INSTITUTE OF INFORMATION AND COMMUNICATION TECHNOLOGIES
BULGARIAN ACADEMY OF SCIENCES
CYBERNETICS AND INFORMATION TECHNOLOGIES • Volume **26**, No 1
Sofia • 2026 Print ISSN: 1311-9702; Online ISSN: 1314-4081
DOI: 10.2478/cait-2026-0009

Binary Fox Optimization Algorithm for Feature Selection

Athraa Jasim Mohammed[1], *Khalil Ibrahim Ghathwan*[1], *Yuhanis Yusof*[2]

[1]*College of Computer Science, University of Technology – Iraq, 10066 Baghdad, Iraq*

[2]*College of Arts and Sciences, University Utara Malaysia, 06010 UUM Sintok, Kedah, Malaysia*,

E-mails: athraa.j.mohammed@uotechnology.edu.iq khalil.i.ghathwan@uotechnology.edu.iq yuhanis@uum.edu.my

***Abstract**: There is an urgent need for algorithms capable of improving the selection of appropriate features that directly affect the improvement process of the algorithm's accuracy efficiently. Therefore, in this paper, a new feature selection algorithm for binary classification is proposed, which is called binary FOX Optimization Feature Selection (BFOXFS), where BFOXFS is combined with the KNN algorithm, which is used for binary classification, and the classification error is utilized as the objective function for the proposed BFOXFS. For evaluation, BFOXFS is compared with Binary Particle Swarm Optimization Feature Selection (BPSOFS), and two versions of Binary Grey Wolf optimization algorithm for Feature Selection (BGW1FS and BGW2FS algorithms). The experimental results show the superiority of BFOXFS in feature reduction, where it chooses the minimum features with a lower fitness value comparable with others. Further, BFOXFS has better convergence capabilities, well at finding optimal values, which results in more reliable and efficient optimization algorithms. It is very suitable for implementing in real-world problems.*

***Keywords**: Fox optimization algorithm, Feature selection, Binary feature selection.*

1. Introduction

Feature selection has been an active area of research in data mining, pattern recognition, statistics, and other engineering communities. Given the broad applications of feature selection in most engineering problems, reviewing different feature selection methods can be effective in choosing the appropriate approach for the application [1]. The objective of the feature selection technique is to choose a subset of features that have the least internal similarity and maximum relevance to the target class. The technique helps to narrow down the data dimension by eliminating irrelevant, redundant, or noisy data [2]. The methods of feature selection tend to search the solution space to ensure two opposing objectives: to maximize the relevance to the target class and to decrease the redundancy of features selected. These aims are optimized by several search strategies [2].

The full search space for selecting the most interesting non-redundant features is 2^n, where n denotes the original number of features. Comprehensive searches guarantee finding the most relevant features; however, it is frequently computationally infeasible, even with medium-sized data sets. Due to the prohibitive complexity of considering all possible combinations, a computationally efficient and qualitatively useful solution has to be identified. In order to avoid high computational complexity, most feature selection methods apply meta-heuristic methods [2, 3].

The ability of meta-heuristics algorithms to sufficiently optimize the feature selection problem within an acceptable timeframe is what distinguishes them. These algorithms fall into two main groups: Evolutionary Algorithms (EA) and Swarm Intelligence (SI). Unlike EA and other single-solution approaches, swarm intelligence is a relatively new class of evolutionary computing. Evolutionary AI algorithms use approximate, non-deterministic methods to efficiently and successfully explore and exploit the search space to discover near-optimal solutions. The most well-known class of meta-heuristics inspired by nature is swarm-based methods. In decentralized, self-organizing systems, swarm intelligence is a type of AI based on group behavior. Such systems typically consist of a set of basic components that interact with their surroundings and with each other locally [4]. There are various heuristic algorithms, such as Fox Optimization Algorithm (FOX) [5], Grey Wolf Optimization Algorithm (GWO) [6, 7], Particle Swarm Optimization Algorithm (PSO) [8, 9].

The FOX Algorithm is a new Meta-heuristics algorithm inspired by fox behavior. The FOX algorithm has the advantages of a simple structure and fast convergence, and it also performs well in some engineering optimization problems [5, 10]. It has success in diverse topics such as: biomedical [11], brain tumour diagnosis [12], Wireless Sensor Network [13]. This paper proposes a feature selection method using the binary Fox Optimization Algorithm.

The structure of this paper is as follows: Section 2 presents a literature review of recent studies on feature selection. Section 3 includes materials and methods that contain the dataset, which normalizes details about the proposed Binary Fox Optimization Algorithm for Feature Selection. Section 4 presents the results and discussions of the experimental results, while Section 5 concludes the study.

2. Literature review

In [14], a PSO optimization process was presented that models competition between swarms by establishing four rules and a survival-of-the-fittest mechanism. A modified Multi-Swarm PSO (MSPSO) optimization procedure was proposed to address discrete problems, based on this mechanism. This procedure contains a multi-swarm scheduler that can monitor and manage each sub-swarm using rules, as well as multiple sub-swarms. By combining F-score, Support Vector Machines (SVMs), and MSPSO, the researcher also presented an Integrated Feature Selection (IFS) optimization approach. The IFS method seeks to enhance generalization by simultaneously performing feature selection and kernel parameter optimization. In paper [15], the feature selection problem was addressed using a hybrid optimization technique that combines particle swarm optimization and the Salp Swarm Algorithm (SSA). By combining the two methods, an algorithm known as SSAPSO was created, which increases the efficiency of both the exploration and exploitation phases. In [16], a better optimization algorithm for whales called EWOA was presented, using a clustering mechanism and three efficient search strategies called: migratory, preferential selection, and enriched surrounding prey. A Binary EWOA algorithm, known as BEWOA, is also proposed for efficient feature selection, especially from medical datasets.

