STATISTICS, OPTIMIZATION AND INFORMATION COMPUTING
Stat., Optim. Inf. Comput., Vol. **14**, November 2025, pp 2643–2662.
Published online in International Academic Press (www.IAPress.org)

Fast and Efficient Feature Selection in AI Application Based on Enhanced Binary Secretary Bird Optimization Algorithm

Amr H. Abdelhaliem [1], Islam S. Fathi [2,*], Mohammed Tawfik [3]

[1]*Department of Cyber Security, Faculty of Science and Information Technology, Irbid National University, Irbid, Jordan*
[2]*Department of Computer Science, Faculty of Information Technology, Ajloun National University, P.O.43, Ajloun-26810, Jordan*
[3]*Department of Cyber Security, Faculty of Information Technology, Ajloun National University, P.O.43, Ajloun-26810, Jordan*

Abstract Metaheuristic algorithms, which draw inspiration from natural phenomena, have emerged as robust tools within computational intelligence and are widely applied across various fields. The effective use of artificial intelligence requires extracting pertinent information from extensive datasets. Working with big data presents several obstacles, including high dimensionality, duplicate data, and extraneous information. Feature selection techniques aim to reduce complexity by identifying and removing unnecessary attributes, which helps optimize computational resources in terms of both processing time and storage requirements. This paper introduces an enhanced binary variant of the Secretary Bird Optimization Algorithm (SBOA) designed to address feature selection challenges. The SBOA is a recent metaheuristic approach that replicates the survival tactics of secretary birds, specifically their hunting and predator avoidance behaviors. As computational methods, metaheuristic algorithms help solve complex optimization tasks. The proposed EB-SBOA incorporates two key improvements to the original SBOA: a refracted opposition-based learning method during initialization to expand population diversity, and a random replacement mechanism to improve convergence precision. The algorithm's effectiveness was tested using 25 benchmark datasets and compared against six contemporary wrapper-based feature selection techniques. Results demonstrate that EB-SBOA achieves superior performance in three key metrics: classification accuracy, average fitness value, and feature reduction. The findings' statistical validity was confirmed through Wilcoxon rank-sum testing.

Keywords Secretary bird optimization algorithm (SBOA), computational intelligence, Feature Selection (FS), Classification, Random replacement

DOI: 10.19139/soic-2310-5070-2597

1. Introduction

The widespread integration of internet and computing technologies has resulted in the generation of enormous datasets containing hundreds of distinct features. Data mining techniques aim to extract actionable insights from these extensive datasets to enable informed decision-making processes. The strategic selection of appropriate and valuable features has demonstrated substantial impact across various technological domains,including data mining [1], IoT systems [2], machine learning applications [3], biomedical signal processing [63, 64, 65], and image processing [4]. In machine learning systems specifically, the presence of redundant, irrelevant, and disorganized records within high-dimensional datasets negatively impacts classification accuracy while increasing computational overhead [5]. IoT systems frequently encounter challenges in storing and processing the vast quantities of sensor-generated data. This issue is compounded by the presence of superfluous and duplicate features. To address these challenges, preprocessing techniques, particularly feature selection, are necessary to effectively manage high-dimensional data and eliminate redundant attributes [6]. The process of feature selection represents a fundamental

*Correspondence to: Islam S. Fathi (Email: i.mohamed@anu.edu.jo). Department of Computer Science, Faculty of Information Technology, Ajloun National University, P.O.43, Ajloun-26810, Jordan.

ISSN 2310-5070 (online) ISSN 2311-004X (print)

component of data preparation and is essential for developing effective models by extracting the relevant features from the available datasets.

A FS system consists of three core elements: (i) classification algorithms, including support vector machines (SVMs) [7] and k nearest neighbor (kNN) [8], (ii) evaluation metrics, and (iii) search methodologies for identifying optimal features. The field of feature selection encompasses two main methodological approaches: wrapper-based and filter-based strategies. Wrapper methodologies evaluate feature subsets depend on their performance with specific classification algorithms. These methods operate the classification algorithm independently to assess the quality of selected subsets through their classification performance [9]. In contrast, filter approaches function independently of any learning models, evaluating feature subsets based purely on data characteristics rather than model-specific criteria. It's important to note that while filter methods may not always identify the ideal feature subset, research has shown that wrapper approaches typically produce superior feature subsets for specific classifiers [10]. The fundamental goal of feature selection is to determine the optimal subset from all possible feature combinations. Search algorithms fall into two primary categories: accurate search methods and metaheuristic [11]. Accurate search techniques conduct an exhaustive exploration of the search space, which grows exponentially with the number of features, demanding significant computational resources. In contrast, metaheuristic approaches employ stochastic optimization by beginning with random solution sets to explore the search space effectively. Metaheuristic approaches have gained prominence in feature selection due to their capacity to generate near-optimal solutions within acceptable computational timeframes [12]. These algorithms are particularly valuable because of their straightforward implementation and adaptability to various problem domains. A key strength of metaheuristic methods lies in their sophisticated mechanism for avoiding early convergence by maintaining an optimal equilibrium between broad search space exploration and focused solution refinement.

The Secretary Bird Algorithm (SBA) [13] represents a recent innovation in metaheuristic approaches that simulates the hunting behavior of the secretary bird, specifically its strategy for stalking and capturing prey with precision. Within the diverse optimization algorithms, a recent innovation has emerged that mimics natural phenomena, specifically drawing its inspiration from the predatory strategies exhibited by the secretary bird, a distinctive raptor species. The Secretary Bird Optimization Algorithm (SBOA) draws its inspiration from the hunting patterns of its namesake bird. By observing how these birds systematically search for and capture prey, researchers developed an optimization method that follows similar principles to find the best possible solutions in optimization problems. The algorithm's structure parallels the bird's natural hunting sequence: the initial setup phase corresponds to how the bird prepares for hunting, followed by three distinct exploration phases that mirror the bird's hunting stages. The final optimization phase incorporates two key strategies, which are directly influenced by the dual hunting techniques employed by secretary birds in nature.

While metaheuristic algorithms have demonstrated significant utility in feature selection applications in recent years, existing approaches continue to face various challenges requiring further research attention. The ongoing refinement of optimization techniques remains essential for improving outcomes. To enhance the original SBOA's effectiveness, two key modifications have been implemented: a refracted opposition-based learning method during initialization to expand population diversity, and a random replacement mechanism to improve convergence precision. The refracted opposition-based learning component serves to enhance population diversity and reduce the likelihood of convergence to local optima, while a random replacement mechanism to improve convergence precision.

