude of improvement, especially when combined with substantial feature reduction and increased accuracy, may seem unusual, raising concerns about dataset bias. The validity of our experimental setup and results has been thoroughly reviewed. For reducing random effects, experiments were run using a standard method, using an 80/20 train/test split and the recommended K-nearest neighbor classifier (K=5). In FWMBFPA, the observed performance is primarily the result of the synergistic effects of the integrated filter and wrapper phases. Spearman correlation and relevance measures are used in the initial filter phase to eliminate highly redundant and irrelevant features before the search process even begins. Preprocessing reduces the search space and provides the wrapper phase with a more refined, less noisy set of candidate features. Therefore, the Modified Binary Flower Pollination Algorithm in the wrapper phase can better explore and exploit this reduced, relevant feature space to identify a truly optimal or near-optimal subset. As a result of this two-stage approach, FWMBFPA avoids becoming trapped in local optima caused by irrelevant or redundant features and focuses instead on choosing the discriminative subset, which results in both higher accuracy and a significantly smaller feature set on certain datasets, particularly those with very high initial dimensionality and presumably a high percentage of irrelevant/redundant features.

However, it is important to consider potential limitations of FWMBFPA, even though it has shown good performance across the evaluated UCI datasets. The filter phase's effectiveness is determined by the dataset's characteristics, namely its redundancy and irrelevance. In datasets with highly interacting features or less clear-cut redundancy, the initial reduction may be less drastic. The performance of metaheuristic algorithms can also be influenced by parameter tuning and stochasticity. In the future, the method should be evaluated on a wider range of dataset types and its robustness should be enhanced by exploring adaptive thresholds.

6. Conclusion and Future Work

Dimensionality reduction is essential in many fields because of big data. In this work, a hybrid version of the modified flower pollination algorithm inspired by nature was presented. To reduce the computational overhead and costs associated with the dataset, two filter methods were applied in the first step. In the wrapping step, an optimal set of features has been selected by the modified flower pollination algorithm after redundant and irrelevant features have been discarded. Besides FPA and FWMBFPA, nine other algorithms were evaluated using 18 standard UCI datasets. A KNN classifier was used to learn classification rules. FWMBFPA significantly improved classification accuracy as well as feature selection over FPA. A robust and stable approach is also demonstrated using standard evaluation criteria. FWMBFPA has shown superior performance in terms of accuracy and feature reduction on the evaluated datasets, particularly as a result of the efficient pre-processing by the filter phase. However, the degree of improvement may vary depending on the characteristics of the dataset. To further enhance its robustness and generalizability, this framework will be applied to a wider variety of real-world problems, adaptive filter thresholds explored, and its performance examined on datasets with different feature dependency structures. In the future, parallel processing will speed up the training of classifiers since many feature vectors constitute a computational bottleneck.

Declarations

Ethics approval and consent to participate

Not applicable

Consent for publication

Not applicable

Availability of data and materials

The datasets generated and/or analyzed during the current study are not publicly available due but are available from the corresponding author on reasonable request.

Competing interests

The authors declare that they have no competing interests

Funding

Not applicable

Authors' contributions

Mohammad Ansari Shiri: Programming, software development, Ideas
Najme Mansouri: Testing of existing code components, writing- original draft preparation

References

[1] P. Dhal and C. Azad, “A lightweight filter based feature selection approach for multi-label text classification,” *J. Ambient Intell. Humaniz. Comput.*, vol. 14, no. 9, pp. 12345–12357, 2023.
[2] X. Cui, Y. Li, J. Fan, and T. Wang, “A novel filter feature selection algorithm based on relief,” *Appl. Intell.*, vol. 52, no. 5, pp. 5063–5081, 2022.
[3] S. S. Shekhawat, H. Sharma, S. Kumar, A. Nayyar, and B. Qureshi, “bSSA: binary salp swarm algorith,” *Ieee Access*, vol. 9, pp. 14867–14882, 2021.
[4] F. Kılıç, Y. Kaya, and S. Yildirim, “A novel multi population based particle swarm optimization for feature selection,” *Knowl.-Based Syst.*, vol. 219, p. 106894, 2021.