Two strategies for proposing several variations on the Binary MBO (BMBO) for exploratory feature selection were described in [17]. First, to prepare MBO to address feature selection optimization problems, S-shaped and V-shaped transfer functions were constructed to transform the continuous space into binary and then force the butterfly to move in the binary search space. The first BMBO mechanism, BMBOS and BMBOV, was constructed using two updated monarch butterfly population locations based on the above transfer functions. Second, to improve the accuracy of monarch butterfly population locations, a new step length parameter was introduced. To prevent MBO from falling into a local optimum, local perturbation and population partitioning tactics were introduced, resulting in a new BMBO approach. To prevent MBO from converging prematurely, a mutation rate was used to optimize the discovery phase of BMBO, and a mutation operator-based BMBO (BMBOM) was constructed.

In [18], the authors proposed modifications to the Binary Grey Wolf Optimizer (BGWO) to address the Feature Selection (FS) problem caused by high-dimensional data, noise, redundancy, and irrelevance. To address this problem, they proposed three alternative forms of the BGWO, each using a different transfer function in addition to the traditional one. To convert the continuous values produced by the BGWO to binary values, several V-, S-, and U-shaped transfer functions were tested for integration with the BGWO, since feature selection requires discrete values.

This paper proposes a Binary Fox optimization algorithm for feature selection, which has not been done.

2.1. Grey Wolf Optimization Algorithm

The Grey Wolf Optimizer Algorithm (GWO Algorithm) is a heuristic based on the social behavior of grey wolves in the wild, particularly their pack structure and hunting behavior. It was initially proposed by Seyed Ali Mirjalili in 2014 [19]. The pseudo code of GWO is shown in Fig. 1.

The mathematical model of grey wolves' hunting behaviour underpins GWO. The remaining wolves represent alternative potential solutions, with the best solution known as alpha (α), the second-best solution known as beta (β), and the third-best solution known as delta (δ). The hunting tactics used in GWO, which correspond to exploring and exploiting a search space, involve tracking, capturing, and ultimately attacking prey. The algorithm operates by initializing several wolves (random solutions) in the area to be optimized. The top three answers – α, β, and δ – are found using an objective function (fitness function) to evaluate these solutions [7, 20]. The wolf locations are then updated based on the top three wolves' locations using the equation

$$X_{\text{new}} = (X_1 + X_2 + X_3) / 3, \tag{1}$$

where X_1, X_2, and X_3 are calculated based on the three best solutions α, β, and δ.

Initialization

- Initialize *a*, *A*, *C*
- Calculate the fitness of each agent (wolf)
- X_α = the best search agent
- X_β = the second best search agent
- X_δ = the third best search agent

Process

- While (t < max-iteration)
- For 1 to n
- Update the position using Equation (1)
- End for
- Update *a*, *A*, *C*
- Calculate the fitness of all search agents
- Update X_α, X_β, X_δ
- $t = t + 1$
- end while

Output

- Return best solution X_α

Fig. 1. Pseudo code of GWO Algorithm [7, 20]

2.2. Particle Swarm Optimization Algorithm

Particle Swarm Optimization Algorithm (PSO Algorithm) is a computational technique inspired by the collective actions of the natural world, particularly the behavior of flocks of birds during foraging. It was first devised by Jim Kennedy and Russell Eberhart in 1995 [21]. The PSO Algorithm is straightforward to implement, requires no gradients (making it suitable for non-differentiable problems), and is effective for exploring large spaces. The PSO pseudo code is shown in Fig. 2 [22, 23].

Initialization

- Initialize the particles population X_j (j=1, 2, …, n) and the velocity V_j (j=1, 2, …, n)
- Initialize pbest, gbest
- Generate random position for particles
- For each particle (j)
- Calculate the fitness of each agent (particle)
- Update pbest, gbest
- End for

Process

- While (t < max-iteration)
- For each particles (j)
- Update the position and using velocity using the equation

(2) $$X_i = X_i + V_i\,,$$

$$V_i = w \times V_i + c_1 \times r_1 \times (X_{\mathrm{pb}i} - X_i) + c_2 \times r_2 \times (X_{\mathrm{gb}} - X_i)$$

- If X_j > limit then X_j = limit
- Calculate the fitness f_i
- Update pbest, gbest
- End for
- End while

Output

- Return best solution

Fig. 2. Pseudo code of PSO Algorithm [22, 23]

The PSO Algorithm represents a set of particles moving around the search space. Each particle represents a potential solution to the problem and moves according to its best position (personal best – pbest) and the best position reached by the entire swarm (global best – gbest) [27, 28]. The first step in PSO is to initialize the swarm by generating a random number of particles, each with a position and velocity. The position of these particles is then evaluated using an objective function, and each particle is analysed to see how well its solution performs. The optimal position is then modified. The pBest value for the particle is updated if the modified solution is better than the old one. The global best position is updated and compared to pbest. It is also updated if it is better than gbest. The position and velocity are updated using the Equation (2) [22, 23].

