Heliyon 10 (2024) e24192

Contents lists available at ScienceDirect

Heliyon

journal homepage: www.cell.com/heliyon

Research article

S-shaped grey wolf optimizer-based FOX algorithm for feature selection

Afi Kekeli Feda [a], Moyosore Adegboye [b], Oluwatayomi Rereloluwa Adegboye [c], Ephraim Bonah Agyekum [d], Wulfran Fendzi Mbasso [e,*], Salah Kamel [f]

[a] *Management Information System Department, European University of Lefke, Mersin, 10, Turkey*
[b] *School of Computing and Information, University of Pittsburgh, USA*
[c] *Management Information Systems, University of Mediterranean Karpasia, Mersin, 10, Turkey*
[d] *Department of Nuclear and Renewable Energy, Ural Federal University named after the first President of Russia Boris Yeltsin, 620002, 19 Mira Street, Ekaterinburg, Russia*
[e] *Laboratory of Technology and Applied Sciences, University Institute of Technology, University of Douala, PO Box: 8698, Douala, Cameroon*
[f] *Department of Electrical Engineering, Faculty of Engineering, Aswan University, Aswan, 81542, Egypt*

ARTICLE INFO

Keywords:
Feature selection
S-Shaped transfer function
FOX algorithm

ABSTRACT

The FOX algorithm is a recently developed metaheuristic approach inspired by the behavior of foxes in their natural habitat. While the FOX algorithm exhibits commendable performance, its basic version, in complex problem scenarios, may become trapped in local optima, failing to identify the optimal solution due to its weak exploitation capabilities. This research addresses a high-dimensional feature selection problem. In feature selection, the most informative features are retained while discarding irrelevant ones. An enhanced version of the FOX algorithm is proposed, aiming to mitigate its drawbacks in feature selection. The improved approach referred to as S-shaped Grey Wolf Optimizer-based FOX (FOX-GWO), which focuses on augmenting the local search capabilities of the FOX algorithm via the integration of GWO. Additionally, the introduction of an S-shaped transfer function enables the population to explore both binary options throughout the search process. Through a series of experiments on 18 datasets with varying dimensions, FOX-GWO outperforms in 83.33 % of datasets for average accuracy, 61.11 % for reduced feature dimensionality, and 72.22 % for average fitness value across the 18 datasets. Meaning it efficiently explores high-dimensional spaces. These findings highlight its practical value and potential to advance feature selection in complex data analysis, enhancing model prediction accuracy.

1. Introduction

In the contemporary landscape of data-driven endeavors, the explosion of information has brought forth vast datasets replete with numerous features [1,2]. While this proliferation of data holds great promise, it also presents a formidable challenge of how to distinguish information from noise and relevant from irrelevant [3]. This fundamental challenge has propelled the field of feature selection into focus as it seeks to address these issues by identifying and retaining the most meaningful attributes while discarding the

* Corresponding author.
E-mail addresses: kekelifeda@gmail.com (A.K. Feda), Moa@pitt.edu (M. Adegboye), aoluwatayomi@gmail.com (O.R. Adegboye), agyekumephraim@yahoo.com (E.B. Agyekum), fendzi.wulfran@yahoo.fr (W. Fendzi Mbasso), skamel@aswu.edu.eg (S. Kamel).

https://doi.org/10.1016/j.heliyon.2024.e24192
Received 6 September 2023; Received in revised form 9 December 2023; Accepted 4 January 2024
Available online 5 January 2024
2405-8440/

rest [4–8]. Feature selection (FS) is not merely a desirable option but a necessity in data analysis and machine learning. Its significance cannot be overstated, as it underpins the efficacy of predictive models in various fields like Nano fluids [9], Neural Network Applications [10,11], and Space Exploration [12]. The elimination of pointless and duplicate features results in an adequate text portrayal and a diminution in data dimensionality, which accelerates the model's learning process and improves the predictive model's effectiveness and accuracy [13]. Primarily, there are two principal types of feature selection techniques: filter and wrapper methods [14–16]. The wrapper technique has shown its superiority to the filter-based approach in classification tasks by successfully resolving real-world problems. However, its execution time is longer due to the need for repeated calls to the learning algorithm [17,18]. While the need for feature selection is clear, and it can be viewed as an optimization problem as it aims to find the optimal subset of features, the process is far from straightforward, particularly in high-dimensional spaces. Employing precise search strategies in this domain becomes impractical since they generate all possible solutions in order to obtain the single best solution [19]. Hence, there is a need for a powerful stochastic algorithm to cut down on the computing time that this sort of problem requires. Metaheuristic algorithms have demonstrated their effectiveness in various domains by providing practical solutions within a reasonable timeframe. They are particularly valuable in addressing the problem of dimensionality, as they optimize classification performance while reducing computational resource usage, storage requirements, and the number of features. The versatility and efficiency of metaheuristic-based algorithms make them well-suited for a wide range of applications [20,21]. A few examples of well-known metaheuristic algorithms are the Genetic Algorithm (GA) [22], Aquila Optimizer (AO) [23], Whale Optimization Algorithm (WOA) [24], and Grey Wolf Optimizer (GWO) [25]. There has been a considerable number of works conducted by researchers to use metaheuristic approaches to solve the FS problems [26–29]. Basic Metaheuristic Algorithms have limitations when addressing complex optimization problem, and their performance can be hampered by issues such as local optima and limited exploration of the feature reduction solution space [30, 31]. As feature selection problems are becoming more common, there is a high demand for advanced algorithms capable of generating high-quality solutions for the given problem. Nevertheless, according to the No Free Lunch (NFL) theorem, no single algorithm can produce the best outcomes for all optimization problems [32,33]. Therefore this research introduces S-Shaped hybrid binary FOX optimizer by introducing a Grey Wolf Optimization Algorithm (FOX-GWO) algorithm, it's a novel and original solution to the challenges of feature selection a significant departure from conventional methods. It merges the FOX algorithm with GWO and introduces an S-Shaped Transfer function. The Fox optimizer (FOX), a new optimization algorithm that imitates the hunting patterns of foxes in their natural environment its a desirable candidate in this research due to its efficiency proven in its comparison to other popular metaheuristic algorithms and its successful handling of real-world engineering problems [37]. However, like other metaheuristic algorithms, the FOX suffers from susceptibility to local optima and weak exploitation limitations when applied to high-dimensional problems. The novelty of our approach lies firstly in its capacity to enhance both exploration and exploitation of FOX via the fusion of GWO algorithm within the feature space reduction problem different from existing FS algorithm in the literature, thereby addressing the limitations of the basic FOX algorithm and secondly an S-shaped transfer function to improve the conversion of the population to binary making it suitable for FS problems. The S-Shaped Transfer Function is a mathematical equation that facilitates the transformation of continuous numerical values into binary forms. In high-dimensional datasets, where there are numerous features, the possibilities for combinations and permutations of different features can be overwhelming. The S-Shaped Transfer function simplifies this complex landscape. It effectively converts the continuous values associated with these features into a binary format, where each feature is either included (1) or excluded (0) in the final selection. The "S-Shaped" nature of this function signifies that it mimics the shape of the letter "S." It's designed to be sensitive to changes in the continuous values of features. As a result, even small variations in the continuous values can lead to significant changes in the binary representation. This sensitivity allows the FOX-GWO to make informed decisions about the inclusion or exclusion of features. The uniqueness of the FOX-GWO algorithm resides in its ability to balance accuracy and efficiency, overcoming challenges that have previously beset feature selection algorithms. FOX-GWO demonstrated higher accuracy than other optimizers in 83.33 % of the datasets, selected the fewest number of features in 61.11 % of the datasets, and achieved the best average fitness value in 72.22 % of the datasets. These results suggest that FOX-GWO offers an improved solution for selecting the most relevant features. Below are contributions of this work:

1 The manuscript introduces the FOX-GWO algorithm, a Binary Hybrid novel approach that combines the FOX algorithm with GWO to address the limitation of the basic FOX algorithm.
2 The S-shaped transfer function is introduced into the FOX-GWO hybrid approach for efficient continuous value to binary value conversion for a more accurate selection of features.
3 FOX-GWO is compared to five other state-of-the-art optimizers on feature selection results. FOX-GWO demonstrates superior performance compared to other optimization algorithms across various metrics, including fitness values, feature selection, prediction accuracy, and computational efficiency, thus offering a robust solution to optimization problems.

The remainder of this work is structured as follows: The next section presents a literature review on FS metaheuristic algorithms. The traditional FOX algorithm is presented in Section 3, and Section 4 discusses the novel S-shaped Binary Grey Wolf Optimizer-based FOX (FOX-GWO) Algorithm. The experimental findings are discussed in Section 5, along with an analysis of the findings. Section 6 concludes by providing recommendations for future work.