We present several novel contributions can be outlined in the following elements:

- Integration the refracted opposition-based learning strategies with SBOA to achieved enhanced population diversity and more thorough exploration of the initial search space.
- Introduces a crisscross approach that significantly improves the convergence precision of the SBOA method.
- We present a binary modification of SBOA, called EB-SBOA that provides solutions for feature selection difficulties.
- Experimental validation was performed using a comprehensive suite of 25 recognized benchmark datasets.

2. Literature review

The exponential growth in computer technologies has led to the generation of massive datasets with complex characteristics. In these high-dimensional datasets, the presence of redundant, irrelevant, and chaotic records significantly impacts classification accuracy and increases computational complexity [14, 15]. Feature selection has emerged as a crucial component of data preparation, playing a vital role in building robust models [16, 17]. The effectiveness of metaheuristic algorithms in addressing feature selection challenges largely depends on their ability to generate optimal solutions [18, 19].

Metaheuristics are typically categorized into four main groups based on their inspiration sources: swarm intelligence, human-based methods, physics-based methods, and evolutionary algorithms [20, 21, 22, 23]. The effectiveness of these algorithms stems from their ease of implementation and uncomplicated nature, allowing them to be readily adapted to various specialized applications. In the field of feature selection, significant progress has been made through the application of swarm intelligence techniques, which are derived from studying how animals behave collectively in groups. This category encompasses several key algorithms, including the BFPA [24], BinHOA [25], BDA [26], BCS [27], and PSO [28]. A notable advancement came from Xue et al., who enhanced PSO to achieve faster processing while optimizing both feature reduction and classification performance [29]. Another important category draws inspiration from human social dynamics and behavioral patterns. A significant contribution in this area was made by Prachi [30], who developed FSNBGSK, an innovative binary adaptation of the GSK method, which proved highly effective across 23 different benchmark datasets [31]. This category also includes several other notable approaches such as ICA [32], cultural evolution algorithm [33, 34], TLBO [35], and the more recently developed VPL [36]. Additionally, the field has benefited from physics-inspired methodologies, which utilize principles from natural physical processes. These approaches include the lightning search algorithm [37], multi-verse optimizer [38], Henry gas solubility optimization [39], and gravitational search techniques [40, 41]. Beyond traditional optimization problems, metaheuristic algorithms have found success in emerging computational paradigms such as fog computing and IoT systems. Fathi and Tawfik [61] recently demonstrated the effectiveness of bio-inspired optimization in fog computing environments, where the Pufferfish Optimization Algorithm achieved superior performance in IoT node placement optimization, maintaining network connectivity above 99.5% and coverage above 99.2%. This showcases the adaptability of nature-inspired algorithms to diverse optimization challenges, reinforcing their potential across multiple technological domains including feature selection. A significant advancement in this category is the Equilibrium Optimizer (EO), with its binary variant (BinEO) introduced by Mohamed Mostafa Saleh et al. [42]. The BinEO incorporates opposition-based learning and local search algorithms, demonstrating remarkable effectiveness when compared to established algorithms. Evolutionary algorithms, inspired by Darwinian principles, continue to play a crucial role in feature selection. The genetic algorithm (GA) [43] has shown exceptional capability in addressing feature selection challenges [44], particularly when combined with chaotic optimization for text categorization [45]. Other evolutionary approaches include differential evolution algorithms [46, 47] and stochastic fractal search [48].The application of metaheuristic algorithms extends beyond traditional feature selection to specialized domains requiring both optimization and feature engineering. Tawfik [62] demonstrated the integration of Adaptive Grey Wolf Optimization (AGWO) with CatBoost-based feature selection for intrusion detection in IoT environments, achieving over 99% accuracy while maintaining computational efficiency suitable for resource-constrained devices. This showcases the versatility of nature-inspired optimization algorithms in addressing complex feature selection challenges across diverse applications.

Recent developments have seen the emergence of hybrid approaches that combine multiple algorithms to leverage their respective strengths [49]. For instance, Al-Tashi et al. [50] developed a hybrid approach by creating a binary adaptation of the Whale Optimization Algorithm (WOA), integrating it with the Simulated Annealing (SA) algorithm. This hybrid approach demonstrated superior performance in terms of both accuracy and computational efficiency across 18 UCI benchmark datasets.

The Secretary Bird Optimization Algorithm (SBOA) represents an innovative contribution to the field of swarm intelligence meta-heuristics. This population-based optimization approach models its functionality on secretary birds, where individual birds within the population serve as solution agents. Within the algorithm's framework, each bird's spatial position corresponds to specific decision variable values, effectively representing potential solutions to the optimization problem. Research has shown SBOA's exceptional capabilities, with its performance metrics surpassing those of many established meta-heuristic techniques, particularly in balancing exploration and exploitation of the solution space. In our study, we develop a binary variant of SBOA, implementing it as a wrapper-based feature selection method to enhance both feature selection and classification performance. We augment the algorithm's effectiveness through two key modifications: incorporating refracted opposition-based learning during the initialization phase to ensure population diversity, and implementing a random replacement strategy to achieve more precise convergence.

3. Secretary bird optimization algorithm

3.1. Inspiration of Secretary bird optimization algorithm

The Secretary Bird (Sagittarius serpentarius), native to sub-Saharan Africa, inhabits various open environments from grasslands to savannas. Secretary birds display grey-brown coloration across their dorsal feathers, contrasting with white plumage on their chest region and distinctive black feathering on their underside as shown in Figure 1 [13]. The species exhibits two survival strategies: camouflaging when possible or quickly fleeing either by foot or flight. These behavioral patterns inspired the Secretary Bird Optimization Algorithm (SBOA), with the bird's hunting preparation and execution reflected in the algorithm's initialization and exploration phases.

Figure 1. Secretary bird

3.2. Mathematical model

3.2.1. Initial preparation phase SBOA is a population-based metaheuristic algorithm where individual Secretary Birds act as population members. Their spatial positions represent potential solutions through decision variable values. The algorithm initializes by randomly distributing birds' positions across the search space using Equation (1).

$$X_{i,j} = lb_j + r \times (ub_j - lb_j), \quad i = 1, 2, \ldots, N, \quad j = 1, 2, \ldots, \text{Dim} \tag{1}$$

$X_{i,j}$ is where the i-th Secretary Bird is located, while ub_j and lb_j establish the minimum and maximum boundaries of the search space respectively. The variable r indicates a randomly generated number that falls between 0 and 1.