[5] M. Kelidari and J. Hamidzadeh, “Feature selection by using chaotic cuckoo optimization algorithm with levy flight, opposition-based learning and disruption operator,” *Soft Comput.*, vol. 25, no. 4, pp. 2911–2933, 2021.
[6] D. Wang, H. Chen, T. Li, J. Wan, and Y. Huang, “A novel quantum grasshopper optimization algorithm for feature selection,” *Int. J. Approx. Reason.*, vol. 127, pp. 33–53, 2020.
[7] Y. Xue, X. Cai, and W. Jia, “Particle swarm optimization based on filter-based population initialization method for feature selection in classification,” *J. Ambient Intell. Humaniz. Comput.*, vol. 14, no. 6, pp. 7355–7366, 2023.
[8] N. S. M. Nafis and S. Awang, “An enhanced hybrid feature selection technique using term frequency-inverse document frequency and support vector machine-recursive feature elimination for sentiment classification,” *Ieee Access*, vol. 9, pp. 52177–52192, 2021.
[9] L. Peng, Z. Cai, A. A. Heidari, L. Zhang, and H. Chen, “Hierarchical Harris hawks optimizer for feature selection,” *J. Adv. Res.*, vol. 53, pp. 261–278, 2023.
[10] K. K. Ghosh, S. Ahmed, P. K. Singh, Z. W. Geem, and R. Sarkar, “Improved binary sailfish optimizer based on adaptive β-hill climbing for feature selection,” *IEEE Access*, vol. 8, pp. 83548–83560, 2020.
[11] W. Ma, X. Zhou, H. Zhu, L. Li, and L. Jiao, “A two-stage hybrid ant colony optimization for high-dimensional feature selection,” *Pattern Recognit.*, vol. 116, p. 107933, 2021.
[12] M. Mafarja *et al.*, “Evolutionary population dynamics and grasshopper optimization approaches for feature selection problems,” *Knowl.-Based Syst.*, vol. 145, pp. 25–45, 2018.
[13] M. M. Mafarja and S. Mirjalili, “Hybrid whale optimization algorithm with simulated annealing for feature selection,” *Neurocomputing*, vol. 260, pp. 302–312, 2017.
[14] H. Faris *et al.*, “Time-varying hierarchical chains of salps with random weight networks for feature selection,” *Expert Syst. Appl.*, vol. 140, p. 112898, 2020.
[15] A. Faramarzi, M. Heidarinejad, B. Stephens, and S. Mirjalili, “Equilibrium optimizer: A novel optimization algorithm,” *Knowl.-Based Syst.*, vol. 191, p. 105190, 2020.
[16] P. Agrawal, H. F. Abutarboush, T. Ganesh, and A. W. Mohamed, “Metaheuristic algorithms on feature selection: A survey of one decade of research (2009-2019),” *Ieee Access*, vol. 9, pp. 26766–26791, 2021.
[17] A. M. Anter and M. Ali, “Feature selection strategy based on hybrid crow search optimization algorithm integrated with chaos theory and fuzzy c-means algorithm for medical diagnosis problems,” *Soft Comput.*, vol. 24, no. 3, pp. 1565–1584, 2020.
[18] L. Abualigah, B. Alsalibi, M. Shehab, M. Alshinwan, A. M. Khasawneh, and H. Alabool, “A parallel hybrid krill herd algorithm for feature selection,” *Int. J. Mach. Learn. Cybern.*, vol. 12, pp. 783–806, 2021.
[19] M. Alweshah, S. Alkhalaileh, D. Albashish, M. Mafarja, Q. Bsoul, and O. Dorgham, “A hybrid mine blast algorithm for feature selection problems,” *Soft Comput.*, vol. 25, pp. 517–534, 2021.
[20] E. Nabil, “A modified flower pollination algorithm for global optimization,” *Expert Syst. Appl.*, vol. 57, pp. 192–203, 2016.
[21] Z. A. A. Alyasseri, A. T. Khader, M. A. Al-Betar, M. A. Awadallah, and X.-S. Yang, “Variants of the flower pollination algorithm: a review,” *Nat.-Inspired Algorithms Appl. Optim.*, pp. 91–118, 2018.
[22] X.-S. Yang, “Flower pollination algorithm for global optimization,” in *International conference on unconventional computing and natural computation*, Springer, 2012, pp. 240–249.
[23] M. Abdel-Basset and L. A. Shawky, “Flower pollination algorithm: a comprehensive review,” *Artif. Intell. Rev.*, vol. 52, pp. 2533–2557, 2019.
[24] M. Abdel-Basset, R. Mohamed, S. Saber, S. S. Askar, and M. Abouhawwash, “Modified flower pollination algorithm for global optimization,” *Mathematics*, vol. 9, no. 14, p. 1661, 2021.
[25] S. Mirjalili, A. H. Gandomi, S. Z. Mirjalili, S. Saremi, H. Faris, and S. M. Mirjalili, “Salp Swarm Algorithm: A bio-inspired optimizer for engineering design problems,” *Adv. Eng. Softw.*, vol. 114, pp. 163–191, 2017.
[26] A. E. Hegazy, M. A. Makhlouf, and G. S. El-Tawel, “Feature selection using chaotic salp swarm algorithm for data classification,” *Arab. J. Sci. Eng.*, vol. 44, pp. 3801–3816, 2019.

