Journal of Universal Computer Science, vol. 28, no. 5 (2022), 499-539
submitted: 19/11/2021, accepted: 3/3/2022, appeared: 28/5/2022

An Enhanced Evolutionary Based Feature Selection Approach Using Grey Wolf Optimizer for the Classification of High-dimensional Biological Data

Thaer Thaher
(Department of Computer Systems Engineering, Arab American University, Jenin, Palestine
https://orcid.org/0000-0001-7047-5886, Thaer.Thaher@aaup.edu)

Mohammed Awad
(Department of Computer Systems Engineering, Arab American University, Jenin, Palestine
https://orcid.org/0000-0002-5053-0785, mohammed.awad@aaup.edu)

Mohammed Aldasht
(Department of Computer Engineering, Palestine Polytechnic University, Hebron, Palestine
https://orcid.org/0000-0003-2757-2204, Mohammed@ppu.edu)

Alaa Sheta
(Computer Science Department, Southern Connecticut State University, CT 06514, USA
https://orcid.org/0000-0002-3727-6276, shetaa1@southernct.edu)

Hamza Turabieh
(Department of Information Technology, Collage of Computers and Information Technology, Taif University, Taif, Saudi Arabia
https://orcid.org/0000-0002-8103-563X, h.turabieh@tu.edu.sa)

Hamouda Chantar
(Faculty of Information Technology, Sebha University, Sebha 18758, Libya
https://orcid.org/0000-0003-2794-8144, ham.chantar@sebhau.edu.ly)

Abstract: Feature selection (FS) is a pre-processing step that aims to eliminate the redundant and less-informative features to enhance the performance of data mining techniques. It is also considered as one of the key factors for improving classification problems in high-dimensional datasets. This paper proposes an efficient wrapper feature selection method based on Grey Wolf Optimizer (GWO). GWO is a recent metaheuristic algorithm that has been widely employed to solve diverse optimization problems. However, GWO mainly follows the search directions toward the leading wolves, making it prone to fall into local optima, especially when dealing with high-dimensional problems, which is the case when dealing with many biological datasets. An enhanced variation of GWO called EGWO, that adapts two enhancements, is introduced to overcome this specific shortcoming. In the first mode, a transition parameter is incorporated to move GWO from the exploration phase to exploitation phase. In addition, several adaptive non-linear decreasing formulas are introduced to control the transition parameters. In the second mode, a random-based search strategy is exploited to empower the diversity of the search process. Two binarization schemes using S-shaped and V-shaped transfer functions are incorporated to map the continuous search space into a binary one for dealing with FS problem. The efficiency of the proposed EGWO is validated on ten high-dimensional low-samples biological data. Our experimental results show promising performance of EGWO compared to the original GWO approach as well as other state-of-the-art techniques in terms of dimensionality reduction and the enhancement of classification performance.

Keywords: Feature selection, Enhanced grey wolf optimizer, Binary grey wolf optimizer, Classification, Biological data, Exploration, Metaheuristics
Categories: H.3.1, H.3.2, H.3.3, H.3.7, H.5.1
DOI: 10.3897/jucs.78218

1 Introduction

Recently, there has been a growing interest to deal with high-dimensional data classification problems. The reason is that the high-dimensional data classification causes a substantial statistical burden, and makes conventional classification algorithms unfeasible to use [Pappu and Pardalos, 2013]. High dimensional data classification can be found in many real world applications, including medical diagnosis of tumors based on micro-array data, sentiment classification of online reviews [Ghaddar and Naoum-Sawaya, 2018], emotion recognition from ECG signals [Sepúlveda et al., 2021], and diagnosis of severity rating of Parkinson's disease [Balaji et al., 2021]. In general, the quality of data is a crucial factor that deeply impacts the performance of data mining techniques such as classification. For instance, the presence of irrelevant or redundant information may adversely affect the predictive ability of the machine learning algorithms. In real-life, most problems are high dimensional. When taking biological data, for example, an excessive number of features are usually collected during the data acquisition phase. So, filtering the biological data for extracting the most valuable information (i.e., features) is considered as a complex and time-consuming task [Lin et al., 2019]. That is to say, having a dataset with a large number of features may negatively affect the performance of the learning algorithm. Feature Selection (FS) is one of the most acceptable approaches that can be used to reduce the data dimensionality, which will enhance the overall performance of the learning process, and reduces execution time [Guyon and Elisseeff, 2003, Liu and Motoda, 2012].

FS is employed to find the most informative features that can be used to build a robust classification model. The main objective of FS is to reduce the dimensionality of a dataset by eliminating the unnecessary (i.e., irrelevant and redundant) features. Moreover, removing irrelevant features will lead to reduce the memory usage, enhance the learning process and decreasing the computational time required for performing the classification task [Zhao et al., 2010]. FS is a challenging multi-objective optimization problem that aims to find the minimum subset of the most relevant features for data classification task (i.e., preserving the maximum classification performance). Based on the criteria used for assessing the quality of the chosen subset of features, FS methods are categorized into two prominent families: filters or wrappers [Chantar et al., 2021]. In the filter methods, features are chosen based on their scores determined using various statistical tests for quantifying their correlation with the response variable. In this way, the selection of features is independent of any machine learning algorithms. In contrast, wrapper FS approaches involve two components to find the optimal subset of features: search algorithm and evaluation method. The search strategy, which is mostly a heuristic algorithm, is employed to explore the search space for finding the ideal subset of features, while a specific machine learning algorithm is used to evaluate the goodness of the features subset provided by the search algorithm. In general, wrapper FS, compared to filter FS, can achieve better in terms of classification accuracy since it can expose and utilize dependencies between selected candidates in a subset of features [Chantar et al., 2020]. Among the various FS techniques, wrapper methods have been successfully applied in many recent studies [Nguyen et al., 2020, Al-Tashi et al., 2020, Xue et al., 2016, Alweshah et al., 2021, Awadallah et al., 2022, Awadallah et al., 2020].

Generating a high informative subset of features is a challenging search task. Three possible strategies can be used to address the FS problem: exhaustive search, random search, and heuristic search strategies [Mafarja and Mirjalili, 2017]. In the exhaustive search, FS methods examine a large number of subsets to find the most accurate one. For example, if the original dataset has N features (i.e., inputs), there will be $2^N - 1$ subsets generated from the original dataset. Examining all possible subsets my lead to find the best subset based on the evaluation criterion. In this case, the execution time will be high and not practical to apply an exhaustive search for real problems. In the case of random search strategy, searching for the subsequent feature subset in the feature space is performed in a random way [Lai et al., 2006]. Similar to complete search, the random search may lead in some cases to generate all possible subsets of features [Talbi, 2009]. To overcome the limitation of exhaustive search methods, heuristics and metaheuristics search can be used as effective alternative methods to find the best subset of features in an acceptable execution time [Glover and Kochenberger, 2006]. Metaheuristic algorithms usually come in two flavors: single-based solution (i.e., trajectory-based) or population-based algorithms. Both methods try to enhance that solution/population in an iterative manner until the stop condition is achieved. However, the population-based methods try to gain a balance between exploration and exploitation processes. This balancing makes population-based approaches more reliable for complex problems with high dimensional input data [Talbi, 2009]. To date, metaheuristics algorithms show an excellent performance for solving real complex problems in FS domains [Nguyen et al., 2020, Mafarja et al., 2020, Hassouneh et al., 2021].

Currently, Swarm Intelligence (SI) based metaheuristics are widely applied as wrapper FS approaches [Rostami et al., 2021]. In 2014, a new SI-based algorithm was proposed by Mirjalili *et al.* [Mirjalili et al., 2014] called Grey Wolf Optimizer (GWO). It simulates the behavior of grey wolves in nature. Since 2014, GWO have been employed successfully in variety of real-world optimization problems such as scheduling problems [Abed-alguni and Alawad, 2021], text mining [Chantar et al., 2019], aerial vehicles path planning [Qu et al., 2020] and others [Erdogan et al., 2021, Deveci and Çetin Demirel, 2018, Çetin Demirel and Deveci, 2017, Akyurt et al., 2021, Faris et al., 2017, Sharma et al., 2020]. GWO algorithm has several advantages such as it is simple, easy to use, has fewer parameters that need to be tuned, and has an excellent switching mechanism between exploration and exploitation processes while searching for an optimal solution [Faris et al., 2017].

Despite the merits of GWO, it suffers, as most meta-heuristics, from the problem of premature convergence. The primary search strategy of the GWO is mainly based on following the search trends towards the best solutions (i.e., leading wolves), making it prone to be trapped early in the local optima, especially when dealing with high-dimension problems [Gupta and Deep, 2020]. According to [Faris et al., 2017, Heidari and Pahlavani, 2017b, Long et al., 2018, Lu et al., 2018, Tu et al., 2019], the problem of early convergence in the GWO is due to two main weaknesses: the lack of population diversity and the insufficient balance between exploration and exploitation behaviors. Consequently, the research is still open to provide more operators that will emphasize exploration potential. These reasons constitute a motivation for embedding different strategies into GWO for improving its performance.

The above-stated limitations of GWO, along with the nature of the NP-Hard FS optimization problem, are the main foundation and motivation of this research. Additionally, the No-Free-Lunch (NFL) theorem [Wolpert and Macready, 1997] for optimization suggests that a universal algorithm that guarantees effective performance to handle all problems is impossible [Ho and Pepyne, 2002]. Furthermore, the presence of challenging classification tasks such as classifying biological and medical detests is another reason that motivated the authors of this work to develop a robust feature selection approach to tackle these problems. In this study, the authors propose an efficient wrapper-based FS method for handling the classification task of high-dimensional biological data. The proposed wrapper feature selection method employs an enhanced variant of GWO as a search algorithm. The main contributions of this research are summarized as follows:

- Different binarization schemes are investigated to adapt the GWO for handling the FS problem.
- An enhanced variant of GWO called EGWO is proposed to empower the exploration potential and make it better for the high-dimensional FS problem. The enhancement included boosting the original GWO with a random-based exploration operator and introducing the idea of the transition control parameter to switch between exploration and exploitation using different adaptive non-linear convergence shapes.
- The proposed approach is validated through ten challenging biological datasets.
- The efficiency of the proposed EGWO is verified by comparing it with six well-regarded FS algorithms and offered promising results.

The rest of the paper is structured as follows: Section 2 explores a set of previous works within the same research field. Section 3 presents an overview of the GWO algorithm, including the inspiration and mathematical model. Section 4 is dedicated to explaining the proposed approach deeply. The experimental results are reported and discussed in section 5. Finally, Section 6 concludes the overall work and presents the future direction.

2 Related Works

2.1 Meta-heuristic based feature selection

The literature reveals that FS plays a major role in many machine learning and classification tasks. Recently, a wide range of SI-based algorithms have been integrated as search strategies in various wrapper FS approaches. Instances of founded metaheuristics-based FS approaches include Particle Swarm Optimization (PSO) [Xue et al., 2014], GWO [Emary and Zawbaa, 2016], Harris Hawks Optimization (HHO) [Thaher et al., 2020b, Thaher et al., 2021b], Slim Mould Algorithm (SMA) [Abdel-Basset et al., 2021], Moth-Flame Optimization (MFO) [Tumar et al., 2020, Abu Khurma et al., 2021], Whale Optimization Algorithm (WOA) [Mafarja et al., 2020, Hassouneh et al., 2021, Mafarja et al., 2019], Dragonfly Algorithm (DA) [Hammouri et al., 2020], Marine Predators Algorithm (MPA) [Elminaam et al., 2021], and others [Nguyen et al., 2020]. For instance, Xue et al. [Xue et al., 2014] investigated two PSO-based multi-objective FS approaches for solving classification problems. The first algorithm uses the idea of non-dominated sorting, while the second one exploits the ideas of mutation, crowding, and dominance. The performance of the proposed multi-objective algorithms was compared with two classical FS approaches, a single objective and two-stage FS algorithms using twelve benchmark data sets. The experimental results indicated that the two PSO-based multi-objective algorithms could deliver comparable results. In addition, Emary et al. [Emary et al., 2015] introduced a binary version of Firefly algorithm (FFA) to tackle FS tasks using a threshold value. The modified FFA algorithm can adaptively provide a proper balance between exploration and exploitation and find the optimal solution quickly. The proposed approach was examined on eighteen data sets and confirmed its superiority over other methods such as PSO and genetic algorithm (GA). Sayed et al. [Ismail Sayed et al., 2018] proposed a chaotic whale optimization algorithm (CWOA) where ten chaotic map functions were applied. The chaotic maps were used instead of random parameters to gain a better trade-off between the exploration and exploitation phases. Medjahed et al. [Medjahed et al., 2017] proposed a complete diagnosis procedure of cancer based on binary dragonfly (BDF) algorithm with an SVM classifier. SVM-recursive feature elimination was applied to extract the gene from the datasets, and BDF was utilized to improve the performance of SVM-RFE. The proposed approach was applied over six microarray datasets and provided high satisfactory accuracy results.

As presented in [Emary and Zawbaa, 2016], three FS approaches based on meta-heuristic algorithms, namely GWO, ant lion optimizer (ALO), and MFO were proposed. Two chaos functions were applied for controlling the exploration rate in the three proposed FS approaches. Results of conducted experiments on a set of datasets obtained from the UCI machine learning repository indicated that chaos functions could provide better exploration and exploitation and hence, excellent performance when applied with GWO and ALO. To deal with the binary FS problem, Thaher et al. [Thaher et al., 2020b] proposed a new binary Harris Hawks Optimization algorithm. The binary HHO was evaluated on high dimensional with low number of samples datasets. The results revealed that the HHO-based FS technique could be applied as a promising approach to dealing with high dimensional with few samples datasets. Mafarja and Mirjalili [Mafarja and Mirjalili, 2017] proposed two models for FS based on Whale Optimization Algorithm (WOA) and Simulated Annealing (SA) algorithms. The proposed approaches were tested on 18 standard benchmark datasets obtained from the UCI repository. The experimental results approved the efficiency of the proposed WOA-based FS approaches in selecting the most informative features for classification tasks. Finally, AgrawAal et al. [Agrawal et al., 2021] presented an extensive literature review on dealing with feature selection problems using metaheuristic algorithms.

2.2 Applications of GWO Algorithm

GWO is considered as one of the effective optimization algorithms that have been proposed in the last few years. Since its appearance, it has been widely used to solve many optimization problems. For instance, [Mittal et al., 2016a] GWO was applied to deal with Cluster head (CH) selection problem in wireless sensor networks (WSNs) field. Wen et al. [LONG Wen, 2015] employed GWO to tackle constrained conditions for solving the problem of non-stationary multi-stage assignment. Various complex constrained optimization problems and a classical engineering design problem named pressure vessel were solved by Joshi and Arora [Arora and Joshi, 2017] using the GWO algorithm. In addition, distributed Compressed Sensing (DCS) problem was tackled by Liu et al. [Liu et al., 2018] using a combination of GWO and q-thresholding algorithms. GWO was applied by Lu et al. [Lu et al., 2017] to deal with the problem of welding scheduling in the modern industry domain. Debnath et al. [Debnath et al., 2017] proposed a model based on GWO and DE algorithms for dealing with the problem of an automatic power production control in an interconnected multi-source power system. Furthermore, GWO is also utilized for solving the problem of economic load dispatch (ELD) [Jayabarathi et al., 2016]. In [Liu and Wang, 2021], GWO in conjunction with RNA encoding crossover-operation was introduced to deal with the non-parametric modeling problem of the FCC process. Farughi et al. [Farughi et al., 2019] applied GWO and ALO optimizers for solving the problem of population districting in health systems. Faris et al. [Faris et al., 2018a] and Negi et al. [Negi et al., 2020] provide reviews of GWO-based applications related to various fields.

In addition to the above-mentioned works, many researchers have adopted the GWO algorithm as a wrapper FS method [Al-Tashi et al., 2020]. For example, a novel approach is proposed by Qiang et al. [Li et al., 2017] that combined GA with GWO. Here, the authors used GA as a tool to generate the initial population to keep a high diversity rate, while GWO is employed as a search algorithm for updating the initial population. Emary and Zawbaa [Emary and Zawbaa, 2016] enhanced the performance of GWO by using some chaotic maps rather than random numbers to find a good balance between exploration and exploitation processes.