2.3. FOX Algorithm

The Fox Optimization Algorithm, based on the foraging habits of foxes, was used as a model for an optimization algorithm known as the "algorithm". It first appeared in 2023. Like other foraging algorithms, this algorithm is characterized by its simplicity and straightforward implementation. The choice of FOX is based on its advanced mechanisms that address the shortcomings of existing foraging algorithms, achieving a better balance between exploration and exploitation phases. FOX's random walk and distance metrics enable a more accurate search than PSO, GWO, and BOA algorithms. Its jumping mechanism (jum) also improves the ability to exceed local optimums, leading to better results in benchmark tests and real-world engineering design challenges [5, 10]. Fig. 3 shows the pseudo code of the FOX Algorithm [10, 24].

Initialization

- Initialize the population
- Determine the C_1, C_2

Process

- While (t < max-iteration)
- Calculate $a = 2\times(j - [1/\max(j)])$.
- Calculate the fitness of current agent.
- If random(0,1) ≥ 0.5 {
- If random(0,1) ≥ 0.18
- Calculate new position X using the equation

(3) $$X_j + 1 = \mathrm{Dist(FP)}_j \times \mathrm{jum}_j \times C_1$$

- Check X if it beyond limitation
- Else
- Calculate new position X using the equation

(4) $$X_j + 1 = \mathrm{Dist(FP)}_j \times \mathrm{jum}_j \times C_2$$

- Check X if it beyond limitation
- Else
- Calculate new position X using the equation

(5) $$X_j + 1 = \mathrm{Best}X_j \times \mathrm{rand}(1, \mathrm{dim}) \times \min(\mathrm{timeaverage}) \times a$$

- Check X if it beyond limitation
- End while

Output

- Calculate the fitness and return the best one

Fig. 3. Pseudo code of Fox Algorithm [10, 24]

Each member of the swarm is referred to as a "fox agent". Foxes alternate between two operations: exploration and exploitation. Exploration uses a structured random walk to randomly search for new regions in the solution space. Conversely, exploitation involves using a mathematically derived dimension to jump toward the most common solution (the prey) [5, 10].

The Fox population begins by creating a population of agents with random locations. An objective function is computed for each agent to evaluate fitness. Each agent decides whether to explore or exploit, and the path is selected based on a predetermined probability (usually 0.5). The location is then determined, and the location is updated accordingly. A new location is computed, and a hop is made based on this dimension if the agent is in exploitation mode. Migration is performed using an ordered random walk if the agent is in exploration mode. If a better solution is discovered, the best-known location is updated to reflect it. The process repeats until agreement is reached or the required number of iterations is completed [10, 24].

3. Materials and methods

In the feature selection problem, the solution values are represented in binary form; there is a need to adapt the Fox algorithm to work in binary form, that is, the Binary Fox Optimization Algorithm (BFOXFS). Fig. 4 shows the flow diagram of the proposed BFOXFS.

First, the dataset is normalized to values between 0 and 1 using the min-max normalization. Then, 20% of the dataset is extracted as testing data. Secondly, generate initial solutions of X (the generated number between 0 and 1 and greater than 0.5) with dimension (i, d), where i represents the number of fox agents and d represents the number of features of the dataset, and generate the value of C_1= 0.18 and C_2=0.82, and determine the lower bound lb=0 and upper bound ub=1. Thirdly, evaluate the initial solution using an objective function that computes the classification error. The objective function is combined with the KNN (K-Nearest Neighbor) classifier. Where the dataset is trained by a KNN-classifier to build the classification model, then the model is used to test the testing data to predict the correct classification and to calculate the error.

Fourthly, the iteration starts until reaching the maximum number of iterations. In each iteration, every solution of X is returned to specific boundaries of the search space, and the objective function is calculated for each search agent. If the fitness of the agent is better than the best solution, then replace it. Generate two random numbers r and p. If r is greater than or equal to 0.5 and p is greater than 0.18, calculate a new solution using the Equation (3); if p is less than or equal to 0.18, calculate using the Equation (4); if r is less than 0.5, then calculate using the Equation (5).