2. Related works

In many domains recently, high-dimensional feature selection issues have been solved using optimization methods. Among those is the Chaotic Dragonfly Algorithm (CDA) [34] introduced by Sayed et al. The algorithm combines chaotic maps with the Dragonfly

Algorithm (DA) to enhance its convergence rate and efficiency. The study utilizes a dataset extracted from the Drug Bank database and consists of three main stages: data pre-processing, feature selection using CDA, and classification using Support Vector Machine (SVM). The outputs from the experiment demonstrate that CDA outperforms DA and other meta-heuristic optimization algorithms in terms of maximizing classification performance while minimizing the number of chosen features. The Gauss chaotic map is found to be particularly effective in boosting the effectiveness of DA. The proposed model exhibits robustness, as indicated by high accuracy, recall, precision, and F-Score values for all toxic effects. Taghian and Nadimi-Shahraki proposed the S-shaped binary Sine Cosine Algorithm (SBSCA) and the V-shaped binary Sine Cosine Algorithm (VBSCA) for feature selection in medical data [35]. These algorithms maintain a continuous search area but generate binary position vectors using S-shaped and V-shaped transfer functions for every single solution. The two algorithms are evaluated against four contemporary binary optimization algorithms using five medical datasets. The results show that, in comparison to the other four algorithms, SBSCA and VBSCA increase the classification accuracy of these medical datasets. Another algorithm applied to feature selection is the Modified Marine Predators Algorithm with sine cosine algorithm (MPASCA) proposed by Elaziz et al. [36]. In this novel approach, the authors incorporate the sine-cosine algorithm (SCA) to enhance the search capability by effectively acting as a local search within the MPA. To assess the effectiveness of the MPASCA algorithm, authors conducted experiments on 18 UCI datasets. Additionally, they utilized a metabolomics dataset to evaluate the suggested algorithm in practical situations. Furthermore, they conducted comprehensive comparisons with different popular algorithms to validate the performance of MPASCA. The results clearly demonstrate that MPASCA exhibits significant performance improvements and outperforms the compared algorithms. The Chaotic Vortex Search Algorithm (CVSA) [37], proposed by Gharehchopogh et al. is the result of the integration of chaos theory into the Vortex Search Algorithm (VSA) search process to improve its performance and accelerate overall convergence. The suggested approach uses a variety of chaotic maps to improve VSA operators and strike a balance between exploration and exploitation. Using 24 UCI standard datasets, the efficacy of this technique was assessed. The outcomes of the experiment showed that chaotic maps, especially the Tent map, considerably enhanced VSA performance. Additionally, the suggested approach produced the smallest number of features and the highest accuracy optimal feature subsets. Further evidence that the proposed method surpassed other algorithms in terms of accuracy percentage came from the outcomes of real-world applications. The Quantum Whale Optimization Algorithm (QWOA) proposed by Agrawal et al. [38] is a novel meta-heuristic algorithm that combines elements from Quantum Concepts and the Whale Optimization Algorithm (WOA) to improve its exploratory and exploitation capabilities. This is achieved through the utilization of quantum bit representation of search agents and the incorporation of a quantum rotation gate operator. Additionally, to support the quantum-based exploration, shrinkage, and spiral movement of the whales, modified mutation and crossover operators were included. By contrasting the novel QWOA with the traditional WOA and other algorithms over fourteen different datasets, the effectiveness of the QWOA was assessed. The QWOA approach performs better than other methods, as shown by the testing findings. Further statistical testing shows that the QWOA greatly outperforms eight popular meta-heuristic algorithms. Ma et al. introduced a Two-Stage Hybrid Feature Selection Ant Colony Optimization (TSHFS-ACO) as a response to the challenges faced by traditional Ant Colony Optimization (ACO) while dealing with big datasets [39]. The dimension of the optimal feature subset (OFS) for the following OFS search is decided using an interval method in the additional step. By assessing the performance of fragmentary feature numbers beforehand, the suggested strategy decreases the complexity of the algorithm and prevents local optima, in contrast to conventional one-stage methods that simultaneously identify the dimension of the OFS and seek for it. The improved ACO method also includes a hybrid model that directs the OFS search using the classification performance and relevance attributes of feature attributes. The TSHFS-ACO algorithm produces OFSs with cutting-edge performance on the majority of high-dimensional datasets, as shown by tests conducted on eleven high-dimensional datasets. In addition, TSHFS-ACO runs more quickly than other ACO-based algorithms applied for feature selection. The Binary Crow Search Algorithm with Time-Varying Flight Length (BCSA-TVFL) [40], was introduced by Abudullahi et al. The proposed algorithm was obtained by introducing five new types of flight length techniques in the Crow Search Algorithm namely: linearly decreasing; sigmoid decreasing; chaotic decreasing; simulated annealing lowering; and logarithm decreasing. To assess the effectiveness of the provided techniques, thirteen popular UCI datasets are employed. The outcomes of the simulation prove that the proposed methods perform better than the traditional CSA.

3. Original FOX algorithm

This section introduces the original Fox Optimizer (FOX), a novel optimization algorithm that takes inspiration from nature and mimics the hunting behavior of foxes. To start, FOX generates the population as a X matrix, where each X represents the location of a fox. The population is then examined in the first iteration to determine whether or not each red fox's location is within the benchmark function's border. The benchmark function's fitness value is then computed based on the population's row. These actions are followed by choosing the BestFitness value and the BestX location. Then, by analyzing the value of the number r randomly generated, a condition is started. When it exceeds or equals 0.5, the exploitation phase is initiated.

3.1. Exploitation phase

During this phase, a condition that determines the likelihood of prey capture is implemented. A variable p is randomly selected from the interval $[0, 1]$. If p exceeds 0.18, the position of the fox is upgraded by determining the distance the sound travels, denoted as $Dist\_S\_T_{it}$, the distance separating the prey from the fox $Dist\_Fox\_Prey\_$ and the jumping value $Jump_{it}$. To do so, the sound travel time, $Time\_S\_T_{it}$ is represented by a number randomly generated from $[0, 1]$. The sound's distance from the red fox is then calculated as in Equation (1):

Algorithm 1. Steps of FOX

```
Input: number of individuals (N), number of iterations (Maxit), and parameters of FOX
(Time_S_T_it, Sp_S, Time S_T_it, Best X, Dist_Fox_Prey, Jump, MinT, a,
BestFitness)
Compute the fitness value for each X_i.
Find the BestX in the population
it = 1
While (it < Maxit)
If r >= 0.5
        If p > 0.18
                Initialize time randomly;
                Calculate jump using Eq. 4
                Find X_(it+i) using Eq.5
        Elseif p <= 0.18
                Calculate jump using Eq. 4
                Find X_(it+i) using Eq. 6
        EndIf
  Else
        Find MinT using Eq. 7
        Find X_(it+i) using Eq. 9
  EndIf
Find the best individual.
it = it + 1
EndWhile
```

Fig. 1. Steps of the fox.

$$Dist\_S\_T_{it} = Sp\_S * Time\_S\_T_{it}, \tag{1}$$

$$S_p\_S = \frac{B_{\text{estPosition}}}{T_{\text{ime}}\_S\_T_{\text{it}}} \tag{2}$$

where $Time\_S\_T_{it}$ is randomly selected from the range [0, 1], the sound's speed via medium denoted by S_p_S is 343, $B_{\text{estPosition}}$ is the best position till that moment Equation (2), and *it* denotes the present number of iterations. The distance from the fox to the prey is determined by Equation (3).

$$Dist\_Fox\_Prey\_ = Dist\_S\_T_{it} * 0.5 \tag{3}$$

Subsequently, the height of the jump to catch the prey is calculated by Equation (4).

$$Jump_{it} = 0.5 * 9.81 * t^2 \tag{4}$$

The value 9.81 represents the acceleration caused by gravity. The variable t corresponds to sound travel average time and is squared to account for the vertical movement during the jump. t is obtained by dividing *it* by 2. If the value of p exceeds 0.18, the direction of Fox's jumps is toward the northeast, and the following equation is used to calculate the novel position Equation (5):

$$X_{(it+1)} = Dist\_Fox\_Prey_{it} * \; Jump_{it} * c_1, \tag{5}$$

Else, the fox jumps toward the opposite direction, and the new position is calculated as shown in Equation (6):

$$X_{(it+1)} = Dist\_Fox\_Prey_{it} * \; Jump_{it} * c_2 \;, \tag{6}$$

where, c_1 and c_2 are respectively selected randomly from the intervals $[0, 0.18]$ and $[0.19, 1]$.

3.2. Exploration phase

In order to regulate the random movement, the fox conducts a random search during this phase, taking into account the best position discovered thus far. The minimal time variable *MinT* and variable *a* are used as the search control variables to make sure that the fox wanders randomly in the direction of the global optimum. The steps involved in calculating *MinT* and the control variable are described in Equations (7) and (8).

$$tt = \frac{sum\ (Time\_S\_T_{it}(i,:))}{dimension}, MinT = Min(tt), \tag{7}$$

$$a = 2^{*}\left(it - \left(\frac{1}{Max_{it}}\right)\right) \tag{8}$$

with Max_{it} denoting the maximum iteration. Equation (9) illustrates fox's search strategy as it moves across the search area looking for a new location.

$$X_{(it+1)} = B_{est}X_{it} * rand(1, dimension\ ) * MinT * a, \tag{9}$$

The exploration and exploitation stages are also balanced using the random number variable r, as seen in Fig. 1.

4. Proposed S-shaped FOX-GWO feature selection methodology

In this section, the proposed FOX-GWO algorithm's components, including the S-shaped transfer function, Grey Wolf Optimization (GWO) operators, binary conversion process, fitness function, and other essential elements, will be elaborated upon in detail to provide a thorough insight into its operational framework and capabilities of the proposed method.