Lately, a multi-strategy ensemble GWO called MEGWO was introduced by Tu *et al.* [Tu et al., 2019] to enhance the diversification and intensification of the conventional GWO. To overcome the limitations of GWO, three different search strategies were incorporated: the adaptive cooperative strategy, the enhanced global best strategy, and disperse foraging strategy. Experimental results revealed the superiority of MEGWO in dealing with FS problems. In addition, [Chantar et al., 2020] solved the FS problem for the Arabic text classification task utilizing the GWO-based wrapper FS approach. The authors incorporated an improved GWO using an elite-based crossover scheme as a search strategy. Promising results were achieved compared to other state-of-the-art methods. Moreover, Too and Abdullah[Too and Abdullah, 2020] developed two binary variants of the recently established Competitive GWO (CGWO) and an opposition-based CGWO (OBCGWO) to tackle the FS problem in electromyography (EMG) pattern recognition. The experimental results confirmed that the OBCGWO yielded better classification performance. Furthermore, Abdel-Basset *et al.*[Abdel-Basset et al., 2020] proposed a wrapper-based FS approach using a new fusion of GWO integrated with a two-phase mutation. Following a different FS approach, [Singh et al., 2020] introduced a non-linear FS Network (FsNet) utilizing a concrete Neural Network (NN) structure comprised of a selection layer for FS and deep NN. The comprehensive survey about FS methods can be found in [Xue et al., 2016, Nguyen et al., 2020]

Recently, Al-Wajih *et al.*[Al-Wajih et al., 2021] proposed a binary hybrid approach called HBGWOHHO by combining GWO with HHO to enhance the performance of the GWO algorithm for tackling the FS problem. An improved binary GWO was proposed by Hu *et al.*[Hu et al., 2020] for FS tasks. The authors conducted a mathematical analysis of the range of AD values in the binary variant of GWO. Based on their analysis, new transfer functions were introduced, and a new updating strategy for the a parameter was proposed to balance the exploration and exploitation potentials. Another GWO-based wrapper FS approach was proposed in [P et al., 2021]. The authors integrated GWO to search the useful features to improve the performance of the botnet attack detection system.

In summary, the inspected related works confirm the effectiveness of the SI-based metaheuristics, precisely the GWO method, for FS tasks in various fields. Most of the previous methods employed transfer functions to switch the nature of SI methods from continuous to discrete (i.e., binary) structures. Moreover, many SI algorithms have been enhanced to escape from premature convergence. These algorithms' ability to explore the search space encourages many researchers to adopt them for solving complex optimization problems in FS domains. In general, swarm algorithms need a proper setting to balance the exploration and exploitation processes. Moreover, most swarm algorithms have only one parameter that controls the ratio between exploration and exploitation. For example, PSO has a parameter called inertia weight (ω) that controls both processes. However, many papers try to enhance the controlling process of this ratio (i.e., adaptively change). Harrison et al. [Harrison et al., 2016], highlighted the updating methods for the inertia weight parameter. Chuang et al. [Chuang et al., 2008] proposed a chaotic logistic map to update the inertia weight parameter. For GWO, there is only one parameter (i.e., a), which controls the switching process between exploration and exploitation. The nature of this parameter decreases linearly throughout iterations. This decreasing process enables GWO to focus on exploration at the beginning of the search process and perform more exploitation at the end of the search process. However, in some challenging FS tasks, such as those with massive features, efficient exploration becomes essential to increase the probability of discovering promising regions within the search space. Consequently, other updating strategies (i.e., nonlinear, logarithm, etc.)) that can guarantee a better balance between exploration and exploitation are desired for the problem at hand. Moreover, The standard GWO allows the search agents to be updated based on the best three agents (i.e., $\alpha, \beta, and \gamma$). In specific, GWO is prone to stagnation in local optima areas. Therefore, better operators to emphasize exploration is required.

In addition to the NFL theorem, these concerns form the motivations of our attempt to propose an enhanced variant of GWO for high-dimensional FS tasks. In the proposed approach, a random-based updating mechanism is employed to empower the exploration tendency of GWO. In addition, instead of the linear decreasing of the value of (a) parameter during the course of the optimization process, we employed different updating strategies to the main control parameter (a) to maintain the trade-off between exploration and exploitation processes. More accurately, non-linear decreasing strategies for updating (a) parameter are used instead of the current liner function in the GWO algorithm. It can be seen that many GWO-based approaches have been introduced by other researchers for dealing with several optimization problems, including FS tasks. In Table 1, we summarize the main proposed modifications on the original version of the GWO algorithm to tackle different optimization problems. In each case, the mechanism of updating the value of (a) parameter is mentioned to show the main difference between the proposed form of GWO algorithm in this work and previously presented modified forms of GWO algorithm. It is clear that almost all previously introduced modifications of GWO use the original linear equation for updating the value of (a) parameter during the course of iterations.

3 Grey Wolf Optimizer

GWO is a population-based optimization algorithm proposed by Mirjalili *et al.* in 2014 [Mirjalili et al., 2014]. GWO belongs to the SI family of metaheuristics. It mainly mimics the intelligent hunting strategy, and social organization of the grey wolves [Faris et al., 2017, Heidari and Pahlavani, 2017a]. Generally, grey wolves are live in herds in which wolves are organized in an interesting social hierarchy of four levels. Figure 1 presents the main organization of each pack of grey wolves.

Cooperative hunting is another promising social behavior of grey wolves in addition to the social hierarchy structure. The hunting process is straightforward; in the first phase, wolves start tracking and chasing weak prey. The second phase begins by surrounding (i.e., pursuing, encircling, and harassing) the prey to prevent it from escape. The final

Proposed modification	Parameter (a)	Reference
Elite-based crossover scheme as a search strategy	Linearly decreased	[Chantar et al., 2020]
Combining GWO with Harris Hawks Optimization (HHO)	Linearly decreased	[Al-Wajih et al., 2021]
Combining GA with GWO	Linearly decreased	[Li et al., 2017]
Chaotic maps for better balancing between exploration and exploitation	Linearly decreased	[Emary and Zawbaa, 2016]
A multi-strategy ensemble GWO	Linearly decreased	[Tu et al., 2019]
Competitive GWO and an opposition-based GWO	Linearly decreased	[Too and Abdullah, 2020]
GWO integrated with a two-phase mutation	Linearly decreased	[Abdel-Basset et al., 2020]
A new updating strategy for the a parameter	Linearly decreased	[Hu et al., 2020]
Applying natural selection methods in the social hierarchy process of GWO	Linearly decreased	[Al-Betar et al., 2018]
A new updating strategy for the a parameter	Non-linearly decreased	[Ahmadi et al., 2021]
A new updating strategy for the a parameter	An Exponential function	[Mittal et al., 2016b]

Table 1: Proposed modifications of GWO algorithm for feature selection

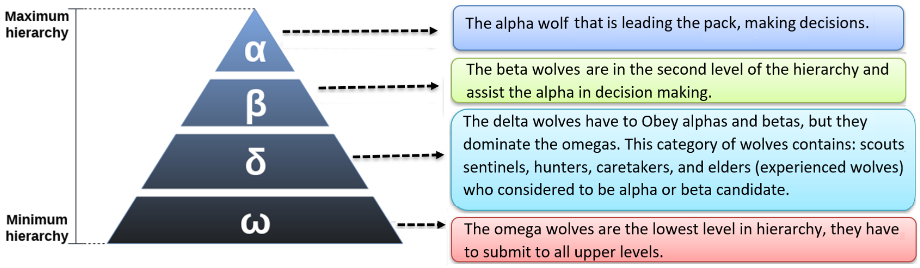


Figure 1: Social hierarchy of grey wolves.

phase is attacking and killing the prey. These techniques are mathematically modeled to propose the GWO optimization algorithm. In GWO, the herd of wolves is represented by the population of search agents (i.e., candidate solutions), and the best-known solution represents the prey. Each search agent's quality (fitness) is given by the evaluation function of the optimization problem being solved. Subsequently, the social hierarchy is modeled by considering the top three agents as α, β, and γ, respectively. The rest of the population is assumed to be the ω. Encircling behavior is the key to the hunting process. Eq. (1) presents the mathematical model of encircling process.

$$\vec{X}(t+1) = \vec{X}_p(t) - \vec{A}.\vec{D} \quad (1)$$

where t represents the current iteration, and $\vec{D}$ refers to the distance vector between the current wolf position $X\vec{(t)}$ and the prey position $\vec{X}_p(t)$. It can be evaluated based on

Eq.(2) .

$$\vec{D} = |\vec{C}.\vec{X}_p(t) - \vec{X}(t)| \tag{2}$$

where vectors $\vec{A}$ and $\vec{C}$ are coefficient vectors that are evaluated based on Eqs. (3), and (4), respectively. || denotes the absolute value, and · is an element-by-element multiplication. It is worth mentioning here that the dimension of vectors is equal to the number of variables (features) of the problem being solved.

$$\vec{A} = 2\vec{a} \cdot \vec{r}_1 - \vec{a} \tag{3}$$

$$\vec{C} = 2 \cdot \vec{r}_2 \tag{4}$$

where $\vec{r_1}$, $\vec{r_2}$ refer to a random vectors whose elements are within [0,1], the variable $\vec{a}$ is the main control parameter, its components are decreased in a linear manner from 2 to 0 over the course of iterations as given by Eq. (5),

$$\vec{a} = 2 \times (1 - \frac{t}{T}) \tag{5}$$

where t refers to the current iteration and T refers to the total number of iterations.

To simplify the effects of Eqs. (1) and (2), figure 2 depicts the encircling process in a two-dimensional search space. In simple, each wolf updates its position (X, Y) randomly based on the position of the prey $(X^{'}, Y^{'})$; where both vectors $\vec{A}$ and $\vec{C}$ are tuned to allow reach different places around the best solution. For instance, the position $(X^{'} - X, Y^{'})$ can be reached by setting $\vec{C} = (1, 1)$ and $\vec{A} = (1, 0)$. It is worth mentioning here that the random vectors r_1 and r_2 are employed for the stochastic purpose (i.e., to allow reach any random location between the points shown in figure 2). The same idea can be generalized to a search space with n dimensions.

After the encircling phase, the hunting process starts. Due to insufficient information on the optimal solution for most real-world problems, it is supposed that the best solution obtained so far represents the prey. The hunting process is mathematically simulated based on the fittest three solutions α, β, and γ, which are assumed to have better knowledge of the potential prey location. In specific, the best three candidate solutions obtained so far will guide the other search agents during the optimization process. The updating process is achieved based on the following formulas:

$$\vec{D}_\alpha = |\vec{C}_1.\vec{X}_\alpha(t) - \vec{X}(t)| \tag{6}$$

$$\vec{D}_\beta = |\vec{C}_2.\vec{X}_\beta(t) - \vec{X}(t)| \tag{7}$$

$$\vec{D}_\delta = |\vec{C}_3.\vec{X}_\delta(t) - \vec{X}(t)| \tag{8}$$

$$\vec{X}_1(t+1) = |\vec{X}_\alpha(t) - \vec{A}_1.\vec{D}_\alpha| \tag{9}$$

$$\vec{X}_2(t+1) = |\vec{X}_\beta(t) - \vec{A}_2.\vec{D}_\beta| \tag{10}$$

$$\vec{X}_3(t+1) = |\vec{X}_\delta(t) - \vec{A}_3.\vec{D}_\delta| \tag{11}$$

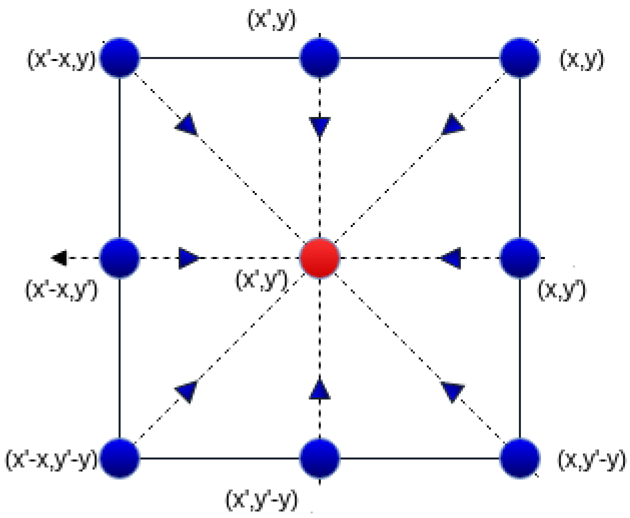


Figure 2: The encircling process of GWO in 2D space

$$\vec{X}(t+1) = \frac{\vec{X}_1(t) + \vec{X}_2(t) + \vec{X}_3(t)}{3} \tag{12}$$

where the vectors $\vec{X}_\alpha(t)$, $\vec{X}_\beta(t)$, $\vec{X}_\delta(t)$ represent the best locations of all types of wolves in the current iteration t. $\vec{C}_1$, $\vec{C}_2$, and $\vec{C}_3$ are evaluated based on Eq.(4), $\vec{X}(t)$ represents the location of the present solution, and $\vec{D}_\alpha$, $\vec{D}_\beta$, $\vec{D}_\delta$ are the distances between the current solution and the best three solutions, respectively. The three vectors $\vec{A}_1$, $\vec{A}_2$, $\vec{A}_3$ are calculated as in equation (3). Figure (3) demonstrates the potential next position of a search agent in a 2D search space. It can be noticed that that the next position would be in a random place within a circle defined by the positions of α, β, and γ. In alternative words, the position of the prey is estimated by α, β, and γ. While the other wolves modify their positions randomly around the prey.

GWO is a global search algorithm; the proposed mathematical operators adjust the ability of exploration and exploitation. In this regard, the main parameter (a) decreases linearly from 2 to 0 throughout iterations. Accordingly, the coefficient A fluctuates dynamically inside [-a, a] as defined in Eq. (3). The exploration process is achieved when $|A| \geq 1$, and the exploitation process will be achieved when $|A| < 1$. That is to say, when random values of A are inside [-1, 1], the new position of a search agent can be in any position between its original position and the estimated position of the prey. In contrast, with $|A| \geq 1$, the new position can fall outside the circle defined by α, β, and γ (see Figure (3). Algorithm (1) presents the Pseudo code of the GWO.

It can be noted that A and C are the main factors that control the exploration and detection tendency of GWO. Therefore, providing improvements to these parameters and introducing more exploration operators will effectively contribute to solving the problem of stagnation in local solutions, especially when dealing with high-dimensional

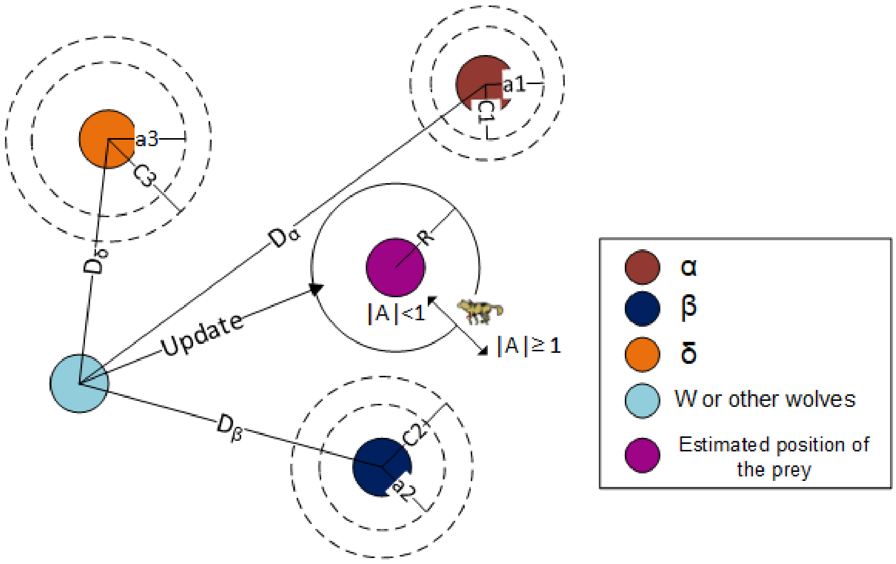


Figure 3: Position updating mechanism for each search agent in GWO

classification tasks [Mirjalili et al., 2014, Chen et al., 2021].

4 The Proposed Methods

4.1 Enhanced Grey Wolf Optimizer

The classical GWO has several merits. It is simple to implement, has fewer parameters to be tuned, and employs social hierarchy behavior. However, the major drawback of GWO is premature convergence. The algorithm lacks the required diversity during the optimization process and can easily be stuck into local optima. Therefore, several modifications have been introduced in the literature to improve the exploratory potential. In GWO, a parameter called (a) controls the balance between exploration and exploitation. This parameter is adaptively decreased over the iterations to provide more exploration at the early stage of the search process while providing more exploitation at the last stage.