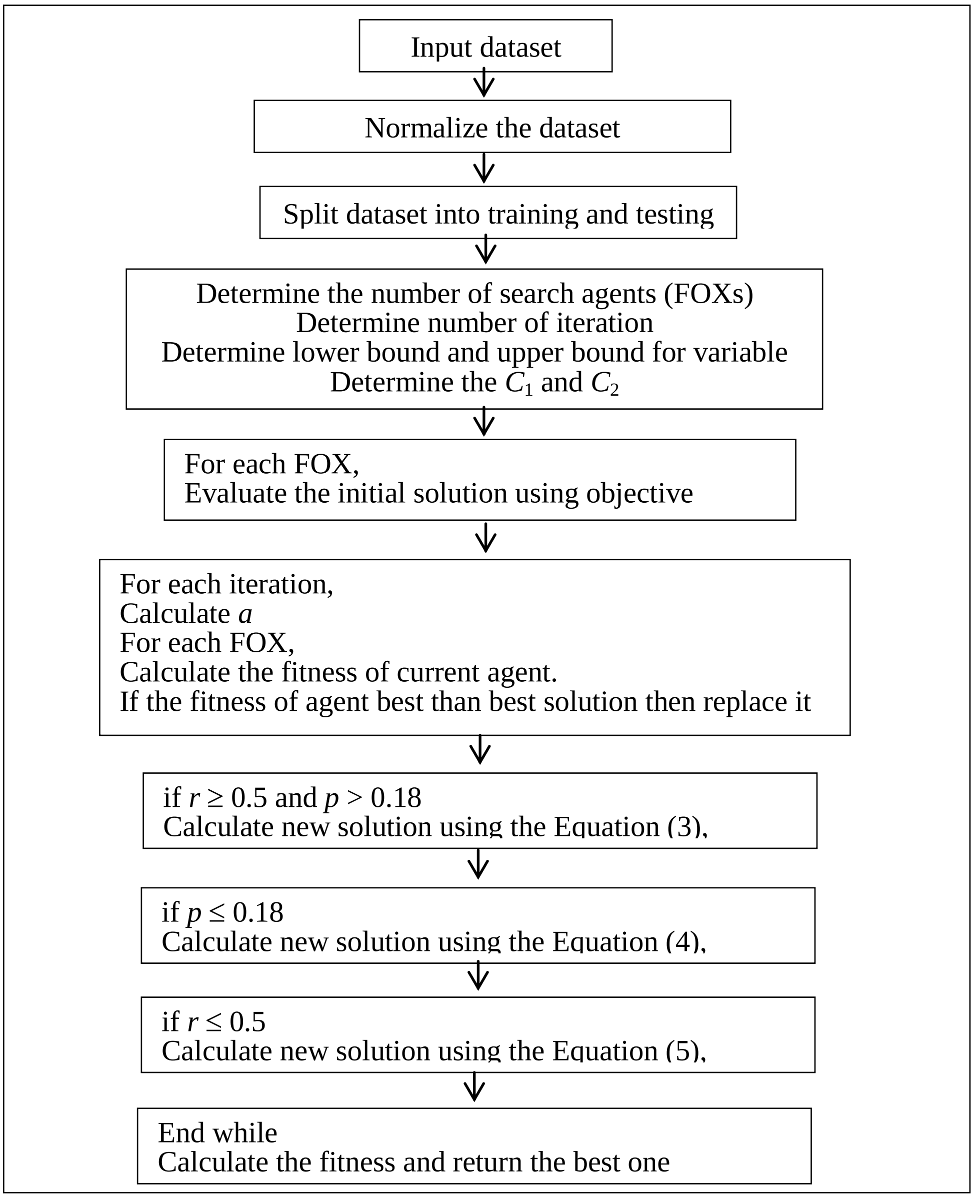

Fig. 4. The flow diagram of the proposed BFOXFS

4. Results and discussions

In this section, the performance of the proposed binary Fox Optimization Algorithm for Feature Selection is compared with that of the comparative methods BGW1FS [7, 20], BGW2FS [7, 20], and BPSOFS [25-26] using three datasets. The results express the Optimal Efficiency (OF), which reflects the classification error, the Selected Features (SF), and the quaNtity of selected Features (NF). The classification error is based on the calculation of classification accuracy, as shown in the equation bellow, which represents the classification error,

$$\begin{aligned}(6)\quad &\text{ErrorofClassification} = 1 - \text{ClassificationAccuracy},\\ &\text{ClassificatinAccuracy} = (\text{TP} + \text{TN}) / (\text{TP} + \text{TN} + \text{FP} + \text{FN}),\end{aligned}$$

where: TP is True Positive, TN is True Negative, FP is False Positive, and FN is False Negative.

In each experiment, each feature selection method is run ten times. Further, convergence analysis is conducted to test the behavior of the Feature Selection method for reaching an optimal solution

4.1. Benchmark dataset

From the UCI Respiratory (UCIR) [29], three datasets with different numbers of features and instances are used in this study. The first dataset is the Ionosphere dataset, with 34 columns. The data are continuous, with one column indicating whether an observation is "good" or "bad." There are 351 records in the dataset. The Wisconsin Diagnostic Breast Cancer Dataset, which consists of 31 continuous columns and one column with malignant or benign values, is the second dataset. There are 569 cases in the dataset. The third dataset is the default for credit card customers. One feature has an integer value of 0 or 1, while the other 23 features have integer values. There are 30,000 instances in the dataset. All three datasets are used in the classification problem.

4.2. Parameters setup

The proposed Binary Fox optimization Algorithm for Feature Selection method and comparative methods BGW1FS, BGW2FS, and BPSOFS were executed on MatlabR2018b with Windows 10 operating system. The parameters setting of the experiment for the proposed method and comparative methods are shown in Table 1.

Table 1. Parameters setting of the BFOXFS, BGW1FS, BGW2FS, and BPSOFS methods

Parameter	BFOXFS	BGW1FS	BGW2FS	BPSOFS
Number of population	10	10	10	10
Maximum number of iterations	100	100	100	100
Lower bound and upper bound for a variable	lb= 0, and ub=1	lb= 0, and ub=1	lb= 0, and ub=1	lb= 0, and ub=1
C_1	0.18	-	-	2
C_2	0.82	-	-	2
w	-	-	-	1

4.3. Experimental results

In Table 2, the results of Optimal Fitness (OF) (which refers to minimum classification error), Selected Features (SF), and the Number of selected Features (NF) for the proposed BFOXFS, BGW1FS, BGW2FS, and BPSOFS algorithms for the Ionosphere dataset are presented. The Optimal Fitness of the proposed BFOXFS for tens of executions is shown in Table 2, where 6 out of 10 executions find the optimal fitness compared to the other BGW1FS, BGW2FS, and BPSOFS algorithms. The result of the number of selected features proposed by BFOXFS is 9 from 10 executions, which is lower than others.

As can be noticed in Table 3, AOF for the proposed BFOXFS are (0.041429, 0.0150443, and 0.182867) for Ionosphere, Diagnostic Wisconsin Breast Cancer, and default of credit card clients datasets, respectively. Followed by the BGW2FS algorithm, which is (0.102857, 0.0168144, and 0.197583) in three datasets, followed by the BPSOFS algorithm, which is (0.067143, 0.197167) in Ionosphere and default of credit card clients, while the BGW1FS algorithm acquires (0.0168144) in the Diagnostic Wisconsin Breast Cancer dataset, better than BPSOFS only. Conclude that the proposed BFOXFS has lower errors, which is better than others.