4.1. Grey Wolf Optimization algorithm

Grey Wolf Optimization (GWO) was first introduced by Mirjalili et al. [25], and it draws inspiration from the social behavior observed in grey wolves, which tend to live in groups consisting of 5–12 individuals. Alpha, beta, delta, and omega are the algorithm's variables, which imitate the GWO leadership hierarchy. The alpha wolves, who stand for both the male and female pack leaders, are in charge of choosing things like where to hunt, where to sleep, and when to get up in the morning. The beta wolves support the alphas by participating in decision-making and mainly offering suggestions and criticism. The delta wolves serve as hunters, sentinels, elders, caregivers, and scouts. To ensure the omega wolves' submission to the alpha and beta wolves, delta wolves are in charge of managing them. All other wolves in the GWO pack follow the alpha, beta, and delta wolves' lead when it comes to hunting. The following equation can be used to compute the GWO's prey-encircling behavior Equation (10):

$$\overrightarrow{\mathrm{X}}(\mathrm{t}+1) = \overrightarrow{\mathrm{X}}_p(t) + \overrightarrow{\mathrm{A}} \cdot \overrightarrow{\mathrm{D}}, \tag{10}$$

when $\overrightarrow{A}$, $\overrightarrow{C}$ represent coefficient vectors, $\overrightarrow{X}_p$ represents the vector of positions of the prey, and X represents the positions of wolves in a d-dimensional space, where d denotes the number of variables. (t) Represents the iteration number and $\overrightarrow{D}$, $\overrightarrow{A}$, $\overrightarrow{C}$ are defined as follows Equation (11) (12) and (13):

$$\overrightarrow{\mathrm{D}} = \left|\overrightarrow{\mathrm{C}} \cdot \overrightarrow{\mathrm{X}}_p(t) - \overrightarrow{\mathrm{X}}(t)\right| \tag{11}$$

$$\overrightarrow{\mathrm{A}} = 2\overrightarrow{a} \cdot \overrightarrow{r_1} - \overrightarrow{a} \tag{12}$$

$$\overrightarrow{\mathrm{C}} = 2 \cdot \overrightarrow{r_2} \tag{13}$$

where $\overrightarrow{r_1}, \overrightarrow{r_2}$ are vectors arbitrarily selected from the interval [0,1]. The vector $\overrightarrow{\mathrm{a}}$ gradually decreases linearly from 2 to 0 with iterations. During the hunting, the alpha wolf is seen to be the best candidate for identifying the global optimal. The beta and delta wolves are supposed to be aware of where the targeted prey might be. As a result, the top three solutions identified up to a certain iteration are kept, while other wolves are forced to move to the best location in the decision space. The following calculation can be used for updating positions Equation (14):

$$\overrightarrow{\mathrm{X}}(t+1) = \frac{\overrightarrow{x_1 + x_2 + x_3}}{3} \tag{14}$$

The values of x_1, x_2 and x_3 are defined and calculated as follows Equations (15)–(17):

$$\overrightarrow{x}_1 = \overrightarrow{X}_\alpha - A_1 \cdot \left(\overrightarrow{D}_\alpha\right), \tag{15}$$

$$\overrightarrow{x}_2 = \overrightarrow{X}_\beta - A_2 \cdot \left(\overrightarrow{D}_\beta\right), \tag{16}$$

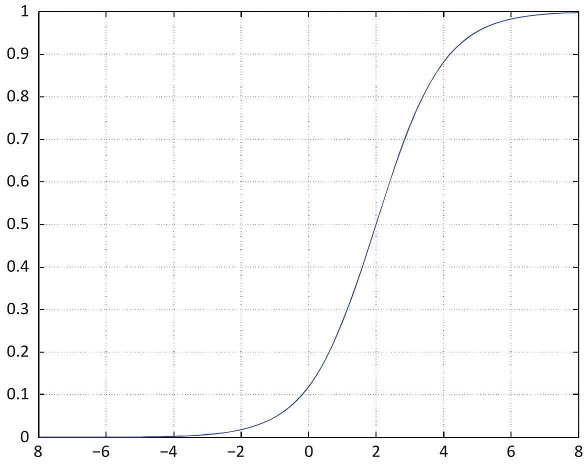

Fig. 2. S-shaped transfer function.

$$\overrightarrow{x_3} = \overrightarrow{X}_\delta - A_3 \cdot \left(\overrightarrow{D}_\delta\right), \tag{17}$$

where $\overrightarrow{x_1}$, $\overrightarrow{x_2}$ and $\overrightarrow{x_3}$ represent the positions of the three best solutions at iteration t. Here, A_1, A_2 and A_3 are obtained as described in Equation (12). $\overrightarrow{D}_\alpha, \overrightarrow{D}_\beta, \overrightarrow{D}_\delta$ are calculated according to Equations (18)–(20):

$$\overrightarrow{D_\alpha} = \left|\overrightarrow{C}_1 \cdot \overrightarrow{X}_\alpha - \overrightarrow{X}\right|, \tag{18}$$

$$\overrightarrow{D}_\beta = \left|\overrightarrow{C}_2 \cdot \overrightarrow{X}_\beta - \overrightarrow{X}\right|, \tag{19}$$

$$\overrightarrow{D}_\delta = \left|\overrightarrow{C}_3 \cdot \overrightarrow{X}_\delta - \overrightarrow{X}\right|, \tag{20}$$

where $\overrightarrow{C}_1$, $\overrightarrow{C}_2$ and $\overrightarrow{C}_3$ where obtained in Equation (13). In the Grey Wolf Optimization (GWO) algorithm, the vector $\overrightarrow{a}$ is essential for maintaining a balance between exploration and exploitation. As iterations go on, $\overrightarrow{a}$ is linearly decreased for each dimension from 2 to 0. Equation (21) is used to update $\overrightarrow{a}$:

$$\overrightarrow{a} = 2 - t \cdot \frac{2}{max_{iter}}, \tag{21}$$

The current iteration is represented by t as the iteration, and the maximum iteration by max_{iter}.

4.2. S-shaped transfer function

The solutions to Feature Selection (FS) problems are constrained to binary values {0, 1}. Therefore, to address FS problems, it becomes necessary to translate the continuous positions into their equivalent binary solutions, 0 and 1. An S-shaped transfer function is used to achieve the conversion. The transfer functions have adopted the likelihood of converting the position vectors' constituent elements from 0 to 1 and vice versa, forcing the individuals in the population to navigate in a binary search area (Fig. 2). The transfer function has proven to be an efficient conversion function in several algorithms [41]. Equation (22) depicts the common S-shaped function update. Equation (23) is used in the estimation.

$$y^k = \frac{1}{1 + e^{-x_i^t(u)}} \tag{22}$$

$$x_i^d = \begin{cases} 1 \text{ if rand } < S\left(x_i^t(t+1)\right), \\ 0 \text{ otherwise} \end{cases} \tag{23}$$

The structure of the Feature Selection (FS) technique developed in this research relies on enhancing the efficiency of FOX by incorporating GWO operators and an S-shaped transfer function, as shown in Fig. 3. The primary objective of employing GWO is to amplify FOX's exploitation capability, which greatly impacts its potential to uncover the optimal solutions within the feasible region