In this study, we present an improved version of GWO intending to enhance the exploration feature. The pseudo-code of the enhanced GWO is shown in Algorithm 2. The proposed version depends on two enhancement schemes as follows:

4.1.1 Transition Control Parameter

We have introduced the transition control parameter concept inspired by WOA [Mirjalili and Lewis, 2016], and HHO [Heidari et al., 2019] algorithms. The idea is based on dividing the search process into two phases: exploration and exploitation. We exploit the transition parameter that already exists in GWO for the transition between these two phases smoothly. In the original GWO, a is reduced linearly from 2 to 0. The variation range of A is also reduced by a. Therefore, A is a random variable in the interval [-2a,

Algorithm 1 Pseudo code of the standard GWO

Set the initial parameters: population size (N), maximum iterations (T)
Generate the initial grey wolf population $X_i (i = 1, 2, \ldots, n)$
Assess the fitness of each search agent
Identify the best three search agents based on their fitness
X_α = the fittest search agent
X_β = the second fittest search agent
X_δ = the third fittest search agent
while ($t < T$) **do**
    for each search agent **do**
        Update the coefficients a, A, and C
        Use Eq. (12) to re-position (update) the current search agent
    end for
    Evaluate the fitness of new population
    Update the best three solutions X_α , X_β and X_δ
    $t = t + 1$
end while
return the best solution X_α

2a]. The parameter a is decreased from 2 to 0 to emphasize exploration and exploitation, respectively. Therefore, we introduce a transition parameter (tp) used to control the transition between exploration and exploitation potentials as in eq. (13).

$$\vec{tp} = 2\vec{a}.\vec{r} - \vec{a} \tag{13}$$

$$a = 2 - (2.\frac{t}{T}) \tag{14}$$

Based on the variation of the tp vector, we update the position of a search agent according to a randomly chosen search agent instead of the best three agents found so far (exploration phase). This mechanism |tp| > 1 emphasizes exploration and allows the GWO algorithm to perform a global search. In the exploitation phase (|tp|<1), we used the original operators of GWO.

4.1.2 Exploration Phase

In this phase, search agents are guided by a randomly selected solution to emphasize searching more areas in the large search space. Eqs.(15) and (16) shows the exploration process of EGWO.

$$\vec{D} = |\vec{C}.\vec{X}_{rand} - \vec{X}| \tag{15}$$

$$\vec{X}_{(t+1)} = \vec{X}_{rand} - \vec{tp}.\vec{D} \tag{16}$$

where $\vec{X}_{rand}$ is a randomly selection search agent.

4.1.3 Different Decreasing Shapes of Convergence Parameter (a)

In the original GWO, a is reduced linearly from 2 to 0. Other updating rules (nonlinear, logarithmic) are utilized to emphasize the required diversity for high-dimensional FS problems to increase the probability of discovering more promising regions. Accordingly, four decreasing shapes of a were utilized. The decreasing formulas and the corresponding variants of EGWO are presented below, while decreasing patterns of a and tp are shown in Figures 4 and 5. In addition to the original GWO algorithm, we have five enhanced variants, where each one has a special decreasing strategy applied to the parameter a, as follows:

- EGWO: the enhanced variant using exploration and exploitation phases where a decreases linearly, according to the formula in Eq. (14).
- EGWO1: the enhanced variant using decreasing strategy 1, according to the formula:

$$a = 2 - (\frac{2t^{1/3}}{T^{1/3}}) \tag{17}$$

- EGWO2: the enhanced variant using decreasing strategy 2, according to the formula:

$$a = 2 - (\frac{2 \times log(t+1)}{log(T)}) \tag{18}$$

- EGWO3: the enhanced variant using decreasing strategy 3, according to the formula:

$$a = 2 \times e^{-(\frac{4t}{T})^2} \tag{19}$$

- EGWO4: the enhanced variant using decreasing strategy 4, according to the formula:

$$a = (\frac{-2 \times t^3}{T^3}) + 2 \tag{20}$$

where t is the current iteration, and T is the maximum number of iterations.

4.2 Binary GWO for Feature Selection

GWO, as most metaheuristic methods, are designed to tackle problems in continuous search space. Accordingly, a binarization scheme is usually employed to adapt real-valued metaheuristics to match the discrete search space of the FS problem. For this purpose, two main techniques have been introduced in the literature. In the first technique called (continuous-binary operator), the original real-values operators are reformulated into binary operators. Whereas in the second technique, which is called two-step binarization, the original real operators are not redefined [Crawford et al., 2017]. To conduct the binarization, fuzzy Transfer Functions (TFs) are first employed to convert the real values

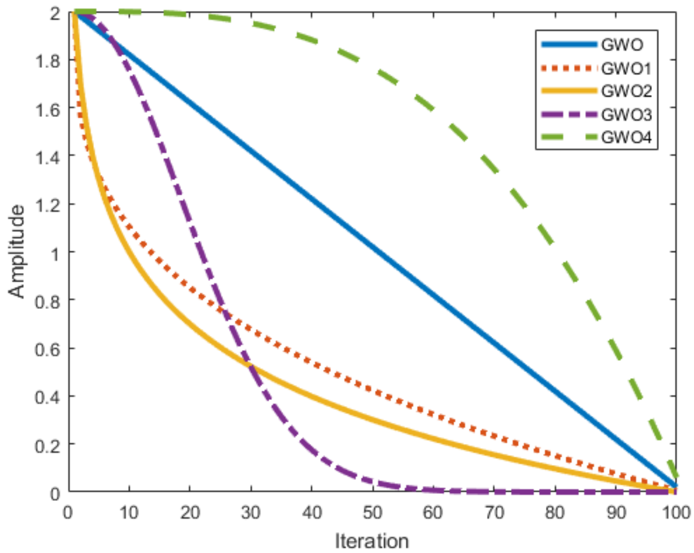


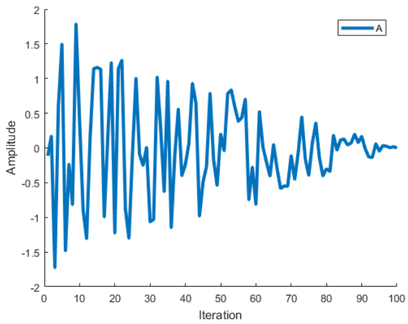
Figure 4: Shapes of convergence parameter a using different decreasing formulas

into intermediate probability values within [0,1]. Therefore, each element in the real solution is given a probability of being 0 or 1. In the second step, the outcomes of TFs are stochastically threshold using a specific rule to find the binary output. In common, the two-step binarization technique is the most popular method, which revealed success performance in different domains [Thaher et al., 2020a, Thaher et al., 2021a].

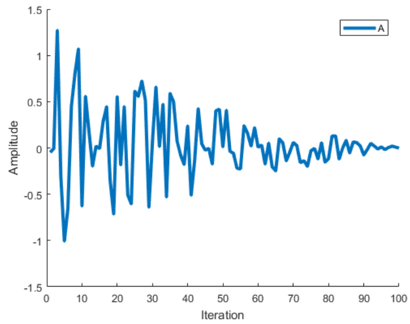
TFs are defined based on their shapes into two families: V-shaped and S-shaped [Mirjalili and Lewis, 2013] (see Figure 6). The literature reveals that two binary versions of the GWO were proposed by Zawbaa *et al.*[Zawbaa et al., 2016] for FS problem. In the first approach (called bGWO1), individual steps toward the best three leaders are binarized, and then stochastic crossover is performed among the three basic moves to find the updated binary gray wolf position (see [Zawbaa et al., 2016] for equations). In the second approach (called bGWO2), the S-shaped function was utilized to squash the continuously updated position, then stochastically threshold these values to find the updated binary gray wolf position. In both strategies, a sigmoid function as in Eq. (21) was utilized. In this work, we have employed the two proposed methods using S2 and V2 TFs. Therefore, four variants named: SBGWO1, VBGWO1, SBGWO2, VBGWO2 were introduced. Note that BGWO1, BGWO2 refer to the first and second approaches, respectively. Mathematical formulas of S2 TF with standard binarization rule are shown in Eqs (21) and (22). While Mathematical formulas of V2 TF with complement binarization rule are shown in Eqs (23) and (24).

$$T(X_j(t)) = \frac{1}{1+e^{-x_j(t)}} \tag{21}$$

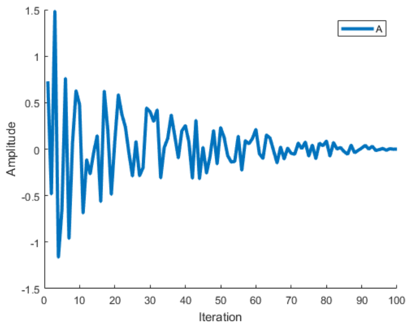
$$X_j(t+1) = \begin{cases} 1 & r < T(X_j(t)) \\ 0 & Otherwise \end{cases} \tag{22}$$



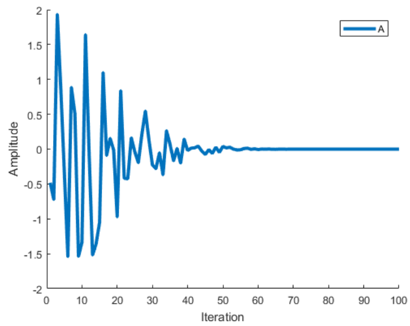
(a) GWO



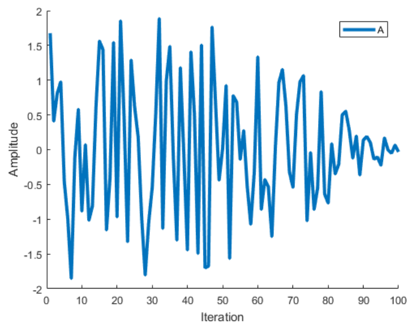
(b) EGWO1



(c) EGWO2



(d) EGWO3

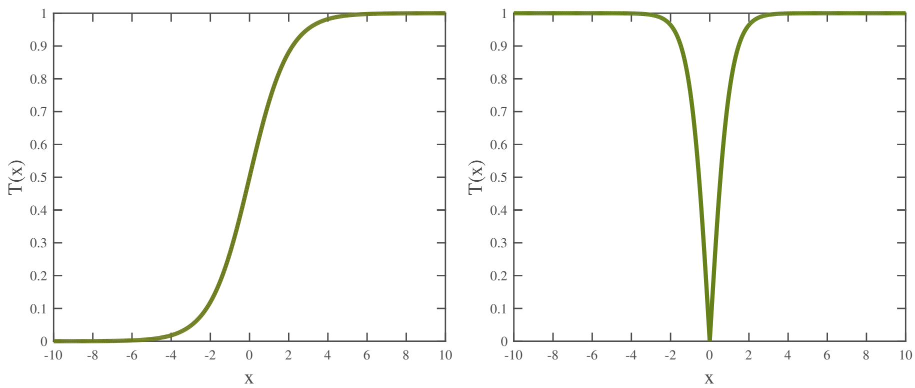


(e) EGWO4

Figure 5: Behavior of tp during 100 iterations for the standard GWO and the proposed variants

Algorithm 2 Pseudocode of the Enhanced GWO

```
Set the initial parameters: population size (N), maximum iterations (T)
Generate the population X_i (i = 1, 2, ..., n)
Assess the fitness of each search agents.
Identify the best three agents based on their fitness.
X_α = the fittest search agent
X_β = the second fittest search agent
X_δ = the third fittest search agent
while (t < T) do
    for each search agent do
        Update convergence parameter tp using Eq. (13)
        if (| tp |>= 1) then                                   ▷ Exploration phase
            Update the position of the current search agent using Eq. (16)
        else if (| cp |< 1) then                               ▷ Exploitation phase
            Update the position of the current search agent by Eq. (12)
        end if
    end for
    end for
    Update convergence parameter a using a selected formula.
    update A, and C
    Evaluate the fitness of new population
    Update the best three solutions X_α , X_β and X_δ
    t = t + 1
end while
end while
return X_α
```



(a) S-shaped (b) V-shaped

Figure 6: S-shaped and V-shaped TFs

where t represents the current iteration, X_j is the real value of the j^{th} element, $T(X_j)$ is the probability that X_j will be 1 or 0, and r is a random value inside [0,1].

$$T(X_j(t)) = |\tanh(X_j(t))| \tag{23}$$

$$X_j(t+1) = \begin{cases} \backsim b_j & r < T(X_j(t)) \\ b_j & Otherwise \end{cases} \tag{24}$$

where $\backsim$ is the complement, b_j is the current binary value for the j^{th} element. With the complement binarization rule, the new binary value $(X_j(t+1)$ is set based on the benefits of the current binary solution, that is to say, based on the probability value $(T(X_j(t))$, the j^{th} element is either kept or flipped.

4.3 Formulation of Feature Selection Using BGWO

Adapting optimization algorithms to deal with any optimization problem requires identifying two main components: evaluation (fitness) function and solution representation. The main objective of the FS task is to find the minimal features subset that aids in achieving the maximum classification accuracy. Therefore, FS is recognized as a challenging multi-objective optimization problem. Aggregation is one of the most popular techniques for multi-objective formulation. It is a prior method in which multiple objectives are combined into a single objective such each objective is given a weight to define its importance [Mirjalili and Dong, 2020]. Accordingly, the two objectives of FS are combined as shown in 25 to evaluate the suitability of the features subset.

$$\downarrow Fitness(X) = \alpha \times (1 - \gamma(X)) + \beta \times \frac{R}{D} \tag{25}$$

where $Fitness(X)$ represents the fitness value of a subset X, $\gamma(X)$ denotes the classification accuracy by filtering out the unselected features in the X subset, R and D are the number of selected features and the number of original features in the dataset respectively, α and β are the weights of the classification accuracy and the reduction ratio, $\alpha \in [0,1]$ and $\beta = (1-\alpha)$ adopted from [Emary and Zawbaa, 2016, Mafarja et al., 2017, Faris et al., 2018b, Thaher et al., 2021b].

Solution representation for the problem being solved is the other main design aspect that should be properly determined. FS is considered a binary optimization problem in which the candidate solution (i.e., a subset of features) is encoded as a binary vector. Each element in that vector has two values: (0) indicates that the corresponding feature is not selected, and (1) indicates that the corresponding feature is selected. Fig. 7 depicts a sample solution for a dataset of D features.

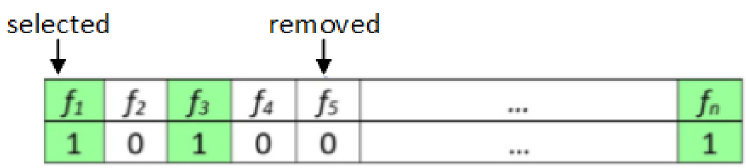


Figure 7: Binary solution representation

5 Experiments and Discussion

This part introduces the experimental work carried out to test the performance of the proposed algorithm. It is divided into four parts. Section 5.1 presents the experimental setup. Section 5.3 provides the implementation of basic GWO with different binarization schemes. After selecting the suitable binary variant, we investigate the performance of the proposed approaches in Section 5.4. Section 5.5 is dedicated to providing a comparison between the enhanced GWO and six state-of-the-art algorithms.

5.1 Experimental Environment Setup

For the sake of fair comparisons, all experiments were performed under the same environment utilizing a system with Intel(R) Core(TM) i7-8550U CPU @ 1.8GHz (8 CPUs) and 8 GB RAM. Al algorithms were implemented in MATLAB R2018a. Due to the non-deterministic behavior of metaheuristic algorithms, each algorithm is executed 20 times, and the results are reported in terms of average (Avg) and standard deviation (Std). The best-reported results are highlighted in **boldface**. Moreover, the non-parametric Friedman test (F-test) [Riffenburgh, 2006] was utilized to calculate the overall rank of each tested algorithm. The detailed settings of all algorithms including common and internal parameters are listed in Table 2. Values of parameters were selected based on trials and errors on small simulations as well as recommended settings in the literature.