Figs 5, 6, and 7 show the Optimal Fitness for proposed BFOXFS, BGW1FS, BGW2FS, and BPSOFS algorithms for Ionosphere, Diagnostic Wisconsin Breast Cancer, and default of credit card, clients' datasets, respectively. It can be seen that the curve of the proposed BFOXFS is lower (i.e., lower error value) than others.

Table 2. Results of OF, SF, and NF for proposed BFOXFS, BGW1FS, BGW2FS, and BPSOFS algorithms on the Ionosphere dataset

Number of executions	Algorithm	Optimal Fitness (OF)	Selected Features (SF)	Number of selected Features (NF)
1	BFOXFS	**0.028571**	1 8 9 14 21 27	**6**
	BGW1FS	0.100000	1 2 3 4 5 7 8 17 18 19 20 21 26 27 28 29 30 31 32 33	20
	BGW2FS	0.042857	1 2 3 4 7 8 9 16 17 18 20 22 24 25 27 34	16
	BPSOFS	0.100000	5 9 11 15 18 22 24 25 27 34	10
2	BFOXFS	0.042857	1 3 5 10	**4**
	BGW1FS	0.085714	2 6 7 8 13 14 15 17 21 22 24 25 27 28 31 33 34	17
	BGW2FS	0.057143	6 7 12 14 20 28 34	7
	BPSOFS	**0.028571**	3 5 18 22	**4**
3	BFOXFS	**0.028571**	1 8 13 15 27	**5**
	BGW1FS	0.057143	1 2 4 5 8 13 14 15 18 19 20 22 23 24 26 27 29	17
	BGW2FS	0.057143	3 6 9 10 13 14 23 27 31	9
	BPSOFS	0.057143	1 2 3 5 7 8 13 16 23	9
4	BFOXFS	**0.028571**	13 17 21 24 33	**5**
	BGW1FS	0.128571	1 3 4 5 6 11 13 14 16 20 22 25 26 27 32 33 34	17
	BGW2FS	0.014286	1 2 3 8 14 15 22 24 27 29 31	11
	BPSOFS	**0.028571**	1 2 3 5 11 12 14 19 21 23 26 30 32	13
5	BFOXFS	**0.042857**	17 21 24 27 30	**5**
	BGW1FS	0.100000	2 3 8 11 14 15 17 18 22 23 25 27 28 31 32 33	16
	BGW2FS	0.057143	1 5 7 13 20 21 23 29 32 34	10
	BPSOFS	0.057143	2 11 15 17 19 22 23 24 25 30	10

Table 2 (c o n t i n u e d)

Number of executions	Algorithm	Optimal Fitness (OF)	Selected Features (SF)	Number of selected Features (NF)
6	BFOXFS	**0.071429**	6 20 21 23	**4**
	BGW1FS	0.100000	1 3 5 6 11 13 14 15 16 18 19 21 22 23 25 26 28 31 34	19
	BGW2FS	0.114286	1 2 3 9 11 15 19 22 23	9
	BPSOFS	**0.071429**	3 10 12 14 19 21 23	7
7	BFOXFS	**0.071429**	3 7 8 10 18 21 23	7
	BGW1FS	0.114286	1 2 5 6 11 12 14 15 16 17 22 24 27 32 34	15
	BGW2FS	0.014286	1 2 3 4 6 8 9 11 17 18 20 25 27 30 31 34	16
	BPSOFS	0.100000	1 3 6 16 17 19	**6**
8	BFOXFS	**0.028571**	5 9 10 18 22 26 27 33	**8**
	BGW1FS	0.114286	1 2 11 12 13 15 17 18 19 20 21 23 25 27 28 30 31 33	18
	BGW2FS	0.042857	5 8 9 11 13 23 24 25 27	9
	BPSOFS	0.085714	5 7 8 9 11 13 16 18 20 22 34	11
9	BFOXFS	**0.014286**	3 6 14 15 22	**5**
	BGW1FS	0.142857	3 4 5 8 9 10 12 15 16 17 18 19 22 23 24 25 27 30 33	19
	BGW2FS	0.057143	1 3 4 5 6 7 8 16 20 23 25 28 31 34	14
	BPSOFS	0.085714	1 7 10 11 13 16 17 18 19 20 21 27 31 34	14
10	BFOXFS	**0.057143**	3 6 14 15 19 27	**6**
	BGW1FS	0.085714	1 2 5 6 9 12 14 16 20 21 24 27 29 31 34	15
	BGW2FS	0.071429	1 2 5 11 12 15 18 21 22 23 29 31 34	13
	BPSOFS	**0.057143**	1 5 6 7 13 16 19 27 34	9

In Table 3, the results of Average Optimal Fitness (AOF) for proposed BFOXFS, BGW1FS, BGW2FS, and BPSOFS algorithms for three datasets are presented.