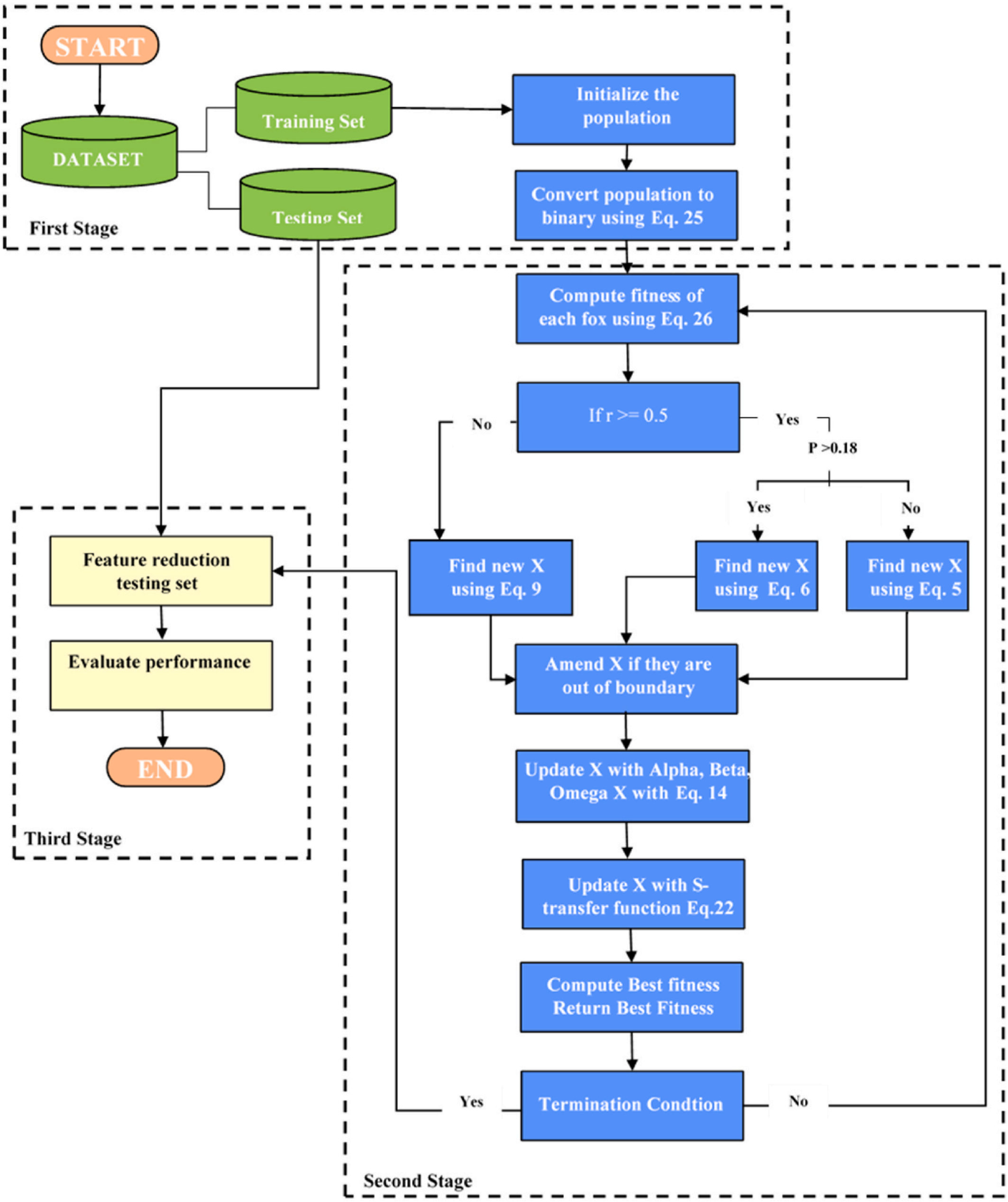

Fig. 3. Proposed S-shaped FOX-GWO

and improve binary conversion by the use of an S-shaped function. Referred to as FOX-GWO, the proposed FS approach initiates the search operation by partitioning the data into testing and training 30 % of the data is used for testing and 70 % for Training; subsequently, the population is initialized, and each individual is converted into binary form, followed by the computation of their fitness values. The individual with the highest fitness value is selected as the best in the population. Following this procedure is updating the solution through the utilization of FOX operators during the exploration and exploitation phase while also incorporating GWO operators to further enhance exploitation performance. The update process of individuals continues until the specified termination conditions are met. Afterward, the best features are selected from testing data using the information given by the best fox in the populace, and the effectiveness of FOX-GWO as a feature selection method is evaluated using various metrics. Further elaboration on the intricacies of FOX-GWO is provided in the subsequent paragraphs.

In the First stage, as illustrated in Fig. 3, the population of prospective solutions is established by generating the initial individuals. The process for creating these individuals is outlined as follows in Equation (24): The upper boundary UB_j and lower boundary LB_j values are defined for each dimension j th. N denotes the number of individuals in the population, while D represents the dimension,

Algorithm 2. Steps of FOX-GWO

1 Input: the dataset which has D features, number of individuals (N), number of iterations (Maxit), and parameters of FOX-GWO ($Time\_S\_T_{it}$, S_p_S , $Time\ S\_T_{it}$, $Best\ X$, $Dist\_Fox\_Prey$, $Jump$, $MinT$, a, $BestFitness$)

First Stage

2 Split data into two parts (Training and Testing)

3 Construct the population X using Equation (24).

4 Convert each X_i into its binary version using Equation (25).

5 Compute the fitness value for each X_i based on the training set as in Equation (26).

6 Find the three best individuals Alpha, Beta, Omega (X_1, X_2, and X_3)

Second Stage

7 $it = 1$

8 While ($it <$ Maxit)

9 If r >= 0.5

10   If p > 0.18

11   Initialize time randomly;

12   Calculate jump using Eq. 4

13   Find $X_{(it+i)}$ using Eq.5

14   Elseif p <= 0.18

15   Calculate jump using Eq. 4

16   Find $X_{(it+i)}$ using Eq. 6

17   EndIf

18   Else

19   Find MinT using Eq. 7

20   Find $X_{(it+i)}$ using Eq. 9

21   EndIf

22 Update $X_{(it+i)}$ with X_1, X_2, and X_3 using Eq. (14)

23 Update $X_{(it+i)}$ with S-transfer function using Eq. (22)

24 Find the best individual.

25 $it = \text{it} + 1$

26 EndWhile

Third Stage

27 Reduce the testing set based on selected features based on the best individual.

28 Evaluate the performance using different measures

Fig. 4. Steps of FOX-GWO

signifying the total number of features depending on the dataset. Moreover, a random number rand is selected from the range of [0, 1].

$$X_{i,j} = \left(UB_j - LB_j\right) \times \text{rand} + UB_j - LB_j, i = 1, 2, \ldots, N, j = 1, 2, \ldots, D, \tag{24}$$

The primary objective of this phase within the newly developed FOX-GWO is to continuously improve search agents via several operators until the specified termination conditions are met. This objective is accomplished through a series of steps, with the next step

Table 1
Data set description.

Datasets	Instances	Features
Breastcancer	699	9
BreastEW	569	30
Congress	435	16
Exactly	1000	13
Exactly2	1000	13
HeartEW	270	13
Ionosphere	351	34
KrVsKpEW	3196	36
Lymphography	148	18
M_of_n	1000	13
PenglungEW	73	325
Sonar	208	60
SpectEW	267	22
Tic_tac_toe	958	9
Vote	300	16
WaveformEW	5000	40
Wine	178	13
Zoo	101	16

Table 2
Average fitness value of FOX-GWO and other optimizers.

Datasets	AEFA	FFA	GWO	PSO	FOX	FOX-GWO
Breastcancer	3.710E-02	4.364E-02	4.039E-02	4.103E-02	3.441E-02	**3.024E-02**
BreastEW	7.187E-02	7.296E-02	6.565E-02	7.195E-02	6.377E-02	**5.722E-02**
Congress	**4.117E-02**	7.314E-02	4.399E-02	5.561E-02	6.154E-02	4.358E-02
Exactly	3.076E-01	3.164E-01	3.099E-01	3.171E-01	3.007E-01	**1.934E-01**
Exactly2	2.561E-01	2.481E-01	**2.403E-01**	2.511E-01	2.425E-01	2.414E-01
HeartEW	2.324E-01	2.598E-01	2.450E-01	2.787E-01	2.059E-01	**1.741E-01**
Ionosphere	**1.077E-01**	1.503E-01	1.159E-01	1.501E-01	1.497E-01	1.356E-01
KrVsKpEW	9.984E-02	1.701E-01	1.707E-01	1.471E-01	1.024E-01	**4.822E-02**
Lymphography	1.932E-01	2.570E-01	2.003E-01	2.365E-01	1.994E-01	**1.600E-01**
M_of_n	1.711E-01	2.173E-01	1.922E-01	1.978E-01	1.699E-01	**6.699E-02**
PenglungEW	1.479E-01	1.998E-01	1.540E-01	**1.297E-01**	1.603E-01	1.448E-01
Sonar	2.123E-01	2.617E-01	2.089E-01	**2.008E-01**	2.298E-01	2.033E-01
SpectEW	1.688E-01	1.935E-01	1.719E-01	1.795E-01	1.791E-01	**1.505E-01**
Tic_tac_toe	2.575E-01	2.908E-01	2.692E-01	2.776E-01	2.573E-01	**2.089E-01**
Vote	6.995E-02	8.150E-02	6.384E-02	6.994E-02	6.065E-02	**4.842E-02**
WaveformEW	2.124E-01	2.336E-01	2.266E-01	2.173E-01	2.178E-01	**1.858E-01**
Wine	1.330E-01	1.795E-01	9.275E-02	1.887E-01	1.275E-01	**4.974E-02**
Zoo	9.746E-02	1.523E-01	1.290E-01	1.144E-01	1.019E-01	**4.467E-02**

involving the conversion of every search agent (Fox) $X_{i,j}$ into a binary as given Equation (25).

$$bX_{i,j} = \begin{cases} 1 \text{ if } X_{i,j} \geq 0.5 \\ 0 \text{ otherwise} \end{cases} \tag{25}$$

Subsequently, in the Second Stage, features that are ones in $bX_{i,j}$ are utilized in order to train the “KNN classifier”, next is the computation of the fitness value of every search agent:

$$\text{fitness} = \alpha\rho_R(D) + \beta\frac{|S|}{|T|} \tag{26}$$

The feature selection problem inherently possesses a bi-objective nature, encompassing two distinct objectives. The first objective aims to minimize the number of features, while the second objective focuses on maximizing the classification accuracy. In order to account for both objectives, the following equation is employed as a fitness function. In Equation (26), α represents a parameter ranging from 0 to 1, and β is its complement $(1 - \alpha)$. The term $\rho_R(D)$ denotes the “error rate of the classifier”. Furthermore, $|S|$ represents the selected features from the total number of features $|T|$. Following the depicted sequence in Fig. 2, the subsequent steps involve utilizing the FOX operators to determine the new positions of the foxes. This is achieved by employing Equations (5), (6) and (9). Furthermore, the next step focuses on enhancing the exploitation ability of the Foxes through the incorporation of the Alpha, Beta, and Omega wolves' information-sharing and collaboration mechanism from GWO. Specifically, the three best Foxes (Alpha, Beta, and Omega) that have been discovered up to a specific iteration are retained and represented as $\vec{X}_\alpha$, $\vec{X}_\beta$, and $\vec{X}_\delta$. This information is utilized to update the positions of the Foxes, as illustrated by Equation (14), and afterward, the S-shaped transfer function is applied to

Table 3
Worst fitness value of FOX-GWO and other optimizers.