The performance of the proposed wrapper methods was evaluated on ten biological benchmark datasets obtained from [Li et al., 2018].The main characteristics of these datasets, including the number of features, number of records, and number of classes, are provided in Table 3. for reliable performance estimation of machine learning (ML) algorithms, we applied the k-folds cross-validation(CV) procedure (k=5). It is a resampling procedure used to assess the performance of ML algorithms. This method is recommended when the data is limited, as in the used biological data where the number of samples ranges between 50-203 (see Table 3). Moreover, CV yields less optimistic (or less biased) results compared to the simple hold-out method [Hastie et al., 2009, James et al., 2013, Tumar et al., 2020]. In this work, the data sample is partitioned randomly into five folds; four folds (80%) are used to train the model, while the remaining fold (20%) is used to validate the model. This procedure is repeated five times, thus ensures that each sample has the chance of being appeared in the training and testing set. Finally, the recorded scores are averaged to represent the performance metric of the model.

To judge the performance of the proposed wrapper-based FS methods, three evaluation measures were used: classification accuracy, number of selected features, fitness values, and running time.

5.2 Evaluation of Classification Methods

It is well-known that the involved classification algorithm highly influences the performance of wrapper FS methods. Besides, each classifier is sensitive to its specific parameters. Accordingly, extensive experiments were conducted to select a suitable classifier to evaluate the proposed searching algorithm. For this purpose, we exploited two well-known classifiers: K-Nearest Neighbors (KNN) and Decision Tree (DT), which are commonly utilized in the literature for FS problems [Tumar et al., 2020, Hassouneh et al., 2021, Mafarja and Mirjalili, 2017]. KNN was tested with different K values, while DT was tested with a different maximum number of splits (MS).

Common parameters		
population size		10
Number of iterations		100
Number of runs		20
Dimension		#features
K for cross validation		5
Fitness function		α=0.9 , β=0.1
Internal parameters		
Algorithm	parameter	value
BHHO	Convergence constant E	[2 0]
BWOA	convergence constant a	[2 0]
	Spiral factor b	1
BBA	Qmin, Qmax	0, 2
	loudness A	0.5
	Pulse rate r	0.5
BGWO	convergence constant a	[2 0]
BPSO	Inertai weight w	[0.9 0.2]
	cognitive constant c1	2
	social constant c2	2
BGSA	initial gravitational constant G0	10
	Rpower	1
	Alpha	20
BALO	w	[2,6]

Table 2: Common and internal settings of tested algorithms

Dataset	No. of features	No. of instances	No. of classes
ALLAML_R	7129	72	2
CLL_SUB_111_R	11340	111	3
colon_R	2000	62	2
GLI_85_R	22283	85	2
GLIOMA_R	4434	50	4
lung_discrete_R	325	73	7
lung_R	3312	203	5
lymphoma_R	4026	96	9
Prostate_GE_R	5966	102	2
SMK_CAN_187_R	19993	187	2

Table 3: List of biological datasets

Dataset	KNN						DT				
	k=3	k=5	k=10	k=20	k=30	k=50	MS=3	MS=7	MS=10	MS=20	MS=50
ALLAML_R	0.8333	0.7917	0.7500	0.6806	0.6528	0.6528	**0.8750**	**0.8750**	**0.8750**	**0.8750**	**0.8750**
CLL_SUB_111_R	0.5045	0.4775	0.5315	0.4685	0.3874	0.3423	**0.7027**	0.6036	0.6036	0.6036	0.6036
colon_R	0.7258	**0.7581**	0.7097	0.6613	0.6452	0.6452	0.7097	0.7419	0.7419	0.7419	0.7419
GLI_85_R	0.7765	0.8235	**0.8824**	0.8353	0.7294	0.6941	0.8000	0.8000	0.8000	0.8000	0.8000
GLIOMA_R	0.7800	**0.8200**	0.7600	0.5000	0.5000	0.2600	0.7000	0.6800	0.6800	0.6800	0.6800
lung_discrete_R	**0.8767**	0.8356	0.7945	0.5890	0.5343	0.4658	0.5480	0.6301	0.6164	0.6164	0.6164
lung_R	0.9507	**0.9606**	0.9310	0.9212	0.7833	0.6847	0.8030	0.8227	0.8227	0.8227	0.8227
lymphoma_R	**0.9375**	0.9167	0.8229	0.7188	0.5833	0.4792	0.5833	0.6042	0.6042	0.6042	0.6042
Prostate_GE_R	0.7941	**0.8333**	0.8235	0.7549	0.7549	0.6373	**0.8333**	0.8137	0.8137	0.8137	0.8137
SMK_CAN_187_R	0.6471	0.6791	0.6631	**0.6952**	0.6898	0.6471	0.5989	0.5401	0.5615	0.5615	0.5615
Rank (F-test)	4.75	**2.95**	4.15	6.50	**9.15**	10.35	5.75	5.60	5.60	5.60	5.60

Table 4: Performance of KNN and DT on the original datasets in terms of accuracy measure

Table 4 reports the accuracy rates scored by KNN and DT on the original biological data (i.e., with all features). Inspecting the results, KNN (k=5) achieved better performance in most cases with the best overall rank of 2.95. The reported results confirm the sensitivity of KNN and DT to their parameter settings. Based on the results obtained, the KNN classifier is utilized to evaluate the generated subset of features in the wrapper FS approach.

5.3 Evaluation of BGWO Using Different Binarization Schemes

The literature reveals that the adopted binarization scheme significantly affects the performance of the binary GWO (BGWO) in dealing with the FS problem. In this subsection, different binarization strategies were investigated to recognize the top version of the proposed BGWO. For this purpose, the S-shaped-based strategies presented by [Zawbaa et al., 2016] were adopted to develop two variants, namely, SBGWO1 and SBGWO2. In addition, we employed V-shaped TF with the proposed strategies to introduce other variants called VBGWO1 and VBGWO2. The description of the four binary variants are summarized as follows:

– SBGWO1 and VBGWO1: Individual steps toward the best three leaders are binarized using S-shaped (SBGWO1) and V-shaped (VBGWO1) TFs, respectively. Then stochastic crossover is performed among the three basic moves to find the updated binary gray wolf position [Zawbaa et al., 2016].

– SBGWO2 and VBGWO2: The S-shaped in SBGWO2 and V-shaped in VBGWO2 were utilized to squash the continuously updated position, then stochastically threshold these values to find the updated binary gray wolf position.

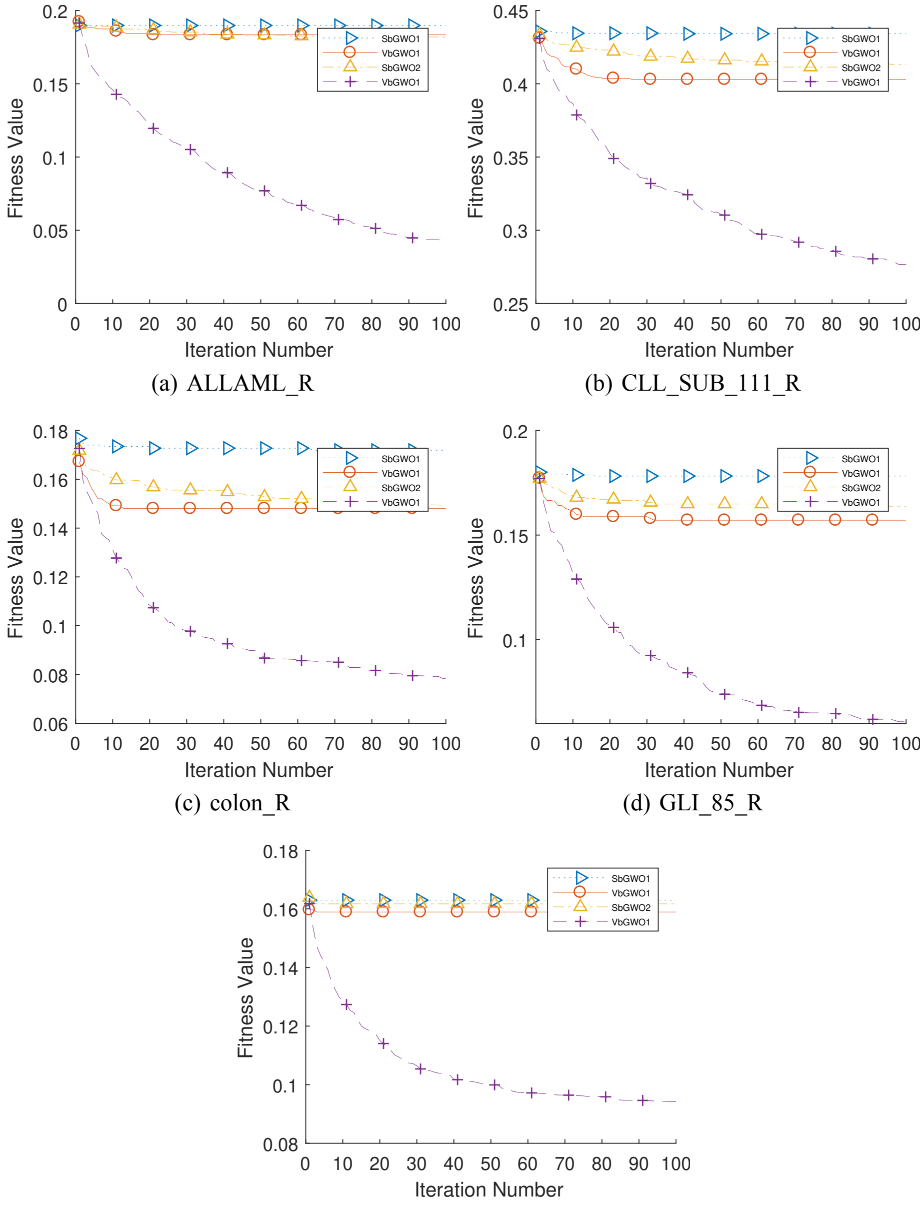
The comparative results of the developed four variants of BGWO are exposed in Tables 5, 6, and 7.The average accuracy rates in Table 5 outlines that the combination between the second binarization scheme and V-shaped TF (VBGWO2) is superior in all datasets. It achieved the first rank of 1.00 followed by VBGWO1, SBGWO2, SBGWO1, respectively. Considering the employed binarization schemes, the efficiency of the two-step binarization method (i.e., VBGWO1 and VBGWO2) is evident as they got the first and second ranks, respectively. In this strategy, the original real operators of GWO are not modified. TFs are utilized to map the continuous solution into binary. Moreover, V-shaped TF confirms clear superiority over S-shaped TF.

Dataset	Measure	SBGWO1	VBGWO1	SBGWO2	VBGWO2
ALLAML_R	Avg	0.8444	0.8625	0.8660	**0.9562**
	Std	0.0110	0.0079	0.0093	0.0151
CLL_SUB_111_R	Avg	0.5838	0.6243	0.6117	**0.6991**
	Std	0.0163	0.0121	0.0065	0.0229
colon_R	Avg	0.8823	0.9065	0.9048	**0.9210**
	Std	0.0153	0.0127	0.0050	0.0103
GLI_85_R	Avg	0.8682	0.8965	0.8859	**0.9400**
	Std	0.0155	0.0093	0.0077	0.0084
GLIOMA_R	Avg	0.8740	0.8800	0.8800	**0.9000**
	Std	0.0097	0.0000	0.0000	0.0000
lung_discrete_R	Avg	0.8630	0.8836	0.8747	**0.9158**
	Std	0.0112	0.0097	0.0080	0.0195
lung_R	Avg	0.9557	0.9606	0.9564	**0.9776**
	Std	0.0040	0.0040	0.0049	0.0030
lymphoma_R	Avg	0.9271	0.9375	0.9339	**0.9547**
	Std	0.0098	0.0000	0.0051	0.0051
Prostate_GE_R	Avg	0.8667	0.8931	0.8917	**0.9417**
	Std	0.0105	0.0086	0.0067	0.0087
SMK_CAN_187_R	Avg	0.6497	0.6711	0.6674	**0.7281**
	Std	0.0081	0.0068	0.0071	0.0102
Rank	F-test	4.00	2.15	2.85	**1.00**

Table 5: Comparison of BGWO using different binarization schemes in terms of accuracy rates

The resultant average number of selected features in Table 6 illustrates that the VBGWO2 gives a remarkable ranking over the other proposed binarizations approaches by filtering out a large number of irrelevant features. For Table 7 of the average fitness values, VBGWO2 also demonstrates its efficiency in providing the best results in all cases. Finally, the convergence behaviors for BGWO variants (SBGWO1, VBGWO1, SBGWO2, and VBGWO2) in dealing with all datasets are demonstrated in Figures 8 and 9 . According to curves trends, it is noted that VBGWO2 exhibits a quick and efficient convergence behavior for all datasets. The Other variants show a premature convergence drawback. Thus, V-shaped TF proves satisfactory results by keeping the required diversity during the search process.

Based on the previous observations, it is recognized that the performance of the two-step binarization scheme combined with V-shaped TF is efficient in terms of the average accuracy, average numbers of features, and convergence trends. In this regard, VBGWO2 is proven to be effective by reducing the number of features and perceiving higher accuracy rates which are the main ain of the FS process. Accordingly, this binarization scheme is considered in all subsequent experiments. For simplicity, the abbreviation BGWO will be used to denote the binary variant of GWO.



(a) ALLAML_R

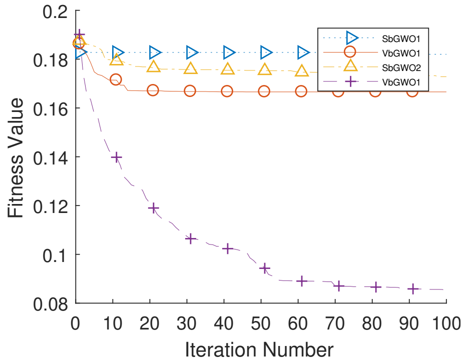
(b) CLL_SUB_111_R

(c) colon_R

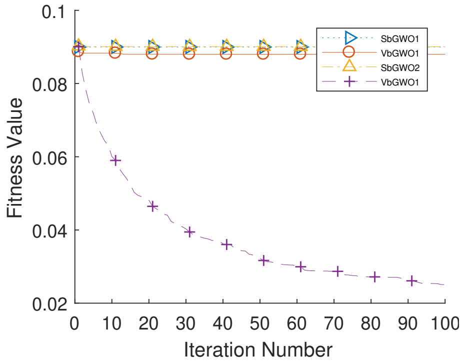
(d) GLI_85_R

(e) GLIOMA_R

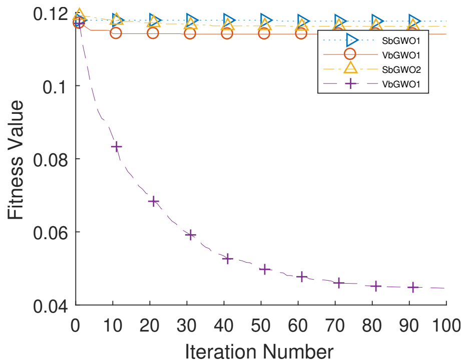
Figure 8: Convergence curves of BGWO variants for ALLAML_R, CLL_SUB_111_R, colon_R, GLI_85_R, and GLIOMA_R datasets



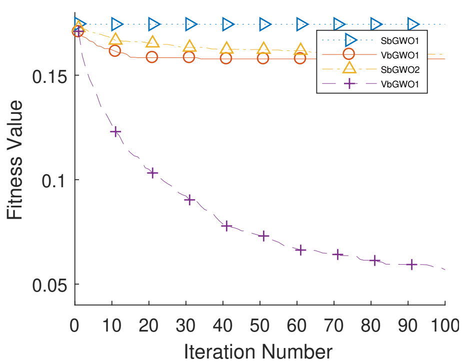
(a) lung_discrete_R



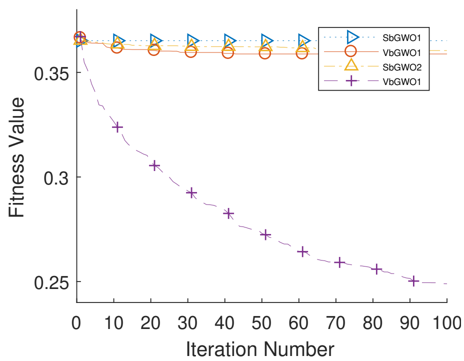
(b) lung_R



(c) lymphoma_R



(d) Prostate_GE_R



(e) SMK_CAN_187_R

Figure 9: Convergence curves of BGWO variants for lung_discrete_R, lung_R, lymphoma_R, Prostate_GE_R, SMK_CAN_187_R datasets

Dataset	Measure	SBGWO1	VBGWO1	SBGWO2	VBGWO2
ALLAML_R	Avg	3549.90	4254.20	4370.60	**252.05**
	Std	46.16	399.12	365.77	52.57
CLL_SUB_111_R	Avg	6752.80	7353.40	7225.20	**663.30**
	Std	1794.40	367.88	380.45	218.26
colon_R	Avg	1318.20	1275.90	1275.80	**145.95**
	Std	357.47	47.03	22.97	49.36
GLI_85_R	Avg	13279.60	14239.50	13557.25	**1501.10**
	Std	3575.17	768.61	1278.66	597.13
GLIOMA_R	Avg	2199.00	2258.80	2382.30	**185.90**
	Std	25.46	150.96	252.50	65.53
lung_discrete_R	Avg	190.80	200.80	195.00	**29.05**
	Std	49.68	6.21	17.92	6.87
lung_R	Avg	1659.00	1739.90	1683.30	**162.65**
	Std	24.82	133.33	123.98	57.83
lymphoma_R	Avg	2095.50	2329.70	2282.25	**152.85**
	Std	325.17	120.37	264.77	48.45
Prostate_GE_R	Avg	3245.00	3676.80	3731.35	**260.75**
	Std	637.19	297.09	255.25	53.52
SMK_CAN_187_R	Avg	9979.90	12558.80	12212.65	**839.65**
	Std	53.29	386.50	1149.59	185.39
Mean Rank	F-test	2.20	3.60	3.20	**1.00**

Table 6: Comparison of BGWO using different binarization schemes in terms of selected features.