SBOA is a population-based optimization method that starts with multiple potential solutions (Equation 2). These solutions are randomly distributed between upper and lower boundaries. In each iteration, the algorithm tracks the best solution found as the approximate optimum.

$$X = \begin{bmatrix} x_{1,1} & x_{1,2} & \cdots & x_{1,j} & \cdots & x_{1,\text{Dim}} \\ x_{2,1} & x_{2,2} & \cdots & x_{2,j} & \cdots & x_{2,\text{Dim}} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ x_{i,1} & x_{i,2} & \cdots & x_{i,j} & \cdots & x_{i,\text{Dim}} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ x_{N,1} & x_{N,2} & \cdots & x_{N,j} & \cdots & x_{N,\text{Dim}} \end{bmatrix}_{N\times\text{Dim}} \tag{2}$$

X represents the entire collection of Secretary Birds, while X_i refers to a specific bird (the ith member). $X_{i,j}$ indicates the value of the j-th variable for the i-th Secretary Bird. N specifies the total number of Secretary Birds in the population, and Dim represents the number of dimensions in the problem being solved.

Each secretary bird's proposed variable values are evaluated by the objective function, with results compiled into a vector using Equation (3).

$$F = \begin{bmatrix} F_1 \\ \vdots \\ F_i \\ \vdots \\ F_N \end{bmatrix}_{N\times 1} = \begin{bmatrix} F(X_1) \\ \vdots \\ F(X_i) \\ \vdots \\ F(X_N) \end{bmatrix}_{N\times 1} \tag{3}$$

Here, F represents the vector of the objective function, while the objective function value achieved by the i-th secretary bird is denoted as F_i.

3.2.2. Search pattern of secretary bird Secretary birds hunt snakes in three distinct phases: detection, consumption, and attack, each taking one-third of the total hunting duration (T). In SBOA, the initial search phase (t ¡ T/3) is crucial, mimicking how secretary birds use their height and keen vision to spot snakes in vegetation while safely probing the ground with their long legs. This phase implements differential evolution, using inter-individual variations to generate new solutions and prevent premature convergence to local optima. The birds' positional updates during prey detection are mathematically modeled through Equations (4) and (5), enabling comprehensive exploration of the solution space.

$$\text{While } t < \frac{1}{3}T, \quad x_{i,j}^{\text{new},P1} = x_{i,j} + (x_{\text{random}_1} - x_{\text{random}_2}) \times R_1 \tag{4}$$

$$X_i = \begin{cases} X_i^{\text{new},P1}, & \text{if } F_i^{\text{new},P_1} < F_i \\ X_i, & \text{else} \end{cases} \tag{5}$$

t denotes the iteration counter, while T represents the maximum number of iterations. The updated position of the i-th secretary bird during the initial phase is indicated by $X_i^{\text{new},P1}$. Within this phase, x_{random_1} and x_{random_2} signify randomly selected candidate solutions. The term R_1 defines a randomly generated one-dimensional array of size $1 \times \text{Dim}$, with values distributed uniformly over [0, 1], where Dim corresponds to the solution space dimensionality. Furthermore, $x_{i,j}^{\text{new},P1}$ represents the positional value along the j-th dimension, while F_i^{new,P_1} indicates the corresponding objective function fitness value.

During prey consumption, secretary birds employ careful positioning and strategic footwork rather than immediate attack. Using their elevated stance and protected legs, they exhaust snakes through calculated movements, modeled mathematically using Brownian motion (RB). The SBOA algorithm incorporates these behaviors by combining historical best positions with Brownian principles through Equations (6) and (7). This approach enables effective local exploration while maintaining connection to successful positions, balancing global search with local optimization to avoid premature convergence.

$$RB = \text{randn}(1, \text{Dim}) \tag{6}$$

$$\text{While } \frac{1}{3}T < t < \frac{2}{3}T, \quad x_{i,j}^{\text{new},P1} = x_{\text{best}} + \exp\left(\left(\frac{t}{T}\right)^4\right) \times (RB - 0.5) \times (x_{\text{best}} - x_{i,j}) \tag{7}$$

$$X_i = \begin{cases} X_i^{\text{new},P_1}, & \text{if } F_i^{\text{new},P_1} < F_i \\ X_i, & \text{else} \end{cases} \tag{8}$$

The equation incorporates a random array, denoted as randn $(1, \text{Dim})$, which generates values following a normal distribution with parameters $\mu = 0$ and $\sigma = 1$. This array has dimensions of $1 \times \text{Dim}$. The variable x_{best} represents the optimal value identified up to the current iteration.

In the final hunting phase, secretary birds execute precise attacks using powerful leg strikes aimed at the snake's head for rapid immobilization. For larger prey, they may lift and drop snakes from height. The SBOA models this behavior using Levy flight patterns, combining short movements with occasional long jumps. This dual-scale approach balances global exploration with local precision optimization, helping prevent convergence to local optima while efficiently seeking optimal solutions.

SBOA incorporates a nonlinear perturbation factor $(1 - t/T)^{(2 \times t/T)}$ to balance exploration and exploitation, prevent early convergence, and improve performance. Position updates during the attack phase are defined by Equations (9) and (10).

$$\text{While } t > \frac{2}{3}T, \quad x_{i,j}^{\text{new},P1} = x_{\text{best}} + \left(\left(1 - \frac{t}{T}\right)^{\left(2 \times \frac{t}{T}\right)}\right) \times x_{i,j} \times RL \tag{9}$$

$$X_i = \begin{cases} X_i^{\text{new},P1}, & \text{if } F_i^{\text{new},P_1} < F_i \\ X_i, & \text{else} \end{cases} \tag{10}$$

The algorithm's optimization precision is improved through the incorporation of a weighted Levy flight component, represented by the notation 'RL'.

$$RL = 0.5 \times \text{Levy}(\text{Dim}) \tag{11}$$

$\text{Levy}(\text{Dim})$ is the Levy flight distribution function, is computed using the following equation:

$$\text{Levy}(D) = s \times \frac{u \times \sigma}{|v|^{\frac{1}{n}}} \tag{12}$$

The model incorporates two constant values: one set at 0.01 and another at π. The system also utilizes random variables constrained within a specific interval. The calculation method for this component is subsequently defined.

$$\sigma = \left(\frac{\Gamma(1+\eta) \times \sin\left(\frac{\pi\eta}{2}\right)}{\Gamma\left(\frac{1+\eta}{2}\right) \times \eta \times 2^{\left(\frac{\eta-1}{2}\right)}}\right)^{\frac{1}{\eta}} \tag{13}$$

Here, Γ is the gamma function and η has a value of 1.5.