Datasets	AEFA	FFA	GWO	PSO	FOX	FOX-GWO
Breastcancer	5.586E-02	6.778E-02	5.536E-02	5.536E-02	4.859E-02	**3.950E-02**
BreastEW	9.398E-02	1.061E-01	9.826E-02	9.218E-02	8.437E-02	**7.761E-02**
Congress	5.966E-02	1.210E-01	1.354E-01	1.049E-01	8.924E-02	**5.950E-02**
Exactly	3.452E-01	3.519E-01	3.444E-01	3.808E-01	3.587E-01	**2.611E-01**
Exactly2	2.806E-01	2.890E-01	**2.403E-01**	2.715E-01	2.842E-01	2.493E-01
HeartEW	3.346E-01	3.690E-01	3.022E-01	3.802E-01	3.014E-01	**2.107E-01**
Ionosphere	**1.533E-01**	2.128E-01	1.634E-01	1.960E-01	1.806E-01	1.592E-01
KrVsKpEW	2.439E-01	3.372E-01	2.576E-01	2.940E-01	1.939E-01	**5.728E-02**
Lymphography	2.475E-01	3.266E-01	2.425E-01	3.121E-01	2.570E-01	**2.179E-01**
M_of_n	2.573E-01	2.957E-01	2.597E-01	3.056E-01	2.347E-01	**1.039E-01**
PenglungEW	2.433E-01	2.729E-01	2.427E-01	**2.145E-01**	2.186E-01	2.201E-01
Sonar	2.804E-01	3.347E-01	2.762E-01	**2.386E-01**	3.094E-01	2.620E-01
SpectEW	2.036E-01	2.078E-01	2.073E-01	2.110E-01	2.073E-01	**1.892E-01**
Tic_tac_toe	3.291E-01	3.442E-01	3.442E-01	3.588E-01	2.874E-01	**2.310E-01**
Vote	1.417E-01	1.304E-01	1.590E-01	1.027E-01	1.531E-01	**6.315E-02**
WaveformEW	2.436E-01	2.645E-01	2.980E-01	2.413E-01	2.353E-01	**1.910E-01**
Wine	3.383E-01	3.717E-01	1.906E-01	3.287E-01	3.019E-01	**1.174E-01**
Zoo	1.997E-01	2.361E-01	2.730E-01	1.822E-01	1.772E-01	**8.327E-02**

Table 4
Best fitness value of FOX-GWO and other optimizers.

Datasets	AEFA	FFA	GWO	PSO	FOX	FOX-GWO
Breastcancer	2.536E-02	2.818E-02	2.424E-02	3.162E-02	1.919E-02	**1.748E-02**
BreastEW	5.082E-02	3.840E-02	4.582E-02	4.882E-02	4.582E-02	**3.346E-02**
Congress	**2.458E-02**	3.883E-02	2.787E-02	3.429E-02	3.554E-02	3.100E-02
Exactly	2.061E-01	2.972E-01	2.889E-01	2.002E-01	1.527E-01	**2.640E-02**
Exactly2	2.411E-01	2.403E-01	2.403E-01	2.403E-01	2.403E-01	**2.339E-01**
HeartEW	1.439E-01	1.864E-01	1.930E-01	1.930E-01	1.636E-01	**1.382E-01**
Ionosphere	7.371E-02	1.024E-01	**6.364E-02**	1.039E-01	1.101E-01	1.071E-01
KrVsKpEW	5.097E-02	5.518E-02	4.886E-02	4.307E-02	5.398E-02	**3.653E-02**
Lymphography	1.115E-01	1.655E-01	1.488E-01	1.522E-01	1.628E-01	**8.638E-02**
M_of_n	4.377E-02	9.525E-02	1.233E-01	2.914E-02	4.615E-03	**4.615E-03**
PenglungEW	**8.335E-02**	1.119E-01	8.522E-02	1.072E-01	1.105E-01	8.522E-02
Sonar	1.655E-01	2.113E-01	**1.338E-01**	1.535E-01	1.475E-01	1.490E-01
SpectEW	1.311E-01	1.532E-01	1.297E-01	1.297E-01	1.431E-01	**1.167E-01**
Tic_tac_toe	2.124E-01	2.515E-01	2.235E-01	2.453E-01	2.184E-01	**1.836E-01**
Vote	3.613E-02	4.210E-02	2.890E-02	2.293E-02	**2.043E-02**	2.605E-02
WaveformEW	1.907E-01	2.046E-01	1.887E-01	1.841E-01	1.951E-01	**1.764E-01**
Wine	1.651E-02	3.876E-02	**1.343E-02**	3.645E-02	2.379E-02	1.728E-02
Zoo	**5.000E-03**	8.140E-02	6.449E-02	6.261E-02	2.316E-02	2.379E-02

the search agents Equation (23). In the concluding phase, the testing set undergoes reduction by exclusively selecting the features that correspond to ones in the binary representation of $bX_{i,j}$. Subsequently, the reduced testing set is employed for prediction using the trained KNN classifier, generating output for the testing set. The subsequent step involves evaluating the output's quality through various metrics. The detailed steps of the FOX-GWO approach are presented in Fig. 4.

5. Result and discussion

The efficacy of the FOX-GWO technique is assessed across 18 datasets in this section. Furthermore, a comparative analysis is conducted between FOX-GWO and five other binary optimization algorithms, namely Artificial Electric Field Algorithm (AEFA) [42], Grey Wolf Optimizer (GWO) [43], Particle Swarm Optimization (PSO) [44], FireFly Algorithm (FFA) [45], and FOX [46]. Table 1 shows a description of the eighteen UCI datasets utilized in the experiment. It is evident from the table that these datasets are sourced from diverse applications, exhibiting distinct characteristics such as varying sample sizes and features. Each algorithm's parameters are set according to their original literature. Parameters like the number of iterations (set to 50), the number of individuals (set to 10), and the KNN classifier that utilizes Euclidean separation with (K = 5) are employed in this work. In order to ensure fairness in the comparison, each method is executed 20 times. The comparison is based on indicators like the best, average, worst, standard deviation value of the fitness function, accuracy, and the number of features selected.

In Tables 2–4, the feature selection results of all methods are presented as follows: average fitness function values of the 20 runs, the Worst fitness value, and the best fitness value, respectively. Observing Table 2, it becomes apparent that the proposed FOX-GWO method outperformed the other approaches in terms of average fitness values in 13 datasets, signifying that the proposed optimizer obtained the best performance in 72.22 % out of the entire dataset.

Table 5
Standard deviation of fitness value.

Datasets	AEFA	FFA	GWO	PSO	FOX	FOX-GWO
Breastcancer	7.537E-03	9.581E-03	7.896E-03	5.969E-03	7.900E-03	**5.669E-03**
BreastEW	1.395E-02	1.927E-02	1.297E-02	1.268E-02	1.130E-02	**9.156E-03**
Congress	9.252E-03	2.337E-02	2.182E-02	1.705E-02	1.437E-02	**7.719E-03**
Exactly	2.888E-02	1.430E-02	**9.205E-03**	3.752E-02	3.654E-02	6.718E-02
Exactly2	1.286E-02	1.559E-02	**1.390E-16**	1.097E-02	9.565E-03	3.502E-03
HeartEW	5.305E-02	5.027E-02	3.218E-02	5.740E-02	3.094E-02	**1.979E-02**
Ionosphere	2.185E-02	2.614E-02	2.705E-02	2.012E-02	1.652E-02	**1.363E-02**
KrVsKpEW	5.838E-02	7.316E-02	5.304E-02	7.219E-02	4.644E-02	**5.332E-03**
Lymphography	3.218E-02	3.993E-02	2.520E-02	4.516E-02	**2.425E-02**	2.643E-02
M_of_n	4.186E-02	5.296E-02	3.646E-02	5.879E-02	4.842E-02	**2.841E-02**
PenglungEW	4.061E-02	4.493E-02	4.402E-02	**2.749E-02**	3.300E-02	3.235E-02
Sonar	3.104E-02	2.873E-02	3.771E-02	**2.769E-02**	4.351E-02	3.118E-02
SpectEW	1.879E-02	**1.715E-02**	2.552E-02	2.502E-02	1.871E-02	2.015E-02
Tic_tac_toe	2.771E-02	3.048E-02	2.634E-02	2.964E-02	2.030E-02	**1.180E-02**
Vote	2.365E-02	2.343E-02	3.081E-02	1.842E-02	3.219E-02	**1.016E-02**
WaveformEW	1.444E-02	1.617E-02	2.168E-02	1.666E-02	1.017E-02	**3.819E-03**
Wine	9.536E-02	9.345E-02	4.581E-02	9.718E-02	7.503E-02	**2.119E-02**
Zoo	5.193E-02	4.367E-02	5.101E-02	3.490E-02	3.706E-02	**1.620E-02**

Table 6
Average selected features.

Datasets	AEFA	FFA	GWO	PSO	FOX	FOX-GWO
Breastcancer	4.75	4.65	**4.40**	4.85	5.45	4.60
BreastEW	12.40	14.10	15.15	11.60	15.35	**11.30**
Congress	8.00	6.95	6.80	**6.15**	7.60	6.65
Exactly	6.80	5.85	5.05	6.30	7.95	**3.05**
Exactly2	5.25	5.00	3.35	4.25	4.70	**2.40**
HeartEW	5.60	5.20	4.30	5.30	6.05	**4.05**
Ionosphere	14.55	13.25	11.65	14.05	17.45	**10.90**
KrVsKpEW	18.20	18.60	18.60	17.80	23.40	**17.30**
Lymphography	9.50	7.45	8.55	7.90	9.90	**7.35**
M_of_n	6.95	7.30	7.05	6.95	8.20	**6.45**
PenglungEW	**109.85**	143.95	153.45	122.00	183.80	125.80
Sonar	25.60	25.05	25.00	25.00	34.85	**23.35**
SpectEW	10.60	**6.40**	10.20	9.75	11.75	8.80
Tic_tac_toe	5.20	5.05	4.95	4.70	7.95	**4.65**
Vote	6.85	6.85	7.80	**6.30**	8.30	7.05
WaveformEW	20.70	20.05	21.85	**19.20**	27.65	19.60
Wine	5.85	6.30	5.05	5.30	6.10	**4.85**
Zoo	8.40	7.70	**6.85**	7.55	9.35	7.40

The PSO method obtained the best performance in the two datasets. GWO, FFA, and AEFA each achieved the best fitness in one dataset, respectively. Table 3 illustrates the poorest outcomes from the 20 separate trials, demonstrating that FOX-GWO achieved the least poor result in 14 instances of the dataset. This indicates that FOX-GWO outperformed its counterparts in 77.77 % of the dataset. Table 4 presents the best fitness results of 20 runs. The table clearly demonstrates that the FOX-GWO method we propose has achieved impressive outcomes in 11 datasets, competing strongly with other methods. FOX-GWO method surpasses the other optimizer by superseding other algorithms in 61 % of the datasets. These remarkable accomplishments of our proposed FOX-GWO method highlight its ability to effectively strike a balance between exploiting and exploring during the optimization process.