5.4 Results of the Enhanced Variants

After identifying the best binary variant of GWO, which is the V-shaped transfer function based variant VBGWO2, in this section, we examine the efficiency of enhanced versions of the VBGWO2 algorithm where different decreasing strategies of parameter a (i.e., EGWO1, EGWO2, EGWO3, and EGWO4). The proposed binary variants are named BEGWO, BEGWO1, BEGWO2, BEGWO3, and BEGWO4, respectively. The efficiency is appraised on the considered datasets by inspecting the average classification accuracy, size of the selected features, average fitness, and convergence trends.

Table 8 reports the comparative results of the proposed variants with the original BGWO in terms of accuracy rates. From Table 8, it is noted that the BEGWO4 achieved a better accuracy score in 60% of the datasets. For instance, compared to the original BGWO, the BEGWO4 achieved an increment of roughly 1.5%, 6%, and 4% accuracy for datasets ALLAML_R, CLL_SUB_111_R, and SMK_CAN_187_R, respectively. In contrast, it can be observed that the BEGWO2 and BEGWO3 are the last preference based on the accuracy results. As per F-test, BEGWO4 is ranked first, followed by BEGWO, BGWO, BEGWO1, BEGWO3, and BEGWO2.

Based on the size of selected features in Table 9, BEGWO4 has successively perceived the best ranking of 1.20 by reducing a large number of unwanted features. By assaying the reported results, it can be observed that the BEGWO4 declares superior results in 80 % of the datasets. Considering the CLL_SUB_111_R dataset as an example, the

Dataset	Measure	SBGWO1	VBGWO1	SBGWO2	VBGWO2
ALLAML_R	Avg	0.1898	0.1834	0.1819	**0.0429**
	Std	0.0094	0.0071	0.0059	0.0139
CLL_SUB_111_R	Avg	0.4341	0.4030	0.4132	**0.2767**
	Std	0.0038	0.0103	0.0056	0.0219
colon_R	Avg	0.1719	0.1480	0.1494	**0.0784**
	Std	0.0139	0.0111	0.0041	0.0092
GLI_85_R	Avg	0.1782	0.1571	0.1635	**0.0607**
	Std	0.0122	0.0108	0.0056	0.0096
GLIOMA_R	Avg	0.1630	0.1589	0.1617	**0.0942**
	Std	0.0087	0.0034	0.0057	0.0015
lung_discrete_R	Avg	0.1820	0.1666	0.1728	**0.0848**
	Std	0.0081	0.0085	0.0046	0.0173
lung_R	Avg	0.0900	0.0880	0.0901	**0.0251**
	Std	0.0036	0.0025	0.0035	0.0030
lymphoma_R	Avg	0.1177	0.1141	0.1162	**0.0446**
	Std	0.0094	0.0030	0.0039	0.0041
Prostate_GE_R	Avg	0.1744	0.1578	0.1600	**0.0569**
	Std	0.0070	0.0095	0.0039	0.0079
SMK_CAN_187_R	Avg	0.3652	0.3588	0.3604	**0.2489**
	Std	0.0072	0.0060	0.0046	0.0093
Mean Rank	F-test	3.90	2.10	3.00	**1.00**

Table 7: Comparison of BGWO using different binarization methods in terms of fitness values.

BEGWO4 scored an accuracy of 0.7626 with a minimal subset of features (size of 69.80). In comparison, the conventional BGWO scored an accuracy of 0.699 with 663.3 features. In general, there is a clear reduction in the number of features while improving or at least maintaining classification accuracy.

The excellent performance of the BEGWO4 is confirmed by the average of fitness values reported in Table 10. Inspecting the results, BEGWO4 can surpass the other competitors on 80% of the utilized datasets. Accordingly, this variant outperforms the other competitive variants in satisfying the FS process's main aim. As per mean rank, BEGWO4 is ranked first, followed by BEGWO and BGWO, respectively.

Finally, the convergence behaviors for the top three variants (i.e., BEGWO4, BEGWO, and BGWO) are shown in Figures 10 and 11. The plotting curves illustrate that the BEGWO4 variant exhibits a better acceleration rate of convergence in about seven problems than all other variants. The convergence curves reveal an acceptable performance of the standard BGWO variant in three problems: lung_discrete_R, lung_R, and lymphoma_R.

Justifying the superiority of the proposed BEGWO4, we can say that employing the transition parameter along with a random-based exploration operator guarantees good exploratory potential at the beginning of the search process and emphasize the required diversity for high-dimensional datasets. Therefore, it can find better minimal subsets of features that contribute to the highest accuracy rate. In specific, delaying

Dataset	Measure	BGWO	BEGWO	BEGWO1	BEGWO2	BEGWO3	BEGWO4
ALLAML_R	Avg	0.9562	0.9667	0.9396	0.9306	0.9368	**0.9708**
	Std	0.0151	0.0171	0.0151	0.0142	0.0171	0.0118
CLL_SUB_111_R	Avg	0.6991	0.7518	0.6838	0.6730	0.7203	**0.7626**
	Std	0.0229	0.0307	0.0244	0.0176	0.0374	0.0268
colon_R	Avg	0.9210	0.9250	0.9242	0.9226	0.9202	**0.9282**
	Std	0.0103	0.0168	0.0076	0.0099	0.0111	0.0111
GLI_85_R	Avg	0.9400	0.9406	0.9359	0.9312	0.9365	**0.9494**
	Std	0.0084	0.0145	0.0160	0.0096	0.0134	0.0179
GLIOMA_R	Avg	**0.9000**	**0.9000**	**0.9000**	**0.9000**	**0.9000**	**0.9000**
	Std	0.0000	0.0000	0.0065	0.0000	0.0000	0.0000
lung_discrete_R	Avg	**0.9158**	0.9082	0.9151	0.9096	0.9130	0.9082
	Std	0.0195	0.0199	0.0123	0.0136	0.0135	0.0127
lung_R	Avg	**0.9776**	0.9741	0.9729	0.9724	0.9700	0.9702
	Std	0.0030	0.0045	0.0034	0.0046	0.0040	0.0052
lymphoma_R	Avg	0.9547	0.9526	**0.9552**	0.9526	0.9402	0.9479
	Std	0.0051	0.0053	0.0049	0.0053	0.0061	0.0089
Prostate_GE_R	Avg	0.9417	0.9426	0.9343	0.9275	0.9333	**0.9471**
	Std	0.0087	0.0160	0.0079	0.0098	0.0060	0.0116
SMK_CAN_187_R	Avg	0.7281	0.7543	0.7131	0.7131	0.7297	**0.7709**
	Std	0.0102	0.0174	0.0117	0.0078	0.0122	0.0177
Mean Rank	F-test	2.95	2.65	3.60	4.85	4.45	**2.50**

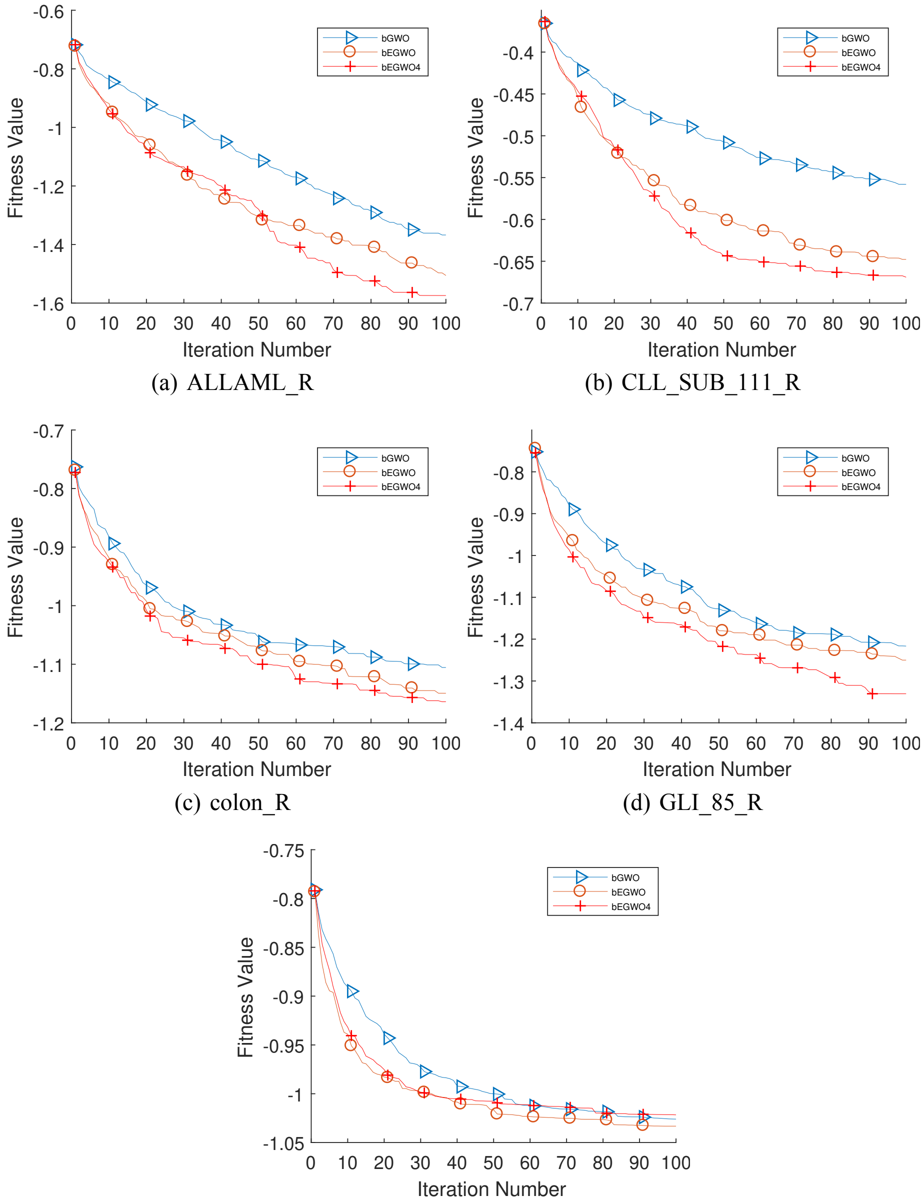
Table 8: Comparison between the enhanced variants and the standard BGWO in terms of accuracy rates

the rate of decrease in the value of the control parameter (a) leads to the possibility of applying the exploration operators for more iterations during the search process, thus improving the algorithm's ability to explore more promising regions in the large search space. Accordingly, it preserves the required diversity during the search process and reduces the chances of being trapped in local optima.

5.5 Comparison of BEGWO4 with Well-known Algorithms

After proving the effectiveness of the proposed BEGWO4 compared to the conventional BGWO. This part aims to validate the performance of BEGWO4 by conducting a deep comparison with other six well-established algorithms in terms of average accuracy, size of the selected features, and fitness values as reported in Tables 11, 12, and 13, respectively. The comparative algorithms are Binary Whale Optimization Algorithm (BWOA), Binary Particle Swarm Optimization (BPSO), Binary Bat Algorithm (BBA), Binary Ant Lion Optimizer (BALO), Binary Harris Hawks Optimizer (BHHO), and Binary Gravitational Search Algorithm (BGSA). To keep a fair comparison, the same binarization scheme was used to adapt these algorithms for the FS problem.

Table 11 outlines the averages of classification accuracy obtained by the BEGWO4 and other algorithms. The results demonstrate that the BEGWO4 variant outperforms the other competitors in achieving high average accuracy ON 70% of the utilized datasets (overall rank of 1.3). For the datasets containing the largest number of features, such as GLI_85_R and SMK_CAN_187_R, the BEGWO4 has retained the highest accuracy

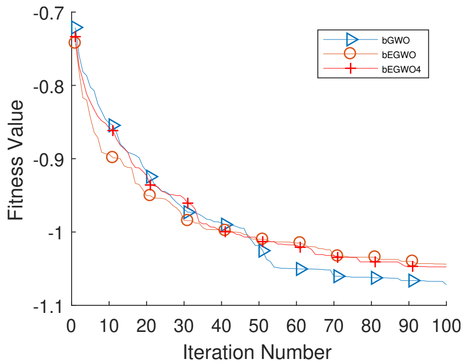


(a) ALLAML_R (b) CLL_SUB_111_R

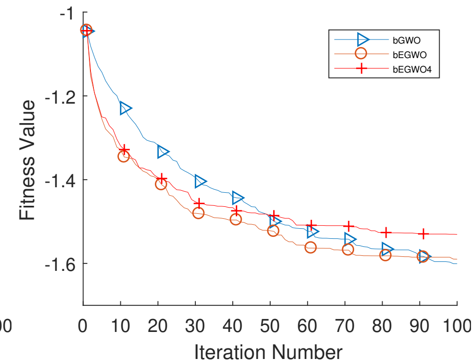
(c) colon_R (d) GLI_85_R

(e) GLIOMA_R

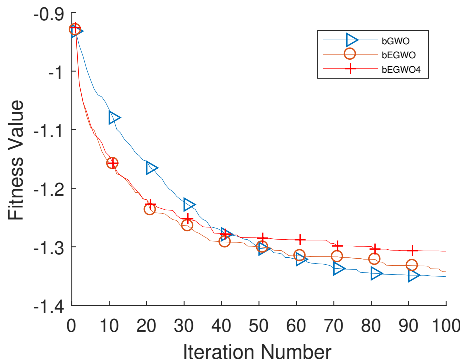
Figure 10: Convergence curves for top three variants bGWO, bEGWO, and bEGWO4 for ALLAML_R, CLL_SUB_111_R, colon_R, GLI_85_R, and GLIOMA_R datasets [log scale]



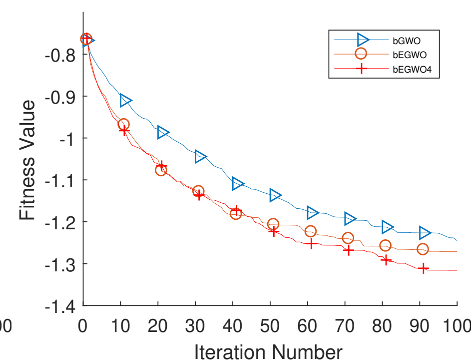
(a) lung_discrete_R



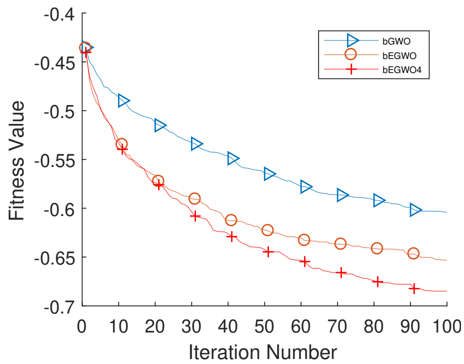
(b) lung_R



(c) lymphoma_R



(d) Prostate_GE_R



(e) SMK_CAN_187_R

Figure 11: Convergence curves for top three variants bGWO, bEGWO, and bEGWO4 for lung_discrete_R, lung_R, lymphoma_R, Prostate_GE_R, SMK_CAN_187_R datasets [log scale]

Dataset	Measure	BGWO	BEGWO	BEGWO1	BEGWO2	BEGWO3	BEGWO4
ALLAML_R	Avg	252.05	86.10	527.45	606.15	274.25	**29.05**
	Std	52.57	37.52	128.66	172.22	103.35	22.74
CLL_SUB_111_R	Avg	663.30	186.60	1002.65	1386.80	527.60	**69.80**
	Std	218.26	121.88	265.72	324.54	266.68	33.22
colon_R	Avg	145.95	**67.60**	201.55	250.85	170.50	78.95
	Std	49.36	44.76	28.25	99.52	50.30	64.43
GLI_85_R	Avg	1501.10	612.85	2331.15	2426.40	1205.90	**262.70**
	Std	597.13	570.46	1002.27	493.23	486.68	140.89
GLIOMA_R	Avg	185.90	117.15	253.30	318.05	221.05	**108.05**
	Std	65.53	49.70	73.71	77.74	65.68	48.88
lung_discrete_R	Avg	29.05	25.30	41.75	45.60	33.20	**22.85**
	Std	6.87	9.70	8.97	14.86	8.40	6.92
lung_R	Avg	162.65	**80.70**	227.60	292.20	170.25	87.30
	Std	57.83	29.13	48.98	60.83	37.05	43.31
lymphoma_R	Avg	152.85	112.05	251.15	314.20	220.65	**96.40**
	Std	48.45	38.01	53.22	76.98	68.87	61.17
Prostate_GE_R	Avg	260.75	113.95	426.80	621.85	285.15	**39.80**
	Std	53.52	74.31	88.51	173.48	83.37	34.63
SMK_CAN_187_R	Avg	839.65	226.50	1617.55	1784.00	636.40	**70.20**
	Std	185.39	127.06	505.24	437.40	316.66	34.14
Mean Rank	F-test	3.30	1.80	5.00	**6.00**	3.70	1.20

Table 9: Comparison between the enhanced variants and the standard BGWO in terms of selected fetaures

94.94% and 77.09%, which proved a better learning and classification process. The BHHO scored higher accuracy results compared to the other competitors in 3 cases (overall rank of 1.7). According to the overall rank, the BEGWO4 came in the first rank, followed by BHHO, BWOA, BPSO, BGSA, BALO, and BBA, respectively.