Table 3. The Results of Average Optimal Fitness for proposed BFOXFS, BGW1FS, BGW2FS, and BPSOFS algorithms for three datasets

Name of dataset	Proposed BFOXFS	BGW1FS	BGW2FS	BPSOFS
Ionosphere	**0.041429**	0.102857	0.052857	0.067143
Diagnostic Wisconsin Breast Cancer	**0.0150443**	0.0168144	0.0150445	0.0221241
Default of credit card clients	**0.182867**	0.197583	0.193733	0.197167

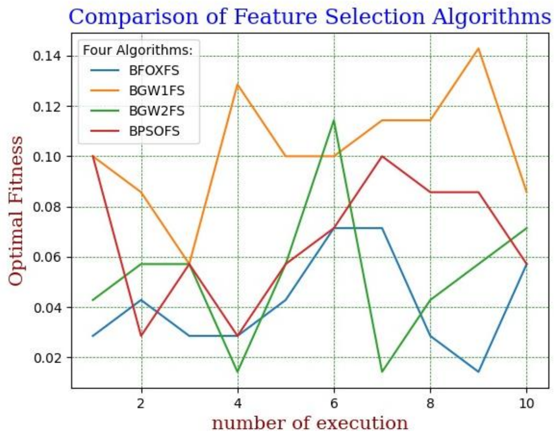

Fig. 5. The Optimal Fitness for the proposed BFOXFS, BGW1FS, BGW2FS, and BPSOFS algorithms for the Ionosphere dataset

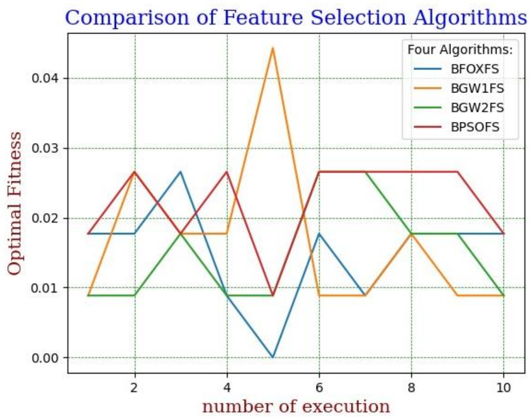

Fig. 6. The Optimal Fitness for the proposed BFOXFS, BGW1FS, BGW2FS, and BPSOFS algorithms for the Diagnostic Wisconsin Breast Cancer dataset

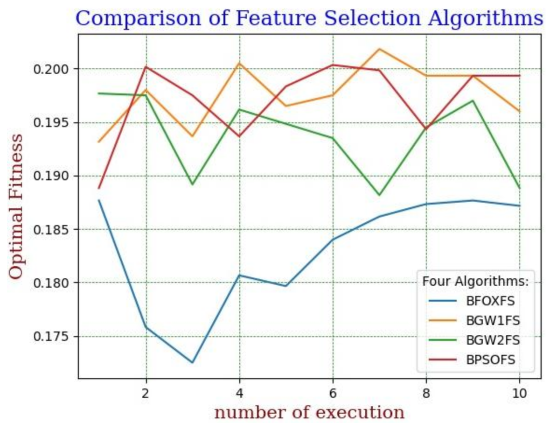

Fig. 7. The Optimal Fitness for proposed BFOXFS, BGW1FS, BGW2FS, and BPSOFS algorithms for the default of credit card clients dataset

Figs 8, 9, and 10 show the number of selected features for the proposed BFOXFS, BGW1FS, BGW2FS, and BPSOFS algorithms for the Ionosphere, Diagnostic Wisconsin Breast Cancer, and default of credit card clients datasets, respectively, where it can be seen that the curve of the proposed BFOXFS is lower (i.e. lower number of selected features) than the others.

Fig. 8. The Number of Feature Selection for the proposed BFOXFS, BGW1FS, BGW2FS, and BPSOFS algorithms for the Ionosphere dataset

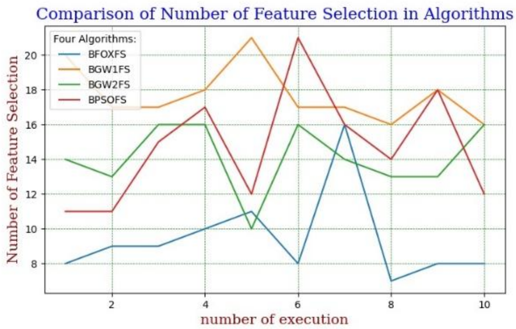

Fig. 9. The Number of Feature Selection for proposed BFOXFS, BGW1FS, BGW2FS, and BPSOFS algorithms for the Diagnostic Wisconsin Breast Cancer dataset

Fig. 10. The Number of Feature Selection for proposed BFOXFS, BGW1FS, BGW2FS, and BPSOFS algorithms for the default of the credit card clients dataset

4.4. Convergence analysis

In meta-heuristic optimization methods, convergence analysis is a fundamental concept that assesses how well an optimization algorithm converges to the optimal solution. It is a fundamental component of optimization, as it helps compare different optimization algorithms, identify the conditions under which they converge, better understand their efficiency, and ensure their reliability. The exploration and exploitation elements of all heuristic optimization methods should be considered in this study.

The convergence curves from the experiments are shown in Fig. 11. As shown in the figure, the straight line of the BGW1FS Algorithm (orange line) gets stuck at a local optimal value for long periods earlier (before iteration 20), while the BGW2FS Algorithm (green line) gets stuck at a local optimal value after iteration 40, and the BPSOFS Algorithm (red line) repeatedly gets stuck at a local optimal value before iteration 40. The proposed BFOXFS Algorithm exhibits superior convergence properties, continuously improving and avoiding long periods of stagnation at local optima.