The standard deviation outcomes for the fitness function across all algorithms can be seen in Table 5. FOX-GWO optimizer displayed the best stability across the majority of datasets by having the lowest value when compared to the alternative optimizer. Notably, it achieved the lowest standard deviation value in 12 datasets, surpassing GWO, which obtained the lowest standard deviation values in two datasets. Additionally, both PSO and FOX demonstrated commendable stability. Conversely, AEFA exhibited the worst standard deviation value among the algorithms, indicating relatively lower stability. This suggests that FOX-GWO consistently achieved the highest fitness value in each run, demonstrating the reliability and effectiveness of FOX-GWO. This consistency provides valuable insights into the advanced development of FOX-GWO and its enhanced stability in addressing optimization problems.

Table 6 presents the average count of selected features by FOX-GWO and other optimizers across 20 runs. From the results in Table 6, it is evident that FOX-GWO consistently selects the fewest number of features among the compared algorithms for 11 out of 18 datasets. This observation suggests the utilization of FOX-GWO. Due to its focus on limited feature selection, it has resulted in enhanced classification precision in comparison to alternative algorithms, as demonstrated in Table 6. It is evident that FOX-GWO achieves a strong equilibrium between the smallest quantity of chosen features and improved precision in classification when contrasted with all other optimizers. Thus, this examination implies that FOX-GWO assigns significance to the selection of informative features that

Table 7
Average accuracy.

Datasets	AEFA	FFA	GWO	PSO	FOX	FOX-GWO
Breastcancer	96.7857	96.1143	96.4143	96.4000	96.5714	**97.5571**
BreastEW	93.1579	93.1053	93.5088	**94.7368**	91.6140	93.1228
Congress	94.2890	93.0505	95.6422	94.7706	95.9862	**96.0780**
Exactly	69.4600	68.2800	68.9000	68.4600	69.4200	**81.0800**
Exactly2	74.5400	75.1300	75.8000	74.9700	75.1700	**75.9800**
HeartEW	76.9630	74.0741	75.4074	72.2593	78.3704	**82.8889**
Ionosphere	85.3125	**88.6648**	88.3807	85.2557	85.1420	86.8182
KrVsKpEW	90.4255	83.3354	83.2103	85.6446	84.6996	**95.7854**
Lymphography	**84.3919**	74.4595	79.9324	76.5541	77.8378	81.0135
M_of_n	83.2600	78.6200	81.0800	80.5600	81.1800	**93.8700**
PenglungEW	85.4054	80.2703	84.5946	84.1892	83.7838	**85.9459**
Sonar	78.9904	73.9904	79.0385	77.2115	76.9712	**80.0481**
SpectEW	83.4328	80.7463	82.9851	82.3134	81.0821	**85.3358**
Tic_tac_toe	74.5720	71.1900	73.2255	72.4843	73.5699	**79.7912**
Vote	93.3667	92.2000	93.7000	93.3333	91.2333	**95.6333**
WaveformEW	79.0720	76.9080	77.6060	78.5360	77.3740	**81.9280**
Wine	87.0225	82.3596	90.8989	81.3483	83.2584	**95.4494**
Zoo	90.6863	85.0980	87.2549	88.9216	86.3725	**96.0784**

Table 8
Average time.

Datasets	AEFA	FFA	GWO	PSO	FOX	FOX-GWO
Breastcancer	1.11618	0.49382	0.99391	1.47037	**0.37841**	2.22084
BreastEW	1.03523	0.47455	0.89784	1.32167	**0.36244**	2.63584
Congress	0.80277	0.29279	0.55288	0.97351	**0.23604**	1.46676
Exactly	1.81688	0.79919	0.76194	2.16582	**0.50772**	3.45621
Exactly2	1.69159	0.72914	0.74842	2.05749	**0.41883**	3.46088
HeartEW	0.52953	0.21408	0.33862	0.65876	**0.16233**	0.93638
Ionosphere	0.76279	0.31462	0.57606	1.05052	**0.24633**	1.61355
KrVsKpEW	9.76757	3.70337	6.45066	12.08564	**2.74347**	18.17247
Lymphography	0.34307	0.12827	0.29196	0.43883	**0.11511**	0.59140
M_of_n	1.80400	0.68845	1.48880	2.20343	**0.60551**	3.40402
PenglungEW	0.39062	0.09473	1.60905	1.34849	**0.07930**	0.45349
Sonar	0.78917	0.27770	0.63784	1.18175	**0.21008**	1.31643
SpectEW	0.53755	0.21769	0.45132	0.76616	**0.18413**	0.94703
Tic_tac_toe	1.64148	0.62996	1.19595	2.17435	**0.54144**	2.99888
Vote	0.56812	0.23377	0.41513	0.79962	**0.16702**	1.02266
WaveformEW	17.44378	**5.27093**	10.08424	18.57489	5.98574	25.81245
Wine	0.41230	0.15913	0.34105	0.51094	**0.12903**	0.72070
Zoo	0.27253	0.10857	0.28488	0.36136	**0.08942**	0.48962

greatly enhance classification accuracy.

In Table 7, we present a comparison of classification accuracy between FOX-GWO and other optimization algorithms across 18 datasets. The results in Table 7 indicate that the proposed FOX-GWO exhibits superior classification accuracy compared to all other optimization algorithms. FOX-GWO demonstrates its effectiveness by outperforming other algorithms in 83.33 % of the dataset, showcasing the successful incorporation of GWO and S-shaped transfer function in enhancing the search within the feasible solution area of the problem space.

In Table 8, FOX emerges as the fastest algorithm, highlighting the importance of computationally efficient approaches. Thus far, FOX has proven to be more effective in providing optimal solutions and classification accuracy compared to the other algorithms evaluated. However, due to its hybrid nature, the proposed FOX-GWO algorithm may require more computational time compared to the original FOX algorithm. As a result, a trade-off may be required between optimal answer and computational time, when applying FOX-GWO to a problem.

Additionally, the GWO enhanced the exploitation of the FOX by excavating the promising region and venturing into the quest for a superior solution. Graphical depictions of the convergence curves of the fitness function are illustrated in Fig. 5. The convergence curves should also be taken into account to assess the rate at which FOX-GWO and other optimization algorithms achieve convergence. In instances where the optimization algorithm fails to achieve a balance between exploration and exploitation throughout iterations, it is prone to converging towards the local optimum. The results from the convergence curves reveal that FOX-GWO outperformed all other algorithms in terms of speed in the majority of cases, indicating the superiority of FOX-GWO in handling datasets of varying dimensions. Furthermore, the effectiveness of the proposed FOX-GWO was notable as it seamlessly transitioned from exploratory to exploitative search, thereby accelerating the convergence speed in all scenarios. In order to authenticate the significance of the advancement, statistical examinations such as the Wilcoxon Test and Friedman Test were performed on the results of the fitness