3.2.3. Escape process of secretary bird Secretary birds face threats from predators like eagles, hawks, and candies. They employ two defense mechanisms: physical evasion and cryptic behavior. Their long legs enable rapid running, covering 20-30 kilometers daily, earning them the name “marching eagles.” They can also fly to escape danger. Additionally, they use camouflage by blending with their environment. The SBOA model incorporates these as two equal scenarios: environmental camouflage (C1) and active evasion (C2).

Upon detecting predators, secretary birds first seek nearby camouflage. If unavailable, they resort to flight or running. Their strategy incorporates a dynamic perturbation factor, $(1 - t/T)^2$, which optimizes the balance between exploring new areas and using known solutions. This factor can be adjusted to favor either exploration or exploitation. These evasion patterns are mathematically expressed in Equations (14) and (15).

$$x_{i,j}^{\text{new},P2} = \begin{cases} p_1 : x_{\text{best}} + (2 \times RB - 1) \times \left(1 - \frac{t}{T}\right)^2 \times x_{ij}, & \text{if } rand < r_i \\ p_2 : x_{i,j} + R_2 \times (x_{\text{random}} - K \times x_{i,j}), & \text{else} \end{cases} \tag{14}$$

$$X_i = \begin{cases} X_i^{\text{new},P2}, & \text{if } F_i^{\text{new},P2} < F_i \\ X_i, & \text{else} \end{cases} \tag{15}$$

The constant r is set to 0.5, while R_2 generates a one-dimensional array of size $(1 \times \text{Dim})$ following a normal distribution. The variable x_{random} denotes a randomly generated candidate solution within the current iteration. K represents an integer value that is randomly chosen as either 1 or 2, with its calculation method detailed in Equation (16).

$$K = \text{round}(1 + \text{rand}(1, 1)) \tag{16}$$

Where rand (1, 1) is randomly generating a random number in (0, 1).

3.3. Algorithm complexity analysis

The evaluation of an algorithm's performance includes analyzing how much time it requires for optimization tasks. To assess the computational efficiency of SBOA, this study employs Big O notation to examine time complexity. Using N to indicate the secretary bird population size, Dim for dimensionality, and T for the maximum iteration count, we can break down the complexity analysis. The initial population randomization process has a complexity of O(N). The subsequent solution updating phase involves two components: position optimization and solution updating across all dimensions, resulting in a complexity of O(T × N) + O(T × N × Dim). Combining these elements, the algorithm's overall computational complexity can be summarized as O(N × (T × Dim + 1)).

4. The proposed improved binary version (EB-SBOA)

For binary-type problems like feature selection, solutions take the form of one-dimensional arrays containing just 1s and 0s. These binary values serve as indicators, with 1 signifying a selected feature and 0 representing an excluded one. The array's length corresponds directly to the total number of features present in the dataset. Figure 2 demonstrates this binary encoding approach for a SBOA related solution, where the array size matches D, the total feature count in the dataset.

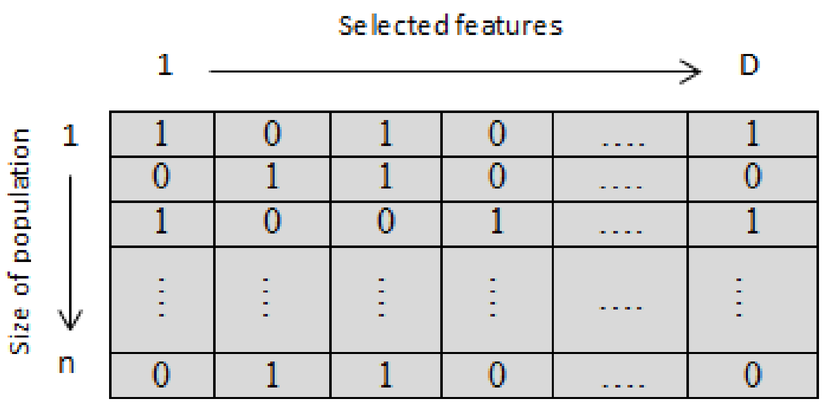

Figure 2. The SBOA binary representation.

4.1. Strategic Initialization through Refracted Opposition-Based Learning

Maximizing the efficiency of localized search parameters is fundamental to achieving high-quality optimal solutions in computational optimization. The EB-SBOA algorithm introduces an innovative approach by implementing the Refracted Opposition-Based Learning technique to enhance initial population generation [51]. This method expands the potential solution landscape by generating alternative solution candidates through opposition-based transformations, thereby increasing the likelihood of discovering more refined problem solutions. Previous research has demonstrated that integrating metaheuristic approaches with opposition-based learning can substantially improve algorithmic precision and solution quality.

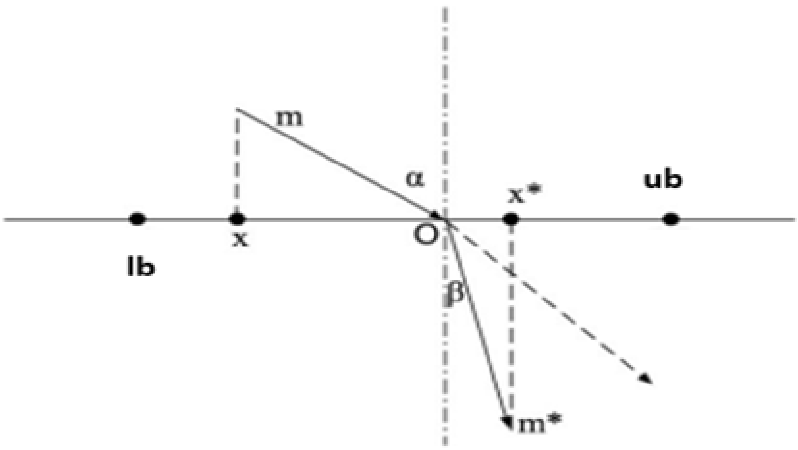

Figure 3. The Refracted opposition-based learning.