[27] A. K. Naik, V. Kuppili, and D. R. Edla, "Efficient feature selection using one-pass generalized classifier neural network and binary bat algorithm with a novel fitness function," *Soft Comput.*, vol. 24, no. 6, pp. 4575–4587, 2020.

[28] Y. Gao, Y. Zhou, and Q. Luo, "An efficient binary equilibrium optimizer algorithm for feature selection," *IEEE Access*, vol. 8, pp. 140936–140963, 2020.

[29] S. Saremi, S. Mirjalili, and A. Lewis, "Grasshopper optimisation algorithm: theory and application," *Adv. Eng. Softw.*, vol. 105, pp. 30–47, 2017.

[30] S. Zhao, P. Wang, A. A. Heidari, X. Zhao, C. Ma, and H. Chen, "An enhanced Cauchy mutation grasshopper optimization with trigonometric substitution: engineering design and feature selection," *Eng. Comput.*, vol. 38, no. Suppl 5, pp. 4583–4616, 2022.

[31] B. Doğan and T. Ölmez, "A new metaheuristic for numerical function optimization: Vortex Search algorithm," *Inf. Sci.*, vol. 293, pp. 125–145, 2015.

[32] F. S. Gharehchopogh, I. Maleki, and Z. A. Dizaji, "Chaotic vortex search algorithm: metaheuristic algorithm for feature selection," *Evol. Intell.*, vol. 15, no. 3, pp. 1777–1808, 2022.

[33] N. Bacanin, K. Venkatachalam, T. Bezdan, M. Zivkovic, and M. Abouhawwash, "A novel firefly algorithm approach for efficient feature selection with COVID-19 dataset," *Microprocess. Microsyst.*, vol. 98, p. 104778, 2023.

[34] Y. Liu, X. Zou, S. Ma, M. Avdeev, and S. Shi, "Feature selection method reducing correlations among features by embedding domain knowledge," *Acta Mater.*, vol. 238, p. 118195, 2022.

[35] J. R. Vergara and P. A. Estévez, "A review of feature selection methods based on mutual information," *Neural Comput. Appl.*, vol. 24, pp. 175–186, 2014.

[36] L. Wang, S. Jiang, and S. Jiang, "A feature selection method via analysis of relevance, redundancy, and interaction," *Expert Syst. Appl.*, vol. 183, p. 115365, 2021.

[37] M. Mafarja, A. Qasem, A. A. Heidari, I. Aljarah, H. Faris, and S. Mirjalili, "Efficient hybrid nature-inspired binary optimizers for feature selection," *Cogn. Comput.*, vol. 12, pp. 150–175, 2020.