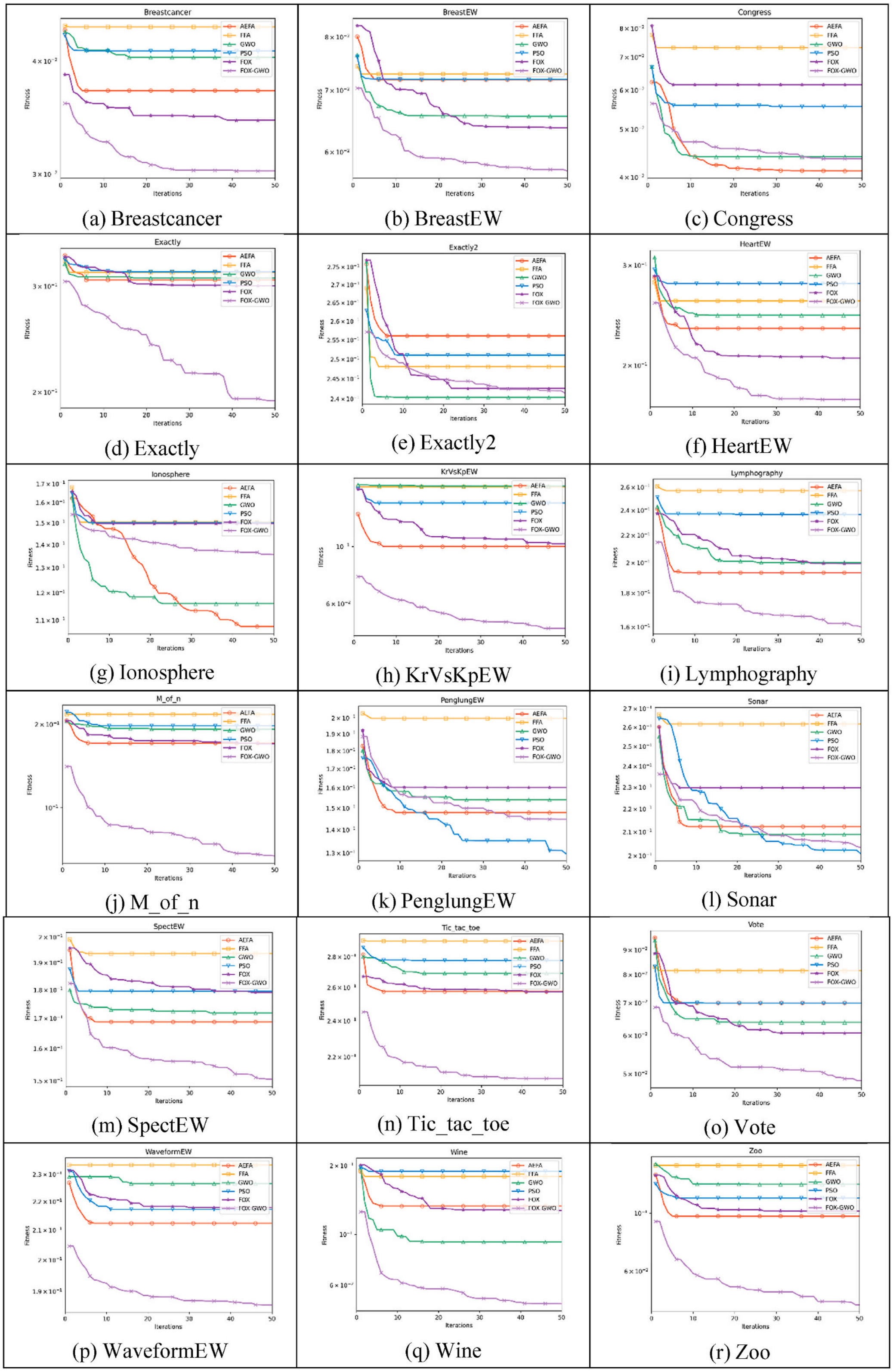

Fig. 5. Convergence of optimizers on fitness functions of 18 datasets.

Table 9
Wilcoxon and Friedman test on fitness value.

Statistic Test	AEFA	FFA	GWO	PSO	FOX	FOX-GWO
P-Value	2.8284e-03	8.4620e-03	2.6698e-02	5.9006e-04	3.3195e-02	–
Friedman Mean	3.3681	4.7472	3.3986	4.0028	3.4583	2.0250
Friedman Rank	2	6	3	5	4	1

function, as exhibited in Table 9.

Based on the reported outcomes, the *P*-value for all compared algorithms is lower than 0.05, and the Friedman Rank is First. Consequently, it is evident that the improvements introduced by FOX-GWO are statistically significant. The proposed algorithm demonstrates superior performance across all metrics, surpassing the other algorithms. This improvement can be attributed to its capacity to produce more optimal solutions using the GWO operator and binary conversion using the newly introduced transfer function. The algorithm's modification amplifies its capability to explore previously unobserved or concealed areas in the problem space. In simpler terms, the suggested FOX-GWO algorithm possesses numerous strengths, including its proficiency in enhancing exploitation, alleviating the problem of local optima, and handling datasets with differing dimensionalities. However, a minor drawback of the FOX-GWO algorithm is its marginally lengthier execution time. This limitation could be addressed in future endeavors by developing a parallel version of the algorithm to reduce its time complexity.

6. Conclusion

This study introduces a novel approach called FOX-GWO, which is an S-shaped hybrid version of the FOX algorithm integrated with Grey Wolf Optimization (GWO), specifically designed to tackle feature selection problems. The proposed enhancements aim to address two main limitations of the original algorithm when dealing with high-dimensional problems: inefficient population value-to-binary conversion and limited exploitation of solutions during the optimization process. The integration of GWO operators and the S-shaped transfer function helps overcome these drawbacks by improving the conversion of the population into binary form and enhancing exploitation. In order to assess the efficacy of the suggested approach, it was subjected to testing on 18 datasets contrasted against various algorithms. The outcomes, gauged across six distinct metrics, consistently exhibited competitive performance in resolving feature selection challenges. The FOX-GWO algorithm offers several distinct advantages when compared to other existing feature selection algorithms. It consistently demonstrates superior performance, excelling in terms of reduced feature selection, accuracy, and computational efficiency. A key strength lies in its robust binary conversion capabilities enabled by an S-shaped transfer function, allowing precise feature selection. Additionally, the integration of Grey Wolf Optimization (GWO) operators enhances exploration and exploitation, enabling FOX-GWO to effectively navigate complex feature selection problem spaces. This ability allows FOX-GWO to explore unexplored domains that might remain uncharted by alternative algorithms. However, it's worth noting that FOX-GWO may have a slightly longer execution time compared to some other algorithms, indicating a trade-off between optimal results and computational time in certain scenarios. As for future prospects, it is recommended to formulate a binary multi-objective variant of the FOX-GWO algorithm and introduce parallel computing to reduce computation time. Additionally, integrating methods to handle constraints within the FOX-GWO algorithm could facilitate the effective resolution of constrained problems. Furthermore, exploring the performance of combining the FOX-GWO algorithm with other optimization algorithms and assessing its effectiveness across a wider spectrum of real-world problems would prove advantageous.

Data availability

The data obtained through the experiments are available upon request.

CRediT authorship contribution statement

Afi Kekeli Feda: Writing – original draft, Conceptualization. **Moyosore Adegboye:** Methodology, Data curation. **Oluwatayomi Rereloluwa Adegboye:** Software, Resources. **Ephraim Bonah Agyekum:** Writing – original draft, Supervision. **Wulfran Fendzi Mbasso:** Writing – review & editing, Formal analysis. **Salah Kamel:** Writing – review & editing, Data curation.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] O.A. Akinola, A.E. Ezugwu, O.N. Oyelade, J.O. Agushaka, A hybrid binary dwarf mongoose optimization algorithm with simulated annealing for feature selection on high dimensional multi-class datasets, Sci. Rep. 12 (1) (2022), https://doi.org/10.1038/s41598-022-18993-0. Art. no. 1.
[2] O.O. Akinola, A.E. Ezugwu, J.O. Agushaka, R.A. Zitar, L. Abualigah, Multiclass feature selection with metaheuristic optimization algorithms: a review, Neural Comput. Appl. 34 (22) (Nov. 2022), https://doi.org/10.1007/s00521-022-07705-4, 19751–19790.