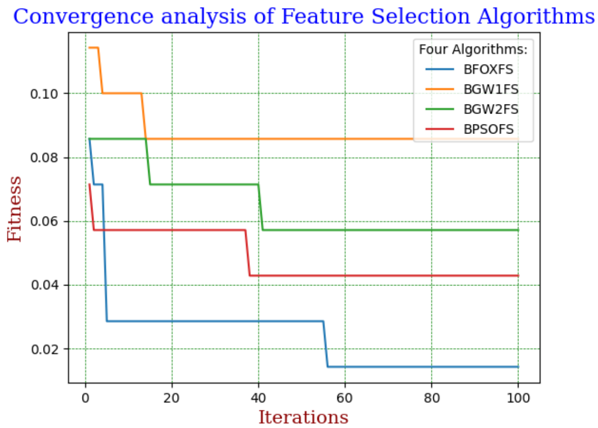

Fig. 11. The Convergence Analysis for proposed BFOXFS, BGW1FS, BGW2FS, and BPSOFS algorithms

5. Conclusions

This work presents a novel feature selection method for binary classification called Binary Fox Optimization Feature Selection (BFOXFS) to address the feature selection problem across three different datasets. Each dataset is used to execute the proposed BFOXFS ten times. The performance of BFOXFS is compared with two versions of the Binary Grey Wolf Optimization Algorithm for Feature Selection (BGW1FS and BGW2FS) and Binary Particle Swarm Optimization for Feature Selection (BPSOFS). Experimental data show that the proposed BFOXFS achieves Average Optimal Fitness (AOF) with lower error on three datasets: the Ionosphere, breast cancer diagnoses in Wisconsin, and credit card defaults. These results outperform the compared methods: BGW1FS, BGW2FS, and BPSOFS. In addition, the number of features selected in the proposed BFOXFS across all datasets is lower than that of other models. The proposed BFOXFS method also demonstrated superior convergence capabilities, continuously improving and not remaining at a local optimal value for long, unlike other algorithms. Therefore, we conclude that the proposed BFOXFS has proven its superiority in feature reduction, making it highly applicable in practical cases.

***Acknowledgement*:** The authors declare no conflicts of interest to report regarding the present study. Further, we would like to extend our deepest gratitude to the research participants for their invaluable knowledge and guidance throughout this research. Their contributions have been crucial to the advancement of this research.

References

1. Feda, A. K., M. Adegboye, O. R. Adegboye, E. B. Agyekum, W. Fendzi Mbasso, S. Kamel. S-Shaped Grey Wolf Optimizer-Based FOX Algorithm for Feature Selection. – Heliyon, Vol. **10**, 2024, No 2. DOI: 10.1016/j.heliyon.2024.e24192.
2. Rostami, M., K. Berahmand, E. Nasiri, S. Forouzandeh. Review of Swarm Intelligence-Based Feature Selection Methods. – Engineering Applications of Artificial Intelligence, Vol. **100**, 2021, 104210. DOI: 10.1016/j.engappai.2021.104210.
3. Rostami, M., K. Berahmand, S. Forouzandeh. A Novel Community Detection Based Genetic Algorithm for Feature Selection. – Journal of Big Data, Vol. **8**, 2021, No 1. DOI: 10.1186/s40537-020-00398-3.
4. Barrera-García, J., F. Cisternas-Caneo, B. Crawford, M. Gómez Sánchez, R. Soto. Feature Selection Problem and Metaheuristics: A Systematic Literature Review about Its Formulation, Evaluation and Applications. – In: Biomimetics Multidisciplinary Digital Publishing Institute (MDPI), Vol. **9**, 2024, Issue 1. DOI: 10.3390/biomimetics9010009.
5. Zhang, Z., X. Wang, L. Cao. FOX Optimization Algorithm Based on Adaptive Spiral Flight and Multi-Strategy Fusion. – Biomimetics, Vol. **9**, 2024, No 9. DOI: 10.3390/biomimetics9090524.
6. Mirjalili, S., S. M. Mirjalili, A. Lewis. Grey Wolf Optimizer. – Advances in Engineering Software, Vol. **69**, 2014, pp. 46-61. DOI: 10.1016/j.advengsoft.2013.12.007.
7. Too, J., A. R. Abdullah. Opposition-Based Competitive Grey Wolf Optimizer for EMG Feature Selection. – Evolutionary Intelligence, Vol. **14**, 2021, No 4, pp. 1691-1705. DOI: 10.1007/s12065-020-00441-5.
8. Gad, A. G. Particle Swarm Optimization Algorithm and Its Applications: A Systematic Review. – Archives of Computational Methods in Engineering, Vol. **29**, 2022, No 5, pp. 2531-2561. DOI: 10.1007/s11831-021-09694-4.
9. Shami, T. M., A. A. El-Saleh, M. Alswaitti, Q. Al-Tashi, M. A. Summakieh, S. Mirjalili. Particle Swarm Optimization: A Comprehensive Survey. – IEEE Access, Vol. **10**, 2022, pp. 10031-10061. DOI: 10.1109/ACCESS.2022.3142859.
10. Al Rahhal, H., R. Jamous. AFOX: A New Adaptive Nature-Inspired Optimization Algorithm. – Artificial Intelligence Review, Vol. **56**, 2023, No 12, pp. 15523-15566. DOI: 10.1007/s10462-023-10542-z.
11. Khorami, E., F. Mahdi Babaei, A. Azadeh. Optimal Diagnosis of COVID-19 Based on Convolutional Neural Network and Red Fox Optimization Algorithm. – Computational Intelligence and Neuroscience, Vol. **2021**, 2021, Article ID 4454507. DOI: 10.1155/2021/4454507.
12. Xu, L., M. Mohammadi. Brain Tumor Diagnosis from MRI Based on Mobilenetv2 Optimized by Contracted Fox Optimization Algorithm. – Heliyon, Vol. **10**, 2024, No 1. DOI: 10.1016/j.heliyon.2023.e23866.