By inspecting the average number of selected features in Table 12. It is clear the superiority of BWOA in offering the minimum number of features on about 90 % of the datasets (rank of 1.2), followed by BEGWO4 (rank of 2.10). These findings imply that BWOA shows success in feature reduction. However, excessive feature reduction may result in the exclusion of some relevant features, which degrade the classification performance (see Table 11). Although BEGWO4 is given the second rank in terms of feature reduction, it can find the most relevant subset of features that provides better classification accuracy.

To confirm the effectiveness of the competing algorithms, the fitness value which combines the two measures (i.e., accuracy and reduction rate) is adopted. The results in Table 13 confirm the superiority of BEGWO4 in offering better fitness values on 70% of the datasets. Based on the overall rank, BEGWO4 is ranked first, followed by BHHO which outperforms the other peers in 3 cases.

Figures 12 and 13 examine the acceleration behavior of the developed BEGWO4 versus other approaches. Considering all curves, it can be observed that the BEGWO4 accelerates faster than other competitors toward better solutions in the majority of problems (6 cases). It is also competitive to BHHO in CLL_SUB_111_R, lymphoma_R, and Prostate_GE_R datasets. BHHO shows the second-best convergence rate. In contrast,

Dataset	Measure	BGWO	BEGWO	BEGWO1	BEGWO2	BEGWO3	BEGWO4
ALLAML_R	Avg	0.0429	0.0312	0.0618	0.0710	0.0607	**0.0267**
	Std	0.0139	0.0157	0.0144	0.0134	0.0156	0.0108
CLL_SUB_111_R	Avg	0.2767	0.2250	0.2934	0.3066	0.2564	**0.2143**
	Std	0.0219	0.0282	0.0226	0.0170	0.0355	0.0242
colon_R	Avg	0.0784	0.0709	0.0783	0.0822	0.0804	**0.0685**
	Std	0.0092	0.0153	0.0071	0.0101	0.0098	0.0116
GLI_85_R	Avg	0.0607	0.0562	0.0682	0.0728	0.0626	**0.0467**
	Std	0.0096	0.0141	0.0145	0.0091	0.0135	0.0162
GLIOMA_R	Avg	0.0942	0.0926	0.0957	0.0972	0.0950	**0.0924**
	Std	0.0015	0.0011	0.0058	0.0018	0.0015	0.0066
lung_discrete_R	Avg	**0.0848**	0.0904	0.0893	0.0954	0.0885	0.0896
	Std	0.0173	0.0195	0.0105	0.0131	0.0129	0.0111
lung_R	Avg	**0.0251**	0.0257	0.0313	0.0337	0.0321	0.0295
	Std	0.0030	0.0042	0.0027	0.0047	0.0038	0.0046
lymphoma_R	Avg	**0.0446**	0.0454	0.0466	0.0505	0.0593	0.0493
	Std	0.0041	0.0049	0.0042	0.0044	0.0063	0.0080
Prostate_GE_R	Avg	0.0569	0.0535	0.0663	0.0757	0.0648	**0.0483**
	Std	0.0079	0.0153	0.0074	0.0097	0.0059	0.0109
SMK_CAN_187_R	Avg	0.2489	0.2223	0.2663	0.2671	0.2465	**0.2066**
	Std	0.0093	0.0160	0.0122	0.0083	0.0118	0.0159
Mean Rank	F-test	2.70	2.30	4.30	5.90	4.00	**1.80**

Table 10: Comparison between the enhanced variants and the standard BGWO in terms of fitness results

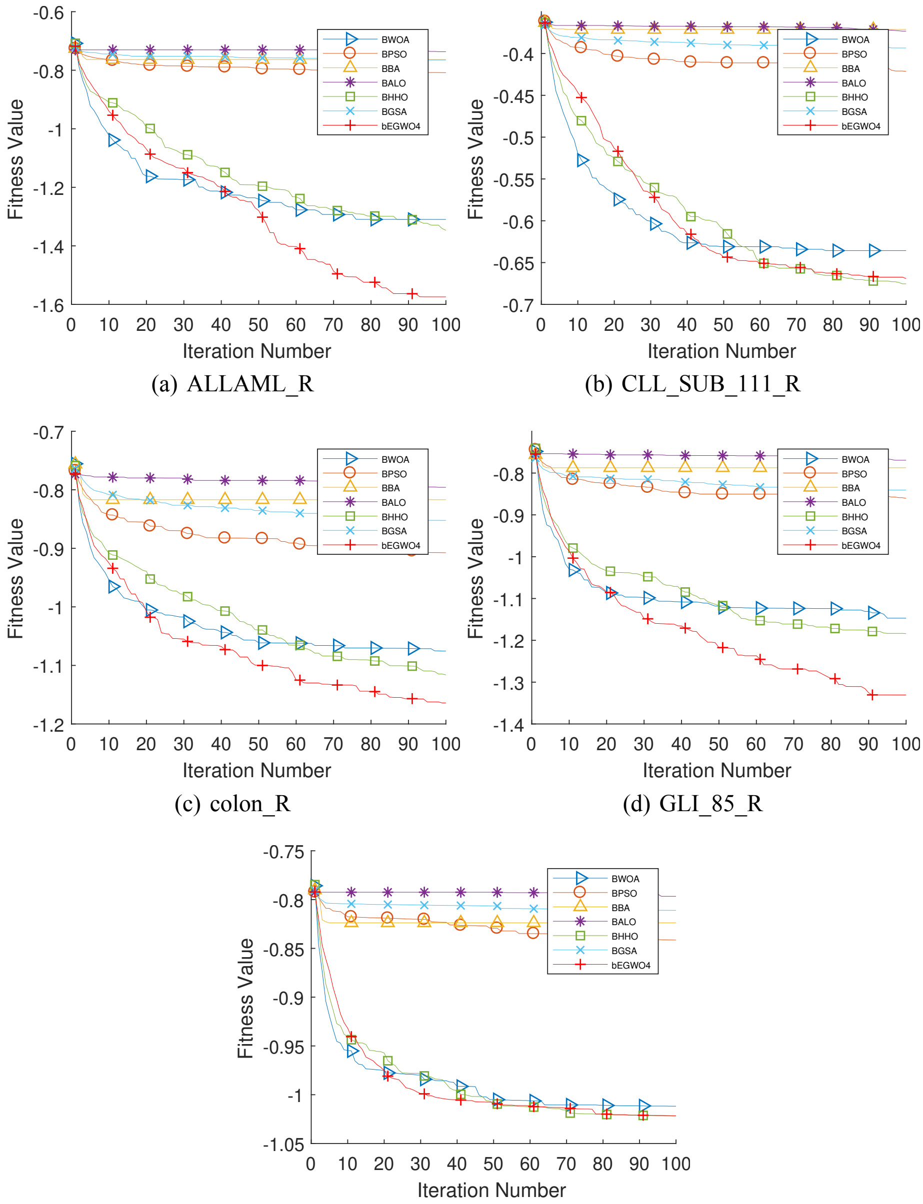
several stagnation problems can be detected in BALO and BBA curves in most cases.

Based on previous observations, it is recognized that BEGWO4 ensures satisfactory results in terms of all measures. The added enhancements to the GWO improve the capability of search agents to explore and exploit the massive search space.

5.6 Evaluation the Impact of Feature Felection

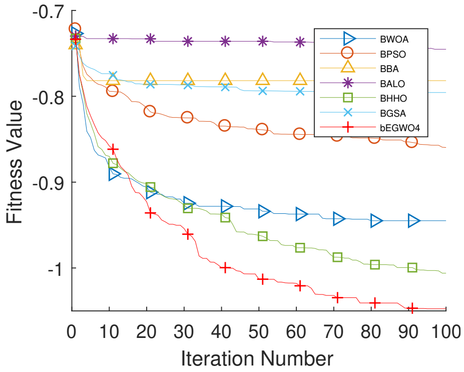
This part aims to verify the positive impact of the FS process on the performance of the classification algorithm. Precisely, we present a deep comparison of KNN performance before and after the application of FS methods. For this purpose, KNN without FS (i.e., using the original feature set) is compared to the proposed wrapper FS approach that combines BEGWO4 and KNN (BEGWO4-KNN) in addition to the Relief-based algorithm (RBA) with KNN (RBA-KNN). The literature reveals that RBA is an efficient filter-based FS method. It has gained appeal by striking a flexibly adapting to various data characteristics with complex patterns of association [Urbanowicz et al., 2018]. In this experiment, several threshold (τ) ratios of 0.01, 0.25, 0.5, and 0.75 were tested. Here τ represents the ratio of selected features that were given the highest weights by RBA.

The number of features and classification accuracy achieved by KNN, RBA-KNN, and BEGWO4 are reported in Table 14. Firstly, Based on the results of the RBA-KNN method, using 25% of the features scores better classification accuracy compared to other ratios in 6 datasets. It is noted that excessive feature reduction (i.e., the ratio of 0.01) and excessive use of features (i.e., the ratio of 0.75) degrade the classification performance

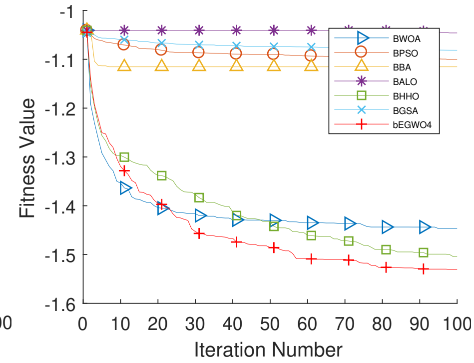


(a) ALLAML_R (b) CLL_SUB_111_R (c) colon_R (d) GLI_85_R (e) GLIOMA_R

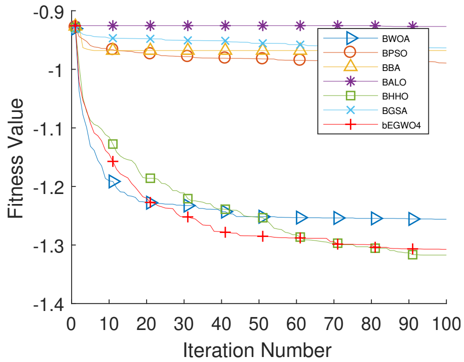
Figure 12: Convergence curves of all compared algorithms for ALLAML_R, CLL_SUB_111_R, colon_R, GLI_85_R, and GLIOMA_R datasets [log scale]



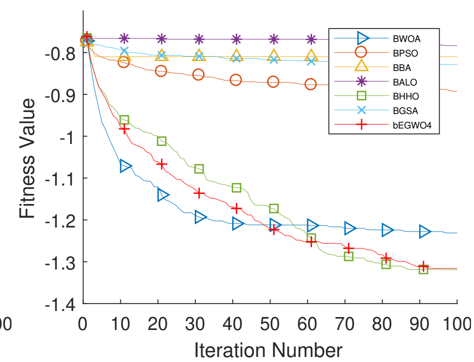
(a) lung_discrete_R



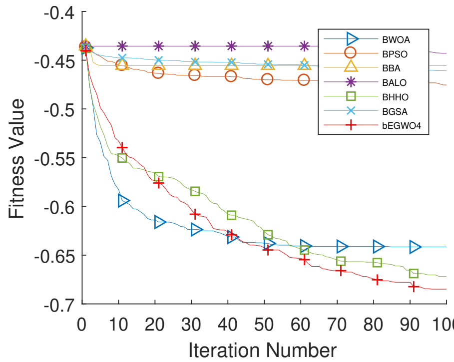
(b) lung_R



(c) lymphoma_R



(d) Prostate_GE_R



(e) SMK_CAN_187_R

Figure 13: Convergence curves of all compared algorithms for lung_discrete_R, lung_R, lymphoma_R, Prostate_GE_R, SMK_CAN_187_R datasets [log scale]

Dataset	Measure	BEGWO4	BWOA	BPSO	BBA	BALO	BHHO	BGSA
ALLAML_R	Avg	**0.9708**	0.9458	0.8813	0.8292	0.8583	0.9507	0.8632
	Std	0.0118	0.0266	0.0084	0.0207	0.0116	0.0204	0.0082
CLL_SUB_111_R	Avg	0.7626	0.7432	0.6338	0.5387	0.6095	**0.7658**	0.6059
	Std	0.0268	0.0383	0.0121	0.0369	0.0161	0.0352	0.0146
colon_R	Avg	**0.9282**	0.9089	0.9153	0.8460	0.9024	0.9194	0.8976
	Std	0.0111	0.0168	0.0089	0.0206	0.0064	0.0148	0.0120
GLI_85_R	Avg	**0.9494**	0.9218	0.9018	0.8329	0.8859	0.9294	0.8947
	Std	0.0179	0.0149	0.0079	0.0287	0.0102	0.0143	0.0081
GLIOMA_R	Avg	**0.9000**	0.8940	0.8920	0.8490	0.8820	0.8980	0.8810
	Std	0.0000	0.0114	0.0101	0.0165	0.0062	0.0062	0.0045
lung_discrete_R	Avg	**0.9082**	0.8815	0.8959	0.8425	0.8712	0.8993	0.8740
	Std	0.0127	0.0260	0.0121	0.0220	0.0093	0.0092	0.0084
lung_R	Avg	**0.9702**	0.9626	0.9650	0.9453	0.9589	0.9687	0.9613
	Std	0.0052	0.0063	0.0042	0.0083	0.0037	0.0051	0.0029
lymphoma_R	Avg	0.9479	0.9406	0.9380	0.9089	0.9266	**0.9505**	0.9323
	Std	0.0089	0.0118	0.0023	0.0106	0.0079	0.0057	0.0053
Prostate_GE_R	Avg	0.9471	0.9353	0.9118	0.8422	0.8868	**0.9475**	0.8892
	Std	0.0116	0.0160	0.0101	0.0165	0.0117	0.0120	0.0096
SMK_CAN_187_R	Avg	**0.7709**	0.7468	0.6834	0.6382	0.6674	0.7636	0.6706
	Std	0.0177	0.0155	0.0064	0.0136	0.0071	0.0147	0.0074
Mean Rank	F-test	**1.30**	3.30	3.70	7.00	5.70	1.70	5.30

Table 11: Average classification accuracy for BEGWO4 and other state-of-the-art algorithms

in most cases compared to the ratio of 0.25. Secondly, considering all methods, the BEGWO-KNN yielded the optimal rank of 1.1; it outperforms the other methods in 90% of cases. RBA-KNN with a ratio of 0.025 is given the second-best rank. In contrast, the KNN with all features yielded the worst rank of 5.30. These observations demonstrate the importance of the FS process in improving the classification performance considerably (see Figure 14). Taking ALLAML_R data as an example, the BEGWO-KNN contributes to an increase of roughly 18% of accuracy with only 29 features out of 7129 features in the original data.