The solution exploration domain is constrained within a predefined interval on the x-axis, bounded by lower [lb] and upper [ub] limits. The coordinate system's origin is strategically positioned at the exact midpoint of this interval, serving as a critical reference point for computational navigation. Two angular parameters, α and β, play pivotal roles in defining the solution transformation process: α representing the incident angle and β characterizing the refraction angle.

Additionally, m and m^* represent the distances of the incident and refracted rays, respectively. The refraction can be calculated using the following formula:

$$n = \frac{\sin\alpha}{\sin\beta} = \frac{\frac{lb+ub}{2} - x}{X^* - \frac{lb+ub}{2}} \times \frac{m^*}{m} \tag{17}$$

Put $\sigma = \frac{m^*}{m}$ and n=1 in Eq. (17) and SBOA is applied to a high-dimensional space, the refracted direction $X^*_{i,j}$ can be determined using the following equation:

$$X^*_{i,j} = \frac{lb_j + ub_j}{2} + \frac{lb_j + ub_j}{2\sigma} - \frac{X_{i,j}}{\sigma} \tag{18}$$

where $X_{i,j}$ indicates the i-th secretary bird position at j-th dimensions, $X^*_{i,j}$ is the refracted inverse solution of $X_{i,j}$, and lb_j and ub_j are the lower and upper bounds.

In the process of binary algorithmic conversion, solutions must first be transformed from their original state into binary format using specialized conversion techniques. The sigmoid function, a widely recognized member of the S-shaped function family, is frequently employed for this purpose [52]. Its primary function is to take any real number as input and map it to either 0 or 1 as output.

$$S\left(X_{i,j}\right) = \frac{1}{1+e^{-X_{i,j}}}, \quad X_{\text{binary}} = \begin{cases} 1 & \text{if rand} \geq S\left(X_{i,j}\right) \\ 0 & \text{otherwise} \end{cases} \tag{19}$$

4.2. Application of Local Search Algorithm (LSA) to EB-SBOA

The algorithm applies a local search procedure in each iteration to the current optimal solution SB to potentially find an improved outcome. The local search algorithm (LSA) operates by randomly identifying three features from the present best solution during each cycle [42]. These selected features undergo value inversion, where 1s are converted to 0s and 0s to 1s. The algorithm then evaluates the fitness score of this modified solution. The existing best solution B is only replaced if the newly generated solution demonstrates superior fitness compared to the previous one.

Algorithm 1 Local search algorithm

$t\_value = SB$ where SB is the optimum solution.
while $t < maximum\_iterations$ **do**
  Three features are selected at random from t_value.
  if $selected\_feature == 1$ (where 1 indicates that the feature is selected and 0 indicates that it is not selected) **then**
    $selected\_feature = 0$
  else
    $selected\_feature = 1$
  end if
  Assess t_value based on kNN or SVM classifiers
  Compute the fitness value of t_value
  if $f(t\_value) < f(SB)$ **then**
    $SB = t\_value$
  end if
  $t = t + 1$
end while
return SB

4.3. The fitness function

Choosing more features from the data isn't always easy because classifier performance usually drops when faced with superfluous or unnecessary features. Hence, it is imperative to tackle this problem by decreasing the data's dimensionality. When assessing solutions, both the accuracy of classification and the quantity of selected features are important factors to consider. If two solutions achieve the same level of accuracy in classification, priority is given to the option that uses the least number of specified characteristics. By minimizing the number of characteristics chosen and the classification error, the fitness function seeks to maximize the classification accuracy. To find a good medium between these two main goals, we use the fitness function that is given below to measure EB-SBOA solutions.

$$fitness = \beta\alpha + \delta\frac{n}{N} \tag{20}$$

Where $\beta \in [0, 1]$, α is the rate of classification error calculated by the kNN or SVM, $\alpha = 1 - \beta$, n represents the chosen features, whereas N is the overall number of features. The kNN or SVM serves as a classifier in our proposed algorithm.

Algorithm 2 illustrates the steps of the proposed algorithm (EB-SBOA).

Algorithm 2 EB-SBOA Algorithm

Initialize parameter (Dim, ub, lb Pop_size (N)), Max_Iter(T), Curr_Iter(t)
Initialize the population using Eqs. (1), (2).
Apply refracted opposition-based learning by Eqs. (17), (18).
for $t = 1 : T$ **do**
  Update Secretary Bird $\mathbf{x_{best}}$
  Transform the Secretary Bird positions into binary space using a transfer function by Eq. (19).
  Evaluate each Secretary Bird within the population using kNN or SVM.
  for $i = 1 : N$ **do**
    Exploration:
    if $t < (T/3)$ **then**
      Calculate new status of the i-th Secretary Bird by Eq. (4).
      Update the i-th Secretary Bird by Eq. (5).
    else if $(T/3) < t < 2 * (T/3)$ **then**
      Calculate new status of the i-th Secretary Bird by Eq. (7).
      Update the i-th Secretary Bird using Eq. (8).
    else
      Calculate new status of the i-th Secretary Bird by Eq. (9).
      Update the i-th Secretary Bird by Eq. (10).
    end if
    Exploitation:
    if $r < 0.5$ **then**
      Calculate new status of the i-th Secretary Bird by p_1 in Eq. (14).
    else
      Calculate new status of the i-th Secretary Bird by p_2 in Eq. (14).
    end if
    Update the i-th Secretary Bird by Eq. (15).
    Apply a Local search algorithm (LSA) on the best solution to find if there is a better solution.
  end for
  Save best candidate solution so far.
end for
return the optimal solution obtained by EB-SBOA.

5. Experiments and Discussion

Our experimental evaluation utilized datasets from the UCI repository [53]. The characteristics of these datasets, including their dimensionality, number of samples, feature count, and classification type, are detailed in Table 1. The computational experiments were performed using MATLAB R2019a on a computer equipped with an Intel Core i7 processor (3.2 GHz) and 8 GB of RAM. The algorithm's parameters were configured with specific values for optimal performance. The implementation involved 30 distinct runs and 100 iterations per run. The search process utilized 10 search agents operating within a feature space defined by the dataset's dimensionality. Two key control parameters were set: β at 0.01 and α at 0.99. For the classification component, the K-neighbors are 5 neighbors, and K-folder cross-validation is 10. The research methodology employs a hold-out validation approach, splitting each dataset into two portions: 80% dedicated to model training and 20% reserved for testing, ensuring result validation. The study evaluates performance [54] using multiple metrics: Average fitness value (AVG), fitness value standard deviation (STD), mean classification accuracy (ACC), average number of selected features (FEA), and computational time in seconds (TIME). The first four of these performance indicators follow specific calculation methods.