13. N a t a r a j a n, R., G. M e g h a r a j, A. M a r c h e w k a, P. B. D i v a k a r a c h a r i, M. R. H a n s. Energy and Distance-Based Multi-Objective Red Fox Optimization Algorithm in Wireless Sensor Network. – Sensors, Vol. **22**, 2022, No 10. DOI: 10.3390/s22103761.
14. L i u, Y., G. W a n g, H. C h e n, H. D o n g, X. Z h u, S. W a n g. An Improved Particle Swarm Optimization for Feature Selection. – Journal of Bionic Engineering, Vol. **8**, 2011, No 2, pp. 191-200. DOI: 10.1016/S1672-6529(11)60020-6.
15. I b r a h i m, R. A., A. A. E w e e s, D. O l i v a, M. A b d E l a z i z, S. L u. Improved Salp Swarm Algorithm Based on Particle Swarm Optimization for Feature Selection. – Journal of Ambient Intelligence and Humanized Computing, Vol. **10**, 2019, No 8, pp. 3155-3169. DOI: 10.1007/s12652-018-1031-9.
16. N a d i m i-S h a h r a k i, M. H., H. Z a m a n i, S. M i r j a l i l i. Enhanced Whale Optimization Algorithm for Medical Feature Selection: A COVID-19 Case Study. – Computers in Biology and Medicine, Vol. **148**, 2022. DOI: 10.1016/j.compbiomed.2022.105858.
17. S u n, L., S. S i, J. Z h a o, J. X u, Y. L i n, Z. L v. Feature Selection Using Binary Monarch Butterfly Optimization. – Applied Intelligence, Vol. **53**, 2023, No 1, pp. 706-727. DOI: 10.1007/s10489-022-03554-9.
18. A b d e l-B a s s e t, M., K. M. S a l l a m, R. M o h a m e d, I. E l g e n d i, K. M u n a s i n g h e, O. M. E l k o m y. An Improved Binary Grey-Wolf Optimizer with Simulated Annealing for Feature Selection. – IEEE Access, Vol. **9**, 2021, pp. 139792-139822. DOI: 10.1109/ACCESS.2021.3117853.
19. J a i n, S., K. K. B h a r t i. Chapter 18 – Genome Sequence Assembly Using Metaheuristics. – In: S. Mirjalili, A. H. Gandomi, Eds. Comprehensive Metaheuristics. Academic Press, 2023, pp. 347-358. DOI: 10.1016/B978-0-323-91781-0.00018-1.
20. T o o, J., A. R. A b d u l l a h, N. M. S a a d, N. M. A l i, W. T e e. A New Competitive Binary Grey Wolf Optimizer to Solve the Feature Selection Problem in EMG Signals Classification. – Computers, Vol. **7**, 2018, No 4. DOI: 10.3390/computers7040058.
21. K e n n e d y, J., R. E b e r h a r t. Particle Swarm Optimization. – In: Proc. of International Conference on Neural Networks (ICNN'95), Vol. **4**, 1995. pp. 1942-1948. DOI: 10.1109/ICNN.1995.488968.
22. X i e, H., L. Z h a n g, C. P. L i m, Y. Y u, H. L i u. Feature Selection Using Enhanced Particle Swarm Optimisation for Classification Models. – Sensors, Vol. **21**, 2021, No 5, pp. 1-40. DOI: 10.3390/s21051816.
23. D u, K.-L., M. N. S. S w a m y. Particle Swarm Optimization. – In: K.-L. Du, M. N. S. Swamy, Eds. Search and Optimization by Metaheuristics: Techniques and Algorithms Inspired by Nature. Springer International Publishing, 2016, pp. 153-173. DOI: 10.1007/978-3-319-41192-7_9.
24. M o h a m m e d, H., T. R a s h i d. FOX: A FOX-Inspired Optimization Algorithm. – Applied Intelligence, Vol. **53**, 2023, No 1, pp. 1030-1050. DOI: 10.1007/s10489-022-03533-0.
25. T o o, J., A. R. A b d u l l a h, N. M. S a a d. A New Co-Evolution Binary Particle Swarm Optimization with Multiple Inertia Weight Strategy for Feature Selection. – Informatics, Vol. **6**, 2019, No 2. DOI: 10.3390/informatics6020021.
26. T o o, J., A. R. A b d u l l a h, N. M. S a a d, W. T e e. EMG Feature Selection and Classification Using a Pbest-Guide Binary Particle Swarm Optimization. – Computation, Vol. **7**, 2019, No 1. DOI: 10.3390/computation7010012.
27. Y u r i y, R., L. V i a t c h e s l a v, B. Z i v. Advanced PSO Algorithms Development with Combined Lbest and Gbest Neighborhood Topologies. – Cybernetics and Information Technologies, Vol. **24**, 2024, No 3, pp. 59-77.
28. A k o p o v, A. S. A Hybrid Multi-Swarm Particle Swarm Optimization Algorithm for Solving Agent-Based Epidemiological Model. – Cybernetics and Information Technologies, Vol. **25**, 2025, No 4, pp. 59-77.
29. UCI Irvine Machine Learning Repository. **https://archive.ics.uci.edu/**

Fast-track. Received: 28.12.2025, Revised version: 06.02.2026, Accepted: 18.02.2026