[3] M.H. Nadimi-Shahraki, H. Zamani, S. Mirjalili, Enhanced whale optimization algorithm for medical feature selection: a COVID-19 case study, Comput. Biol. Med. 148 (Sep. 2022) 105858, https://doi.org/10.1016/j.compbiomed.2022.105858.
[4] S. Ahmed, K.H. Sheikh, S. Mirjalili, R. Sarkar, Binary simulated Normal distribution optimizer for feature selection: theory and application in COVID-19 datasets, Expert Syst. Appl. 200 (Aug. 2022) 116834, https://doi.org/10.1016/j.eswa.2022.116834.
[5] T. Bhattacharyya, B. Chatterjee, P.K. Singh, J.H. Yoon, Z.W. Geem, R. Sarkar, Mayfly in harmony: a new hybrid meta-heuristic feature selection algorithm, IEEE Access 8 (2020) 195929–195945, https://doi.org/10.1109/ACCESS.2020.3031718.
[6] V. Bolón-Canedo, N. Sánchez-Maroño, A. Alonso-Betanzos, Introduction to high-dimensionality, in: V. Bolón-Canedo, N. Sánchez-Maroño, A. Alonso-Betanzos (Eds.), Feature Selection for High-Dimensional Data, Artificial Intelligence: Foundations, Theory, and Algorithms, Springer International Publishing, Cham, 2015, pp. 1–12, https://doi.org/10.1007/978-3-319-21858-8_1.
[7] K.H. Sheikh, et al., EHHM: electrical harmony based hybrid meta-heuristic for feature selection, IEEE Access 8 (2020) 158125–158141, https://doi.org/10.1109/ACCESS.2020.3019809.
[8] O. Adegboye, M. Aldağ, E.D. Ülker, Support vector machines in determining the characteristic impedance of microstrip lines, in: J. Hemanth, T. Yigit, B. Patrut, A. Angelopoulou (Eds.), Trends in Data Engineering Methods for Intelligent Systems, Lecture Notes on Data Engineering and Communications Technologies, Springer International Publishing, Cham, 2021, pp. 400–408, https://doi.org/10.1007/978-3-030-79357-9_39.
[9] M.M. Bhatti, K. Vafai, S.I. Abdelsalam, The role of nanofluids in renewable energy engineering, Nanomaterials 13 (19) (Jan. 2023), https://doi.org/10.3390/nano13192671. Art. no. 19.
[10] Y. Liu, W. Liu, M.A. Obaid, I.A. Abbas, Exponential stability of Markovian jumping Cohen–Grossberg neural networks with mixed mode-dependent time-delays, Neurocomputing 177 (Feb. 2016) 409–415, https://doi.org/10.1016/j.neucom.2015.11.046.
[11] B. Du, Y. Liu, I. Atiatallah Abbas, Existence and asymptotic behavior results of periodic solution for discrete-time neutral-type neural networks, J. Franklin Inst. 353 (2) (Jan. 2016) 448–461, https://doi.org/10.1016/j.jfranklin.2015.11.013.
[12] E.I. Abouelmagd, M.E. Awad, E.M.A. Elzayat, I.A. Abbas, Reduction the secular solution to periodic solution in the generalized restricted three-body problem, Astrophys. Space Sci. 350 (2) (2014) 495–505, https://doi.org/10.1007/s10509-013-1756-z. Apr.
[13] E. Pashaei, E. Pashaei, An efficient binary chimp optimization algorithm for feature selection in biomedical data classification, Neural Comput. Appl. 34 (Apr. 2022) 1–25, https://doi.org/10.1007/s00521-021-06775-0.
[14] M. Tubishat, M. Alswaitti, S. Mirjalili, M.A. Al-Garadi, M.T. Alrashdan, T.A. Rana, Dynamic butterfly optimization algorithm for feature selection, IEEE Access 8 (2020) 194303–194314, https://doi.org/10.1109/ACCESS.2020.3033757.
[15] G. Dhiman, et al., BEPO: a novel binary emperor penguin optimizer for automatic feature selection, Knowl. Base Syst. 211 (Jan. 2021) 106560, https://doi.org/10.1016/j.knosys.2020.106560.
[16] M. Sharma, P. Kaur, A comprehensive analysis of nature-inspired meta-heuristic techniques for feature selection problem, Arch. Comput. Methods Eng. 28 (3) (May 2021) 1103–1127, https://doi.org/10.1007/s11831-020-09412-6.
[17] Y. Li, T. Li, H. Liu, Recent advances in feature selection and its applications, Knowl. Inf. Syst. 53 (3) (Dec. 2017) 551–577, https://doi.org/10.1007/s10115-017-1059-8.
[18] M. Ghosh, S. Adhikary, K.K. Ghosh, A. Sardar, S. Begum, R. Sarkar, Genetic algorithm based cancerous gene identification from microarray data using ensemble of filter methods, Med. Biol. Eng. Comput. 57 (1) (Jan. 2019) 159–176, https://doi.org/10.1007/s11517-018-1874-4.
[19] M. Mafarja, I. Aljarah, H. Faris, A.I. Hammouri, A.M. Al-Zoubi, S. Mirjalili, Binary grasshopper optimisation algorithm approaches for feature selection problems, Expert Syst. Appl. 117 (Mar. 2019) 267–286, https://doi.org/10.1016/j.eswa.2018.09.015.
[20] P. Agrawal, H.F. Abutarboush, T. Ganesh, A.W. Mohamed, Metaheuristic algorithms on feature selection: a survey of one decade of research (2009-2019), IEEE Access 9 (2021) 26766–26791, https://doi.org/10.1109/ACCESS.2021.3056407.
[21] S. Bahassine, A. Madani, M. Al-Sarem, M. Kissi, Feature selection using an improved Chi-square for Arabic text classification, Journal of King Saud University - Computer and Information Sciences 32 (2) (Feb. 2020) 225–231, https://doi.org/10.1016/j.jksuci.2018.05.010.
[22] S. Mirjalili, Genetic algorithm, in: S. Mirjalili (Ed.), Evolutionary Algorithms and Neural Networks: Theory and Applications, Studies in Computational Intelligence., Springer International Publishing, Cham, 2019, pp. 43–55, https://doi.org/10.1007/978-3-319-93025-1_4.
[23] L. Abualigah, D. Yousri, M. Abd Elaziz, A.A. Ewees, M.A.A. Al-qaness, A.H. Gandomi, Aquila Optimizer: a novel meta-heuristic optimization algorithm, Comput. Ind. Eng. 157 (2021) 107250, https://doi.org/10.1016/j.cie.2021.107250. Jul.
[24] S. Mirjalili, A. Lewis, The whale optimization algorithm, Adv. Eng. Software 95 (May 2016) 51–67, https://doi.org/10.1016/j.advengsoft.2016.01.008.
[25] S. Mirjalili, S.M. Mirjalili, A. Lewis, Grey wolf optimizer, Adv. Eng. Software 69 (Mar. 2014) 46–61, https://doi.org/10.1016/j.advengsoft.2013.12.007.
[26] P. Agrawal, T. Ganesh, A.W. Mohamed, Chaotic gaining sharing knowledge-based optimization algorithm: an improved metaheuristic algorithm for feature selection, Soft Comput. 25 (14) (Jul. 2021) 9505–9528, https://doi.org/10.1007/s00500-021-05874-3.
[27] R. SaiSindhuTheja, G.K. Shyam, An efficient metaheuristic algorithm based feature selection and recurrent neural network for DoS attack detection in cloud computing environment, Appl. Soft Comput. 100 (Mar. 2021) 106997, https://doi.org/10.1016/j.asoc.2020.106997.
[28] T. Dokeroglu, A. Deniz, H.E. Kiziloz, A comprehensive survey on recent metaheuristics for feature selection, Neurocomputing 494 (Jul. 2022) 269–296, https://doi.org/10.1016/j.neucom.2022.04.083.
[29] S.S. Kareem, R.R. Mostafa, F.A. Hashim, H.M. El-Bakry, An effective feature selection model using hybrid metaheuristic algorithms for IoT intrusion detection, Sensors 22 (4) (Jan. 2022), https://doi.org/10.3390/s22041396. Art. no. 4.
[30] S. Barua, A. Merabet, Lévy Arithmetic Algorithm: an enhanced metaheuristic algorithm and its application to engineering optimization, Expert Syst. Appl. (2023) 122335, https://doi.org/10.1016/j.eswa.2023.122335. Nov.
[31] O.R. Adegboye, E. Deniz Ülker, Gaussian mutation specular reflection learning with local escaping operator based artificial electric field algorithm and its engineering application, Appl. Sci. 13 (7) (Jan. 2023), https://doi.org/10.3390/app13074157, 7.
[32] D.H. Wolpert, W.G. Macready, No free lunch theorems for optimization, IEEE Trans. Evol. Comput. 1 (1) (Apr. 1997) 67–82, https://doi.org/10.1109/4235.585893.
[33] O.R. Adegboye, E. Deniz Ülker, Hybrid artificial electric field employing cuckoo search algorithm with refraction learning for engineering optimization problems, Sci. Rep. 13 (1) (2023), https://doi.org/10.1038/s41598-023-31081-1. Art. no. 1, Mar.
[34] G.I. Sayed, A. Tharwat, A.E. Hassanien, Chaotic dragonfly algorithm: an improved metaheuristic algorithm for feature selection, Appl. Intell. 49 (1) (Jan. 2019) 188–205, https://doi.org/10.1007/s10489-018-1261-8.
[35] S. Taghian, M.H. Nadimi-Shahraki, Binary sine cosine algorithms for feature selection from medical data, ACIJ 10 (5) (Sep. 2019) 1–10, https://doi.org/10.5121/acij.2019.10501.
[36] M. Abd Elaziz, A.A. Ewees, D. Yousri, L. Abualigah, M.A.A. Al-qaness, Modified marine predators algorithm for feature selection: case study metabolomics, Knowl. Inf. Syst. 64 (1) (Jan. 2022) 261–287, https://doi.org/10.1007/s10115-021-01641-w.
[37] F.S. Gharehchopogh, I. Maleki, Z.A. Dizaji, Chaotic vortex search algorithm: metaheuristic algorithm for feature selection, Evol. Intel. 15 (3) (Sep. 2022) 1777–1808, https://doi.org/10.1007/s12065-021-00590-1.
[38] R.K. Agrawal, B. Kaur, S. Sharma, Quantum based Whale Optimization Algorithm for wrapper feature selection, Appl. Soft Comput. 89 (Apr. 2020) 106092, https://doi.org/10.1016/j.asoc.2020.106092.
[39] W. Ma, X. Zhou, H. Zhu, L. Li, L. Jiao, A two-stage hybrid ant colony optimization for high-dimensional feature selection, Pattern Recogn. 116 (Aug. 2021) 107933, https://doi.org/10.1016/j.patcog.2021.107933.
[40] A. Chaudhuri, T.P. Sahu, Feature selection using Binary Crow Search Algorithm with time varying flight length, Expert Syst. Appl. 168 (Apr. 2021) 114288, https://doi.org/10.1016/j.eswa.2020.114288.
[41] A.G. Hussien, A.E. Hassanien, E.H. Houssein, S. Bhattacharyya, M. Amin, S-Shaped binary whale optimization algorithm for feature selection, in: S. Bhattacharyya, A. Mukherjee, H. Bhaumik, S. Das, K. Yoshida (Eds.), Recent Trends in Signal and Image Processing, Advances in Intelligent Systems and Computing, Springer, Singapore, 2019, pp. 79–87, https://doi.org/10.1007/978-981-10-8863-6_9.

[42] D. Chauhan, A. Yadav, Binary artificial electric field algorithm, Evol. Intel., Apr. (2022), https://doi.org/10.1007/s12065-022-00726-x.
[43] E. Emary, H.M. Zawbaa, A.E. Hassanien, Binary grey wolf optimization approaches for feature selection, Neurocomputing 172 (Jan. 2016) 371–381, https://doi.org/10.1016/j.neucom.2015.06.083.
[44] M.A. Khanesar, M. Teshnehlab, M.A. Shoorehdeli, A novel binary particle swarm optimization, in: 2007 Mediterranean Conference on Control & Automation, Jun. 2007, pp. 1–6, https://doi.org/10.1109/MED.2007.4433821.
[45] B. Crawford, R. Soto, M.O. Suárez, F. Paredes, F. Johnson, Binary Firefly algorithm for the set covering problem, in: 2014 9th Iberian Conference on Information Systems and Technologies (CISTI), Jun, 2014, pp. 1–5, https://doi.org/10.1109/CISTI.2014.6877090.
[46] H. Mohammed, T. Rashid, FOX: a FOX-inspired optimization algorithm, Appl. Intell. 53 (1) (Jan. 2023) 1030–1050, https://doi.org/10.1007/s10489-022-03533-0.