The performance metrics are calculated using the following equations:

Table 1. List of datasets

NO	size	Dataset	No. of features	No. of instances	No. of classes
1	Small-Scale < 19	Wine	13	178	Physical
2		Ecolidata	7	335	Life
3		Segmentation	19	210	Life
4		Vote	16	435	Social
5		Australian	14	690	Financial
6		Vehicle	18	846	Life
7		Heart	13	303	Life
8		Diabetes	8	768	Life
9		Letter	16	5000	Computer
10	Medium-Scale [21, 90]	Wpbc	33	198	Life
11		Sonar	60	208	Physical
12		German	24	1000	Business
13		Landsat	36	2000	Physical
14		Waveform	21	5000	Physical
15		Ionosphere	34	351	Physical
16	Large-Scale > 100	Semeion	256	1593	Computer
17		CNAE_9	856	1080	Business
18		Receptor	1024	1678	Physical
19		Msplice	240	3175	Life
20		LSVT	309	126	Life
21		Hillvalley	101	606	Social

$$AVG = \frac{1}{Run} \sum_{r=1}^{Run} G_r \tag{21}$$

$$STD = \sqrt{\frac{\sum_{r=1}^{Run} (G_r - AVG)^2}{Run - 1}} \tag{22}$$

$$ACC = 1 - \frac{1}{Run} \sum_{r=1}^{Run} \frac{\text{error predicted}_r}{\text{Total instances}} \tag{23}$$

$$FEA = \frac{1}{Run} \sum_{r=1}^{Run} |C_r| \tag{24}$$

In these equations, Run represents the total number of experimental runs performed, while r is the order of runs. The variable G_r corresponds to the optimal solution achieved during run r, and $|C_r|$ signifies the quantity of features selected within that particular run.

Table 2. Comparison of EB-SBOA and other algorithms with respect to AVG, STD and TIME

Dataset	Measure	EB-SBOA	BAHA	BAOA	BJSO	BGSA	BMVO	BGWO	BSCA
Wine	AVG	**0.0000**	0.0371	**0.0000**	0.0200	0.0371	0.0557	0.0143	0.0614
Wine	STD	**0.0000**	0.0303	**0.0000**	0.0229	0.0193	0.0376	0.0173	0.0350
Wine	TIME	**21.50**	28.90	40.10	30.20	28.70	28.20	23.70	38.20
Ecolidata	AVG	**0.1001**	0.1388	0.1105	0.1358	0.1388	0.2291	0.1508	0.1351

Dataset	**Measure**	**EB-SBOA**	**BAHA**	**BAOA**	**BJSO**	**BGSA**	**BMVO**	**BGWO**	**BSCA**
Ecolidata	STD	**0.0164**	0.0307	0.0389	0.0436	0.0244	0.0253	0.0453	0.0398
Ecolidata	TIME	30.70	28.70	40.50	28.60	30.10	28.40	**10009.2**	60.10
Segmentation	AVG	**0.0307**	0.1405	0.0857	0.1262	0.1191	0.1262	0.1452	0.1179
Segmentation	STD	**0.0131**	0.0520	0.0340	0.0356	0.0372	0.0558	0.0262	0.0521
Segmentation	TIME	**25.40**	29.40	35.80	28.60	28.90	27.50	19.80	45.40
Vote	AVG	**0.0075**	0.0259	0.0149	0.0270	0.0402	0.0448	0.0494	0.0402
Vote	STD	0.0106	0.0129	0.0133	0.0141	**0.0098**	0.0175	0.0224	0.0156
Vote	TIME	**22.30**	31.10	42.30	31.40	30.60	29.10	25.90	42.10
Australian	AVG	**0.0554**	0.1362	0.1145	0.1529	0.2051	0.1996	0.1833	0.1848
Australian	STD	**0.0132**	0.0193	0.0264	0.0238	0.0345	0.0370	0.0331	0.0364
Australian	TIME	30.00	32.10	38.50	33.80	32.60	30.70	**22.60**	44.50
Vehicle	AVG	**0.1770**	0.2518	0.2243	0.2607	0.2459	0.2926	0.2503	0.2908
Vehicle	STD	**0.0133**	0.0194	0.0190	0.0186	0.0209	0.0291	0.0197	0.0253
Vehicle	TIME	35.40	325.72	37.60	318.21	320.69	308.78	**23.50**	52.50
Heart	AVG	**0.0644**	0.0967	0.1017	0.1100	0.1317	0.1467	0.1050	0.1375
Heart	STD	0.0270	0.0299	0.0328	0.0313	**0.0254**	0.0369	0.0387	0.0346
Heart	TIME	23.50	**20.10**	42.00	30.10	29.50	28.40	21.60	40.00
Diabetes	AVG	**0.1843**	0.2242	0.2039	0.2255	0.2186	0.2366	0.2301	0.2395
Diabetes	STD	**0.0141**	0.0204	0.0198	0.0319	0.0218	0.0226	0.0176	0.0195
Diabetes	TIME	**16.10**	31.30	39.50	32.50	30.90	18.00	21.70	50.90
Letter	AVG	0.1095	0.1112	**0.0964**	0.1262	0.1112	0.1455	0.1288	0.1443
Letter	STD	**0.0057**	0.0118	0.0094	0.0082	0.0112	0.0234	0.0121	0.0195
Letter	TIME	**55.05**	69.70	101.20	70.50	76.20	74.90	65.70	493.20
Wpbc	AVG	**0.0225**	0.0487	0.0283	0.0429	0.1590	0.1641	0.1769	0.1808
Wpbc	STD	0.0473	0.0248	0.0166	0.0158	**0.0315**	0.0376	0.0730	0.0549
Wpbc	TIME	22.90	29.20	42.30	28.70	27.30	26.50	**19.30**	43.60
Sonar	AVG	**0.0489**	0.1220	0.0659	0.0976	0.1122	0.1427	0.1439	0.1232
Sonar	STD	0.1767	0.0430	**0.0275**	0.0363	0.0309	0.0463	0.0480	0.0528
Sonar	TIME	**21.30**	28.20	39.50	28.60	27.90	27.60	23.30	37.30
German	AVG	**0.1772**	0.2300	0.2110	0.2320	0.2460	0.2568	0.2445	0.2555
German	STD	**0.0133**	0.0197	0.0248	0.0175	0.0216	0.0242	0.0258	0.0179
German	TIME	**25.70**	32.70	46.00	33.90	31.50	32.10	23.90	53.10
Landsat	AVG	**0.0474**	0.1008	0.0885	0.1033	0.1040	0.1110	0.1043	0.1070
Landsat	STD	0.0097	0.0121	0.0096	**0.0090**	0.0129	0.0107	0.0112	0.0129
Landsat	TIME	**30.00**	39.90	58.10	41.80	40.60	41.60	39.50	68.10
Waveform	AVG	**0.1204**	0.1705	0.1581	0.1717	0.1652	0.1845	0.1773	0.1879
Waveform	STD	**0.0033**	0.0085	0.0071	0.0087	0.0072	0.0133	0.0058	0.0100
Waveform	TIME	**65.00**	78.00	110.50	78.90	81.50	84.50	76.70	203.30
Ionosphere	AVG	**0.0228**	0.0729	0.0486	0.0893	0.0721	0.1286	0.0679	0.1179
Ionosphere	STD	0.0258	0.0270	0.0217	0.0412	0.0188	0.0307	**0.0212**	0.0393
Ionosphere	TIME	29.60	28.50	40.20	28.70	**28.40**	26.90	63.20	38.60
Semeion	AVG	0.0871	0.0899	**0.0796**	0.0918	0.0899	0.1072	0.0833	0.1022
Semeion	STD	**0.0100**	0.0165	0.0119	0.0161	0.0149	0.0138	**0.0100**	0.0128
Semeion	TIME	**40.80**	57.50	142.00	64.10	94.80	47.10	66.10	35.80
CNAE_9	AVG	**0.1108**	0.1232	0.1352	0.1741	0.1120	0.2394	0.1815	0.2204
CNAE_9	STD	0.0117	0.0203	0.0222	0.0220	**0.0160**	0.0380	0.0253	0.0339
CNAE_9	TIME	**56.40**	64.20	153.50	71.80	84.60	69.80	112.90	60.60
Receptor	AVG	**0.0501**	0.0795	0.0721	0.0812	0.0880	0.0822	0.0960	0.0941
Receptor	STD	**0.0033**	0.0177	0.0092	0.0136	0.0046	0.0078	0.0180	0.0143