6 Conclusion

This study proposed an efficient wrapper-based FS approach for improving the classification accuracy of high-dimensional biological data. An enhanced variant of the GWO called BEGWO was introduced for exploring the search space to find an optimal subset of features. Different S-Shaped and V-Shaped binarization schemes were applied to transform the continuous search space into a binary one for the FS task. Following the determination of the most effective BGWO binary variant, two enhancements were introduced to emphasize the balance between exploration and exploitation. Firstly, a random-based search operator was employed to ensure better global search ability at the early stage of the optimization process. Secondly, the transition between the exploration

Dataset	Measure	bEGWO4	BWOA	BPSO	BBA	BALO	BHHO	BGSA
ALLAML_R	Avg	29.05	**18.25**	3465.00	2932.80	3981.80	44.30	3467.80
	Std	22.74	25.20	47.26	198.84	390.69	40.66	45.05
CLL_SUB_111_R	Avg	69.80	**33.55**	5596.80	4562.45	8153.80	39.50	5611.30
	Std	33.22	46.21	63.05	565.88	1590.93	39.45	50.02
colon_R	Avg	78.95	**40.55**	950.35	781.95	1444.20	79.15	966.65
	Std	64.43	57.01	14.73	120.22	192.41	83.07	24.71
GLI_85_R	Avg	262.70	**192.60**	11022.35	8778.80	15011.95	431.30	11050.85
	Std	140.89	277.23	73.59	987.89	2127.55	459.66	68.00
GLIOMA_R	Avg	108.05	**85.75**	2077.50	1805.45	2372.90	147.45	2102.25
	Std	48.88	64.73	44.17	130.19	321.18	85.63	28.76
lung_discrete_R	Avg	22.85	**22.35**	144.50	124.20	207.70	26.00	151.70
	Std	6.92	11.52	9.41	15.42	24.62	9.00	11.09
lung_R	Avg	87.30	**68.15**	1583.60	1356.25	1753.60	104.40	1593.65
	Std	43.31	46.44	29.74	101.81	116.39	72.58	32.89
lymphoma_R	Avg	96.40	**81.85**	1880.35	1642.00	2103.50	146.75	1926.45
	Std	61.17	32.36	27.40	103.51	165.92	47.26	34.50
Prostate_GE_R	Avg	39.80	**29.00**	2900.45	2400.85	3747.45	46.20	2908.55
	Std	34.63	45.88	37.08	188.95	540.82	82.67	45.61
SMK_CAN_187_R	Avg	70.20	71.30	9951.25	8058.75	12259.75	**30.20**	9929.20
	Std	34.14	183.83	74.38	813.27	1340.43	18.59	64.58
Mean Rank	F-test	2.10	**1.20**	5.10	4.00	7.00	2.70	5.90

Table 12: Average number of selected features for BEGWO4 and other state-of-the-art algorithms

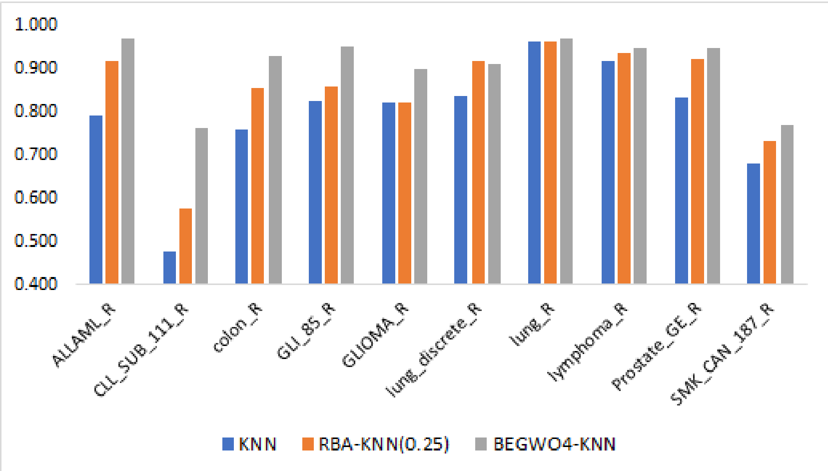


Figure 14: Comparison of BEGWO-KNN, KNN, and RBA-KNN in terms of accuracy rates

Dataset	Measure	bEGWO4	BWOA	BPSO	BBA	BALO	BHHO	BGSA
ALLAML_R	Avg	**0.0267**	0.0490	0.1555	0.1723	0.1834	0.0450	0.1718
	Std	0.0108	0.0240	0.0075	0.0058	0.0093	0.0185	0.0073
CLL_SUB_111_R	Avg	0.2143	0.2314	0.3790	0.4253	0.4234	**0.2112**	0.4042
	Std	0.0242	0.0345	0.0108	0.0176	0.0107	0.0318	0.0132
colon_R	Avg	**0.0685**	0.0840	0.1237	0.1524	0.1600	0.0765	0.1405
	Std	0.0116	0.0150	0.0077	0.0177	0.0114	0.0138	0.0103
GLI_85_R	Avg	**0.0467**	0.0713	0.1379	0.1632	0.1701	0.0655	0.1444
	Std	0.0162	0.0132	0.0071	0.0091	0.0107	0.0123	0.0074
GLIOMA_R	Avg	**0.0924**	0.0973	0.1441	0.1500	0.1597	0.0951	0.1545
	Std	0.0066	0.0103	0.0084	0.0109	0.0091	0.0052	0.0036
lung_discrete_R	Avg	**0.0896**	0.1135	0.1382	0.1653	0.1798	0.0986	0.1601
	Std	0.0111	0.0236	0.0108	0.0141	0.0074	0.0092	0.0077
lung_R	Avg	**0.0295**	0.0358	0.0793	0.0767	0.0900	0.0313	0.0829
	Std	0.0046	0.0053	0.0038	0.0064	0.0026	0.0049	0.0025
lymphoma_R	Avg	0.0493	0.0555	0.1025	0.1076	0.1183	**0.0482**	0.1088
	Std	0.0080	0.0104	0.0019	0.0074	0.0060	0.0054	0.0043
Prostate_GE_R	Avg	0.0483	0.0587	0.1280	0.1549	0.1647	**0.0480**	0.1485
	Std	0.0109	0.0149	0.0092	0.0138	0.0092	0.0115	0.0084
SMK_CAN_187_R	Avg	**0.2066**	0.2282	0.3347	0.3503	0.3607	0.2129	0.3461
	Std	0.0159	0.0143	0.0057	0.0083	0.0065	0.0132	0.0066
Mean Rank	F-test	**1.30**	3.00	4.10	5.70	6.90	1.70	5.30

Table 13: Average fitness values for BEGWO4 and other state-of-the-art algorithms

Benchmark			RBA-KNN									
	KNN		0.01		0.25		0.5		0.75		bEGWO4-KNN	
	Feat	Acc	Feat	Acc	Feat	Acc	Feat	Acc	Feat	Acc	Feat	Acc
ALLAML_R	7129	0.792	71	0.958	1782	0.917	3565	0.861	5347	0.833	29.05	**0.971**
CLL_SUB_111_R	11340	0.477	113	0.514	2835	0.577	5670	0.514	8505	0.505	69.80	**0.763**
colon_R	2000	0.758	20	0.887	500	0.855	1000	0.806	1500	0.806	78.95	**0.928**
GLI_85_R	22283	0.824	223	0.859	5571	0.859	11142	0.835	16712	0.847	262.70	**0.949**
GLIOMA_R	4434	0.820	44	0.840	1109	0.820	2217	0.800	3326	0.820	108.05	**0.900**
lung_discrete_R	325	0.836	3	0.411	81	**0.918**	163	0.904	244	0.849	22.85	0.908
lung_R	3312	0.961	33	0.936	828	0.961	1656	0.961	2484	0.956	87.30	**0.970**
lymphoma_R	4026	0.917	40	0.844	1007	0.938	2013	0.938	3020	0.938	96.40	**0.948**
Prostate_GE_R	5966	0.833	60	0.931	1492	0.922	2983	0.882	4475	0.873	39.80	**0.947**
SMK_CAN_187_R	19993	0.679	200	0.706	4998	0.733	9997	0.706	14995	0.706	70.20	**0.771**
Rank (F-test)	5.30		3.60		2.65		4.00		4.35		**1.10**	

Table 14: Comparison of wrapper FS (BEGWO-KNN) with filter method (RBA-KNN) in terms of accuracy and feature size (Feat).

and exploitation phases was adjusted by embedding the concept of transition parameters using various non-linear decreasing strategies. The proposed approaches were validated on a set of ten challenging biological datasets. It is found that the proposed improved version BEGWO4 outperformed the other three enhanced variants as well as the conventional binary version of the GWO optimizer. In addition, the BEGWO4 FS approach has also shown better performance, especially in terms of classification accuracy, than several optimization algorithms, including BWOA, BPSO, BBA, BALO, BHHO, and BGSA. It can be concluded that the proposed BEGWO-based wrapper FS can satisfy biological classification tasks with a minimal subset of features and better accuracy rates.

Although the proposed GWO-based FS approach has shown promising performance, we still need to assess its performance in other domains such as text classification and micro-array data classification to emphasize its robustness. The evaluation of the proposed approach using further datasets would offer further insight for the efficacy of the proposed approach. Our future works will focus on evaluating the proposed improved BEGWO4 algorithm on other real classification problems. In addition, the same parameter (a) that is used for controlling the transition of the search process from exploration to exploration is also used in other algorithms such as WOA and HHO. One of the future work directions is to apply the same proposed decreasing strategy of the value of (a) parameter to the same parameter in WOA and HHO algorithms to investigate their performance in the FS domain.

Acknowledgements

The authors would like to acknowledge Taif University Researchers Supporting Project Number (TURSP-2020/125), Taif University, Taif, Saudi Arabia.

References

[Abdel-Basset et al., 2020] Abdel-Basset, M., El-Shahat, D., El-henawy, I., de Albuquerque, V., and Mirjalili, S. (2020). A new fusion of grey wolf optimizer algorithm with a two-phase mutation for feature selection. *Expert Systems with Applications*, 139:112824.

[Abdel-Basset et al., 2021] Abdel-Basset, M., Mohamed, R., Chakrabortty, R. K., Ryan, M. J., and Mirjalili, S. (2021). An efficient binary slime mould algorithm integrated with a novel attacking-feeding strategy for feature selection. *Computers & Industrial Engineering*, 153:107078.

[Abed-alguni and Alawad, 2021] Abed-alguni, B. H. and Alawad, N. A. (2021). Distributed grey wolf optimizer for scheduling of workflow applications in cloud environments. *Applied Soft Computing*, 102:107113.

[Abu Khurma et al., 2021] Abu Khurma, R., Aljarah, I., and Sharieh, A. (2021). A simultaneous moth flame optimizer feature selection approach based on levy flight and selection operators for medical diagnosis. *ARABIAN JOURNAL FOR SCIENCE AND ENGINEERING*.

[Agrawal et al., 2021] Agrawal, P., Abutarboush, H., Talari, G., and Wagdy, A. (2021). Metaheuristic algorithms on feature selection: A survey of one decade of research (2009-2019). *IEEE Access*, PP:1–1.

[Ahmadi et al., 2021] Ahmadi, R., Ekbatanifard, G., and Bayat, P. (2021). A modified grey wolf optimizer based data clustering algorithm. *Applied Artificial Intelligence*, 35(1):63–79.

[Akyurt et al., 2021] Akyurt, I., Kuvvetli, Y., Deveci, M., Garg, H., and Yüzsever, M. (2021). A new mathematical model for determining optimal workforce planning of pilots in an airline company. *Complex Intelligent Systems*.

[Al-Betar et al., 2018] Al-Betar, M., Awadallah, M., Faris, H., Aljarah, I., and Hammouri, A. (2018). Natural selection methods for grey wolf optimizer. *Expert Systems with Applications*, 113.

[Al-Tashi et al., 2020] Al-Tashi, Q., Md Rais, H., Abdulkadir, S. J., Mirjalili, S., and Alhussian, H. (2020). *A Review of Grey Wolf Optimizer-Based Feature Selection Methods for Classification*, pages 273–286. Springer Singapore, Singapore.

[Al-Wajih et al., 2021] Al-Wajih, R., Abdulkadir, S. J., Aziz, N., Al-Tashi, Q., and Talpur, N. (2021). Hybrid binary grey wolf with harris hawks optimizer for feature selection. *IEEE Access*, 9:31662–31677.

[Alweshah et al., 2021] Alweshah, M., Alkhalaileh, S., Al-Betar, M., and Bakar, A. (2021). Coronavirus herd immunity optimizer with greedy crossover for feature selection in medical diagnosis. *Knowledge-Based Systems*, 235:107629.

[Arora and Joshi, 2017] Arora, S. and Joshi, H. (2017). Enhanced grey wolf optimisation algorithm for constrained optimisation problems. *International Journal of Swarm Intelligence*, 3:126.

[Awadallah et al., 2020] Awadallah, M., Al-Betar, M., Hammouri, A., and Alomari, O. (2020). Binary jaya algorithm with adaptive mutation for feature selection. *Arabian Journal for Science and Engineering*, 45:1–16.

[Awadallah et al., 2022] Awadallah, M. A., Hammouri, A. I., Al-Betar, M. A., Braik, M. S., and Elaziz, M. A. (2022). Binary horse herd optimization algorithm with crossover operators for feature selection. *Computers in Biology and Medicine*, 141:105152.

[Balaji et al., 2021] Balaji, E., Brindha, D., Vinodh Kumar, E., and Umesh, K. (2021). Data-driven gait analysis for diagnosis and severity rating of parkinson's disease. *Medical Engineering & Physics*, 91:54–64.

[Chantar et al., 2019] Chantar, H., Mafarja, M., Alsawalqah, H., Heidari, A. A., Aljarah, I., and Faris, H. (2019). Feature selection using binary grey wolf optimizer with elite-based crossover for Arabic text classification. *Neural Computing and Applications*, pages 1–20.

[Chantar et al., 2020] Chantar, H., Mafarja, M., Alsawalqah, H., Heidari, A. A., Aljarah, I., and Faris, H. (2020). Feature selection using binary grey wolf optimizer with elite-based crossover for arabic text classification. *Neural Computing and Applications*, 32(16):12201–12220.

[Chantar et al., 2021] Chantar, H., Thaher, T., Turabieh, H., Mafarja, M., and Sheta, A. (2021). Bhho-tvs: A binary harris hawks optimizer with time-varying scheme for solving data classification problems. *Applied Sciences*, 11(14).

[Chen et al., 2021] Chen, C., Wang, X., Chen, H., Wu, C., Mafarja, M., and Turabieh, H. (2021). Towards precision fertilization: Multi-strategy grey wolf optimizer based model evaluation and yield estimation. *Electronics*, 10(18).

[Chuang et al., 2008] Chuang, L.-Y., Li, J.-C., and Yang, C.-H. (2008). Chaotic binary particle swarm optimization for feature selection using logistic map. In *Proceedings of the International MultiConference of Engineers and Computer Scientists*, volume 1.

[Crawford et al., 2017] Crawford, B., Soto, R., Astorga, G., García, J., Castro, C., and Paredes, F. (2017). Putting continuous metaheuristics to work in binary search spaces. *Complexity*, 2017.

[Debnath et al., 2017] Debnath, M. K., Mallick, R. K., and Sahu, B. K. (2017). Application of hybrid differential evolution–grey wolf optimization algorithm for automatic generation control of a multi-source interconnected power system using optimal fuzzy–pid controller. *Electric Power Components and Systems*, 45(19):2104–2117.

[Deveci and Çetin Demirel, 2018] Deveci, M. and Çetin Demirel, N. (2018). Evolutionary algorithms for solving the airline crew pairing problem. *Computers Industrial Engineering*, 115:389–406.