Dataset	Measure	EB-SBOA	BAHA	BAOA	BJSO	BGSA	BMVO	BGWO	BSCA
Receptor	TIME	**120.60**	181.50	355.10	145.20	184.90	138.20	212.50	142.90
Msplice	AVG	**0.0804**	0.1353	0.1196	0.1375	0.1441	0.1676	0.1328	0.1635
Msplice	STD	**0.0066**	0.0101	0.0089	0.0095	0.0140	0.0172	0.0087	0.0222
Msplice	TIME	205.60	**188.60**	285.90	244.90	232.60	228.90	232.20	610.80
LSVT	AVG	**0.2060**	0.3120	0.2360	0.3040	0.3200	0.2920	0.3200	0.2960
LSVT	STD	0.0579	0.0867	0.0765	0.0727	0.0943	0.0755	0.0938	**0.0430**
LSVT	TIME	**16.70**	19.20	39.00	17.40	14.20	18.10	17.40	19.50
Hillvalley	AVG	**0.2604**	0.4372	0.3893	0.4116	0.4248	0.4380	0.4215	0.4422
Hillvalley	STD	**0.0207**	0.0315	0.0478	0.0338	0.0440	0.0591	0.0691	0.0360
Hillvalley	TIME	**11.80**	14.20	32.50	19.20	13.70	17.80	25.60	15.10

The Table 2 presents a comprehensive comparison between the proposed algorithm (EB-SBOA) and seven algorithms: BSCA [55], BGWO [56], BMVO [57], BGSA [58], BJSO [59], BAOA [13] and BAHA [60] in AVG, STD and TIME. The comparative analysis presented in Table 2 reveals that EB-SBOA consistently demonstrates superior performance with lower average error rates across most datasets. The EB-SBOA algorithm has emerged as a notable contribution, showing remarkable performance across multiple evaluation metrics. When examining the Wine dataset, both EB-SBOA and BAOA achieved optimal performance with an average error rate of 0.0000, surpassing other contemporary approaches such as BAHA and BMVO, which showed error rates of 0.0371 and 0.0557 respectively. The execution time analysis presents interesting patterns. For instance, in the German dataset, EB-SBOA completed its execution in 25.7000 units, while BAOA required 46.0000 units. This significant difference in computational overhead highlights the importance of algorithm efficiency in practical applications. The stability of these algorithms, as measured by standard deviation, reveals another crucial aspect of their performance. EB-SBOA consistently maintained lower standard deviation values across multiple datasets, indicating more reliable and predictable performance. This is particularly evident in the Vehicle dataset, where EB-SBOA achieved a standard deviation of 0.0133, while other algorithms such as BMVO and BSCA showed higher variations of 0.0291 and 0.0253 respectively. Complex datasets such as Hillvalley and CNAE_9 provided particularly challenging test cases. In these instances, EB-SBOA demonstrated notably better performance with average error rates of 0.2604 and 0.1108 respectively, compared to substantially higher error rates from other algorithms. This performance gap becomes more pronounced as dataset complexity increases, suggesting that EB-SBOA's optimization strategy is particularly effective for handling complex problem spaces.