[Elminaam et al., 2021] Elminaam, D. S. A., Nabil, A., Ibraheem, S. A., and Houssein, E. H. (2021). An efficient marine predators algorithm for feature selection. *IEEE Access*, 9:60136–60153.

[Emary et al., 2015] Emary, E., Zawbaa, H., A.Ghany, K., Hassanien, A. E., and Pârv, B. (2015). Firefly optimization algorithm for feature selection.

[Emary and Zawbaa, 2016] Emary, E. and Zawbaa, H. M. (2016). Impact of chaos functions on modern swarm optimizers. *PloS one*, 11(7):e0158738.

[Erdogan et al., 2021] Erdogan, N., Pamucar, D., Kucuksari, S., and Deveci, M. (2021). An integrated multi-objective optimization and multi-criteria decision-making model for optimal planning of workplace charging stations. *Applied Energy*, 304:117866.

[Faris et al., 2017] Faris, H., Aljarah, I., Al-Betar, M. A., and Mirjalili, S. (2017). Grey wolf optimizer: a review of recent variants and applications. *Neural Computing and Applications*, pages 1–23.

[Faris et al., 2018a] Faris, H., Aljarah, I., Al-Betar, M. A., and Mirjalili, S. (2018a). Grey wolf optimizer: A review of recent variants and applications. *Neural Comput. Appl.*, 30(2):413–435.

[Faris et al., 2018b] Faris, H., Mafarja, M. M., Heidari, A. A., Aljarah, I., Ala'M, A.-Z., Mirjalili, S., and Fujita, H. (2018b). An efficient binary salp swarm algorithm with crossover scheme for feature selection problems. *Knowledge-Based Systems*.

[Farughi et al., 2019] Farughi, H., Mostafayi, S., and Arkat, J. (2019). Healthcare districting optimization using gray wolf optimizer and ant lion optimizer algorithms (case study: South khorasan healthcare system in iran). *Journal of Optimization in Industrial Engineering*, 12(1):119–131.

[Ghaddar and Naoum-Sawaya, 2018] Ghaddar, B. and Naoum-Sawaya, J. (2018). High dimensional data classification and feature selection using support vector machines. *European Journal of Operational Research*, 265(3):993–1004.

[Glover and Kochenberger, 2006] Glover, F. W. and Kochenberger, G. A. (2006). *Handbook of metaheuristics*, volume 57. Springer Science & Business Media.

[Gupta and Deep, 2020] Gupta, S. and Deep, K. (2020). A memory-based grey wolf optimizer for global optimization tasks. *Applied Soft Computing*, page 106367.

[Guyon and Elisseeff, 2003] Guyon, I. and Elisseeff, A. (2003). An introduction to variable and feature selection. *Journal of machine learning research*, 3(Mar):1157–1182.

[Hammouri et al., 2020] Hammouri, A. I., Mafarja, M., Al-Betar, M. A., Awadallah, M. A., and Abu-Doush, I. (2020). An improved dragonfly algorithm for feature selection. *Knowledge-Based Systems*, 203:106131.

[Harrison et al., 2016] Harrison, K. R., Engelbrecht, A. P., and Ombuki-Berman, B. M. (2016). Inertia weight control strategies for particle swarm optimization. *Swarm Intelligence*, 10(4):267–305.

[Hassouneh et al., 2021] Hassouneh, Y., Turabieh, H., Thaher, T., Tumar, I., Chantar, H., and Too, J. (2021). Boosted whale optimization algorithm with natural selection operators for software fault prediction. *IEEE Access*, 9:14239–14258.

[Hassouneh et al., 2021] Hassouneh, Y., Turabieh, H., Thaher, T., Tumar, I., Chantar, H., and Too, J. (2021). Boosted whale optimization algorithm with natural selection operators for software fault prediction. *IEEE Access*, 9:14239–14258.

[Hastie et al., 2009] Hastie, T., Tibshirani, R., and Friedman, J. (2009). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction, Second Edition (Springer Series in Statistics)*.

[Heidari et al., 2019] Heidari, A. A., Mirjalili, S., Faris, H., Aljarah, I., Mafarja, M., and Chen, H. (2019). Harris hawks optimization: Algorithm and applications. *Future Generation Computer Systems*, 97:849 – 872.

[Heidari and Pahlavani, 2017a] Heidari, A. A. and Pahlavani, P. (2017a). An efficient modified grey wolf optimizer with lévy flight for optimization tasks. *Applied Soft Computing*, 60:115–134.

[Heidari and Pahlavani, 2017b] Heidari, A. A. and Pahlavani, P. (2017b). An efficient modified grey wolf optimizer with lévy flight for optimization tasks. *Applied Soft Computing*, 60:115–134.

[Ho and Pepyne, 2002] Ho, Y.-C. and Pepyne, D. (2002). Simple explanation of the no-free-lunch theorem and its implications. *Journal of Optimization Theory and Applications*, 115:549–570.

[Hu et al., 2020] Hu, P., Pan, J.-S., and Chu, S.-C. (2020). Improved binary grey wolf optimizer and its application for feature selection. *Knowledge-Based Systems*, 195:105746.

[Ismail Sayed et al., 2018] Ismail Sayed, G., Darwish, A., and Hassanien, A. E. (2018). A new chaotic whale optimization algorithm for features selection. *Journal of Classification*, 35.

[James et al., 2013] James, G., Witten, D., Hastie, T., and Tibshirani, R. (2013). *An Introduction to Statistical Learning: With Applications in R*.

[Jayabarathi et al., 2016] Jayabarathi, T., Raghunathan, T., Adarsh, B., and Suganthan, P. (2016). Economic dispatch using hybrid grey wolf optimizer. *Energy*, 111:630–641.

[Lai et al., 2006] Lai, C., Reinders, M. J., and Wessels, L. (2006). Random subspace method for multivariate feature selection. *Pattern recognition letters*, 27(10):1067–1076.

[Li et al., 2018] Li, J., Cheng, K., Wang, S., Morstatter, F., Trevino, R. P., Tang, J., and Liu, H. (2018). Feature selection: A data perspective. *ACM Computing Surveys (CSUR)*, 50(6):94.

[Li et al., 2017] Li, Q., Chen, H., Huang, H., Zhao, X., Cai, Z., Tong, C., Liu, W., and Tian, X. (2017). An enhanced grey wolf optimization based feature selection wrapped kernel extreme learning machine for medical diagnosis. *Computational and mathematical methods in medicine*, 2017.

[Lin et al., 2019] Lin, X., Li, C., Ren, W., Luo, X., and Qi, Y. (2019). A new feature selection method based on symmetrical uncertainty and interaction gain. *Computational biology and chemistry*, 83:107149.

[Liu et al., 2018] Liu, H., Hua, G., Yin, H., and Xu, Y. (2018). An intelligent grey wolf optimizer algorithm for distributed compressed sensing. *Computational Intelligence and Neuroscience*, 2018:1–10.

[Liu and Motoda, 2012] Liu, H. and Motoda, H. (2012). *Feature selection for knowledge discovery and data mining*, volume 454. Springer Science & Business Media.

[Liu and Wang, 2021] Liu, X. and Wang, N. (2021). A novel gray wolf optimizer with rna crossover operation for tackling the non-parametric modeling problem of fcc process. *Knowledge-Based Systems*, 216:106751.

[Long et al., 2018] Long, W., Jiao, J., Liang, X., and Tang, M. (2018). An exploration-enhanced grey wolf optimizer to solve high-dimensional numerical optimization. *Engineering Applications of Artificial Intelligence*, 68:63–80.

[LONG Wen, 2015] LONG Wen, ZHAO Dongquan, X. S. (2015). Improved grey wolf optimization algorithm for constrained optimization problem. *Journal of Computer Applications*, 35(9):2590–2595.

[Lu et al., 2017] Lu, C., Gao, L., Li, X., and Xiao, S. (2017). A hybrid multi-objective grey wolf optimizer for dynamic scheduling in a real-world welding industry. *Engineering Applications of Artificial Intelligence*, 57:61–79.

[Lu et al., 2018] Lu, C., Gao, L., and Yi, J. (2018). Grey wolf optimizer with cellular topological structure. *Expert Systems with Applications*, 107:89–114.

[Mafarja et al., 2017] Mafarja, M., Aljarah, I., Heidari, A. A., Hammouri, A. I., Faris, H., Ala'M, A.-Z., and Mirjalili, S. (2017). Evolutionary population dynamics and grasshopper optimization approaches for feature selection problems. *Knowledge-Based Systems*.

[Mafarja et al., 2020] Mafarja, M., Heidari, A. A., Habib, M., Faris, H., Thaher, T., and Aljarah, I. (2020). Augmented whale feature selection for iot attacks: Structure, analysis and applications. *Future Generation Computer Systems*, 112:18–40.

[Mafarja et al., 2019] Mafarja, M., Jaber, I., Ahmed, S., and Thaher, T. (2019). Whale optimisation algorithm for high-dimensional small-instance feature selection. *International Journal of Parallel, Emergent and Distributed Systems*, 0(0):1–17.

[Mafarja and Mirjalili, 2017] Mafarja, M. and Mirjalili, S. (2017). Hybrid whale optimization algorithm with simulated annealing for feature selection. *Neurocomputing*, 260:302–312.

[Medjahed et al., 2017] Medjahed, S. A., Saadi, T. A., Benyettou, A., and Ouali, M. (2017). Kernel-based learning and feature selection analysis for cancer diagnosis. *Appl. Soft Comput.*, 51(C):39–48.

[Mirjalili and Dong, 2020] Mirjalili, S. and Dong, J. (2020). *Multi-Objective Optimization using Artificial Intelligence Techniques*.

[Mirjalili and Lewis, 2013] Mirjalili, S. and Lewis, A. (2013). S-shaped versus v-shaped transfer functions for binary particle swarm optimization. *Swarm and Evolutionary Computation*, 9:1–14.

[Mirjalili and Lewis, 2016] Mirjalili, S. and Lewis, A. (2016). The whale optimization algorithm. *Advances in Engineering Software*, 95:51–67.

[Mirjalili et al., 2014] Mirjalili, S., Mirjalili, S. M., and Lewis, A. (2014). Grey wolf optimizer. *Advances in engineering software*, 69:46–61.

[Mittal et al., 2016a] Mittal, N., Singh, U., and Sohi, B. (2016a). Modified grey wolf optimizer for global engineering optimization. *Applied Computational Intelligence and Soft Computing*, 2016:1–16.

[Mittal et al., 2016b] Mittal, N., Singh, U., and Sohi, B. (2016b). Modified grey wolf optimizer for global engineering optimization. *Applied Computational Intelligence and Soft Computing*, 2016:1–16.

[Negi et al., 2020] Negi, G., Kumar, A., Pant, S., and Ram, M. (2020). Gwo: a review and applications. *International Journal of System Assurance Engineering and Management*, 12.

[Nguyen et al., 2020] Nguyen, B. H., Xue, B., and Zhang, M. (2020). A survey on swarm intelligence approaches to feature selection in data mining. *Swarm and Evolutionary Computation*, 54:100663.

[P et al., 2021] P, R., Mallidi, S., and Muni, R. (2021). A wrapper based feature selection using grey wolf optimization for botnet attack detection. *International Journal of Sensors, Wireless Communications and Control*, 11:1–6.

[Pappu and Pardalos, 2013] Pappu, V. and Pardalos, P. (2013). *High Dimensional Data Classification*, page 34.

[Qu et al., 2020] Qu, C., Gai, W., Zhong, M., and Zhang, J. (2020). A novel reinforcement learning based grey wolf optimizer algorithm for unmanned aerial vehicles (uavs) path planning. *Applied Soft Computing*, 89:106099.

[Riffenburgh, 2006] Riffenburgh, R. H. (2006). Chapter summaries. In Riffenburgh, R. H., editor, *Statistics in Medicine (Second Edition)*, pages 533 – 580. Academic Press, Burlington, second edition edition.

[Rostami et al., 2021] Rostami, M., Berahmand, K., Nasiri, E., and Forouzandeh, S. (2021). Review of swarm intelligence-based feature selection methods. *Engineering Applications of Artificial Intelligence*, 100:104210.

[Sepúlveda et al., 2021] Sepúlveda, A., Castillo, F., Palma, C., and Rodriguez-Fernandez, M. (2021). Emotion recognition from ecg signals using wavelet scattering and machine learning. *Applied Sciences*, 11(11).

[Sharma et al., 2020] Sharma, I., Chahar, V., and Agri, S. (2020). A comprehensive survey on grey wolf optimization. *Recent Patents on Computer Science*, pages 1–11.

[Singh et al., 2020] Singh, D., Climente-González, H., Petrovich, M., Kawakami, E., and Yamada, M. (2020). Fsnet: Feature selection network on high-dimensional biological data.

[Talbi, 2009] Talbi, E.-G. (2009). *Metaheuristics: from design to implementation*, volume 74. John Wiley & Sons.

[Thaher et al., 2020a] Thaher, T., Heidari, A. A., Mafarja, M., Dong, J., and Mirjalili, S. (2020a). *Binary Harris Hawks Optimizer for High-Dimensional, Low Sample Size Feature Selection*, pages 251–272.

[Thaher et al., 2020b] Thaher, T., Heidari, A. A., Mafarja, M., Dong, J. S., and Mirjalili, S. (2020b). Binary harris hawks optimizer for high-dimensional, low sample size feature selection. In *Evolutionary machine learning techniques*, pages 251–272. Springer.

[Thaher et al., 2021a] Thaher, T., Mafarja, M., Turabieh, H., Castillo, P. A., Faris, H., and Aljarah, I. (2021a). Teaching learning-based optimization with evolutionary binarization schemes for tackling feature selection problems. *IEEE Access*, 9:41082–41103.

[Thaher et al., 2021b] Thaher, T., Saheb, M., Turabieh, H., and Chantar, H. (2021b). Intelligent detection of false information in arabic tweets utilizing hybrid harris hawks based feature selection and machine learning models. *Symmetry*, 13(4).

[Too and Abdullah, 2020] Too, J. and Abdullah, A. R. (2020). Opposition based competitive grey wolf optimizer for emg feature selection. *Evolutionary Intelligence*, pages 1–15.

[Tu et al., 2019] Tu, Q., Chen, X., and Liu, X. (2019). Hierarchy strengthened grey wolf optimizer for numerical optimization and feature selection. *IEEE Access*, 7:78012–78028.

[Tu et al., 2019] Tu, Q., Chen, X., and Liu, X. (2019). Multi-strategy ensemble grey wolf optimizer and its application to feature selection. *Applied Soft Computing*, 76:16–30.

[Tumar et al., 2020] Tumar, I., Hassouneh, Y., Turabieh, H., and Thaher, T. (2020). Enhanced binary moth flame optimization as a feature selection algorithm to predict software fault prediction. *IEEE Access*, PP:1–1.

[Urbanowicz et al., 2018] Urbanowicz, R. J., Meeker, M., La Cava, W., Olson, R. S., and Moore, J. H. (2018). Relief-based feature selection: Introduction and review. *Journal of Biomedical Informatics*, 85:189–203.

[Wolpert and Macready, 1997] Wolpert, D. H. and Macready, W. G. (1997). No free lunch theorems for optimization. *IEEE transactions on evolutionary computation*, 1(1):67–82.

[Xue et al., 2014] Xue, B., Zhang, M., and Browne, W. N. (2014). Particle swarm optimisation for feature selection in classification: Novel initialisation and updating mechanisms. *Applied Soft Computing*, 18:261–276.

[Xue et al., 2016] Xue, B., Zhang, M., Browne, W. N., and Yao, X. (2016). A survey on evolutionary computation approaches to feature selection. *IEEE Transactions on Evolutionary Computation*, 20(4):606–626.

[Zawbaa et al., 2016] Zawbaa, H., Emary, E., and Hassanien, A. E. (2016). Binary grey wolf optimization approaches for feature selection. *Neurocomputing*, 172:371–381.

[Zhao et al., 2010] Zhao, Z., Morstatter, F., Sharma, S., Alelyani, S., Anand, A., and Liu, H. (2010). Advancing feature selection research. *ASU feature selection repository*, pages 1–28.

[Çetin Demirel and Deveci, 2017] Çetin Demirel, N. and Deveci, M. (2017). Novel search space updating heuristics-based genetic algorithm for optimizing medium-scale airline crew pairing problems. *International Journal of Computational Intelligence Systems*, 10:1082 – 1101.