Table 3. Comparison of EB-SBOA and recent algorithms with respect to ACC and FEA

Dataset	Measure	EB-SBOA	BAHA	BAOA	BJSO	BGSA	BMVO	BGWO	BSCA
Wine	ACC	**100.00**	96.29	**100.00**	98.00	96.29	94.43	98.57	93.86
Wine	FEA	**5.11**	6.90	5.50	5.35	6.80	5.95	5.70	6.40
Ecolidata	ACC	90.74	86.12	**88.96**	84.17	86.12	77.09	84.93	86.49
Ecolidata	FEA	**4.10**	5.00	4.60	6.00	6.00	5.10	6.00	5.05
Segmentation	ACC	88.52	85.95	**91.43**	87.38	88.10	87.38	85.48	88.21
Segmentation	FEA	**6.40**	9.80	8.20	12.30	10.80	9.20	10.80	9.95
Vote	ACC	**99.57**	97.41	98.51	97.30	96.11	95.52	95.06	95.98
Vote	FEA	8.70	6.70	**6.10**	6.95	9.10	7.00	8.00	8.20
Australian	ACC	**91.84**	86.38	88.55	84.71	79.49	80.04	81.67	81.52
Australian	FEA	**3.40**	4.50	4.50	4.30	6.00	6.25	5.50	5.65
Vehicle	ACC	75.64	74.82	**77.57**	73.93	75.41	70.74	74.97	70.92
Vehicle	FEA	**6.20**	7.70	7.20	9.65	9.25	9.25	8.55	8.90
Heart	ACC	**94.64**	90.33	89.83	89.00	86.83	85.33	89.50	86.25
Heart	FEA	7.30	6.30	6.50	7.00	7.80	7.40	**6.25**	7.85
Diabetes	ACC	**83.66**	77.58	79.61	77.45	78.14	76.34	76.99	76.05

Dataset	**Measure**	**EB-SBOA**	**BAHA**	**BAOA**	**BJSO**	**BGSA**	**BMVO**	**BGWO**	**BSCA**
Diabetes	FEA	**3.70**	5.30	4.40	4.25	4.80	5.10	5.30	4.85
Letter	ACC	**93.40**	88.89	90.36	87.38	88.88	85.46	87.13	85.57
Letter	FEA	10.47	11.40	**9.90**	13.40	13.80	10.65	9.95	10.65
Wpbc	ACC	**96.42**	80.00	89.74	86.15	84.10	83.59	82.31	94.87
Wpbc	FEA	**11.40**	14.70	12.20	17.20	20.70	16.95	16.10	15.55
Sonar	ACC	**95.01**	87.80	93.41	90.24	88.78	85.73	85.61	87.68
Sonar	FEA	**20.50**	31.20	22.10	28.55	35.10	29.60	29.70	28.85
German	ACC	75.54	**77.00**	78.90	76.80	75.40	74.33	75.55	74.45
German	FEA	**9.94**	10.80	10.70	12.35	15.30	12.60	11.40	12.60
Landsat	ACC	90.31	89.93	**91.15**	89.68	89.60	88.90	89.58	89.30
Landsat	FEA	17.55	21.90	**15.20**	18.00	27.20	19.30	18.90	18.65
Waveform	ACC	**86.65**	82.95	84.19	85.74	83.49	81.56	82.28	81.21
Waveform	FEA	**10.80**	14.60	12.10	14.70	14.70	13.25	13.55	13.80
Ionosphere	ACC	**96.87**	92.71	95.14	92.79	92.79	87.14	93.21	88.21
Ionosphere	FEA	9.50	11.05	**7.00**	13.15	13.15	16.65	10.90	16.45
Semeion	ACC	**94.69**	91.01	92.04	90.82	91.01	89.28	91.67	89.78
Semeion	FEA	130.40	167.90	**128.30**	129.10	135.40	136.10	129.60	133.50
CNAE_9	ACC	86.65	87.69	86.48	82.59	**88.80**	76.06	81.85	77.96
CNAE_9	FEA	**390.50**	804.00	421.80	415.90	830.90	430.60	438.50	435.90
Receptor	ACC	**94.10**	92.05	92.79	91.88	91.20	91.78	90.40	90.59
Receptor	FEA	**385.37**	567.60	406.70	458.40	516.40	519.50	520.50	510.10
Msplice	ACC	**91.01**	86.47	88.04	86.25	85.59	83.24	86.72	83.65
Msplice	FEA	105.60	102.85	**99.20**	115.70	122.00	125.80	100.15	122.25
LSVT	ACC	**79.70**	68.80	76.40	69.60	68.00	70.80	68.00	70.40
LSVT	FEA	**62.10**	124.80	75.50	136.00	158.80	157.00	151.40	155.80
Hillvalley	ACC	**70.80**	56.28	61.07	58.84	57.52	56.20	57.85	55.79
Hillvalley	FEA	**30.30**	42.90	35.00	43.00	49.00	41.80	50.40	51.10

The Table 3 presents a comprehensive comparison of eight algorithms: EB-SBOA, BAHA, BAOA, BJSO, BGSA, BMVO, BGWO, and BSCA in ACC and FEA. The comparative analysis reveals significant performance variations across different datasets and algorithms. In the Wine dataset, EB-SBOA and BAOA achieved perfect classification accuracy (100%), substantially outperforming other algorithms such as BSCA and BMVO, which achieved 93.8571% and 94.4286% respectively. This superior performance was achieved with a relatively efficient feature evaluation accuracy of 5.1084 for EB-SBOA, compared to higher FEA values for other algorithms. For complex datasets such as Ecolidata and Segmentation, the performance dynamics show interesting patterns. While BAOA achieved the highest accuracy in these cases (88.9552% and 91.4286% respectively), EB-SBOA demonstrated the most efficient feature evaluation with FEA values of 4.1000 and 6.4000, significantly lower than other algorithms. This suggests that EB-SBOA achieves competitive accuracy while requiring fewer feature evaluations. In the context of larger datasets like Letter and Waveform, EB-SBOA demonstrated remarkable performance. For the Letter dataset, it achieved 93.4007% accuracy with a competitive FEA of 10.4741, while for Waveform, it reached 86.6541% accuracy with an FEA of 10.8000, outperforming other algorithms in both accuracy and efficiency. The analysis of more challenging datasets such as CNAE_9 and Receptor reveals EB-SBOA's robustness. In the Receptor dataset, EB-SBOA achieved the highest accuracy of 94.1001% with the most efficient feature evaluation (385.3650), significantly outperforming other algorithms. However, in the CNAE_9 dataset, BGSA showed slightly better accuracy (88.7963%), though EB-SBOA maintained more efficient feature evaluation.

Figure 4 depicts results of ACC of the proposed algorithm (EB-SBOA) compared with recent algorithms over Small-Scale datasets. The results of ACC of the proposed algorithm (EB-SBOA) compared with recent algorithms

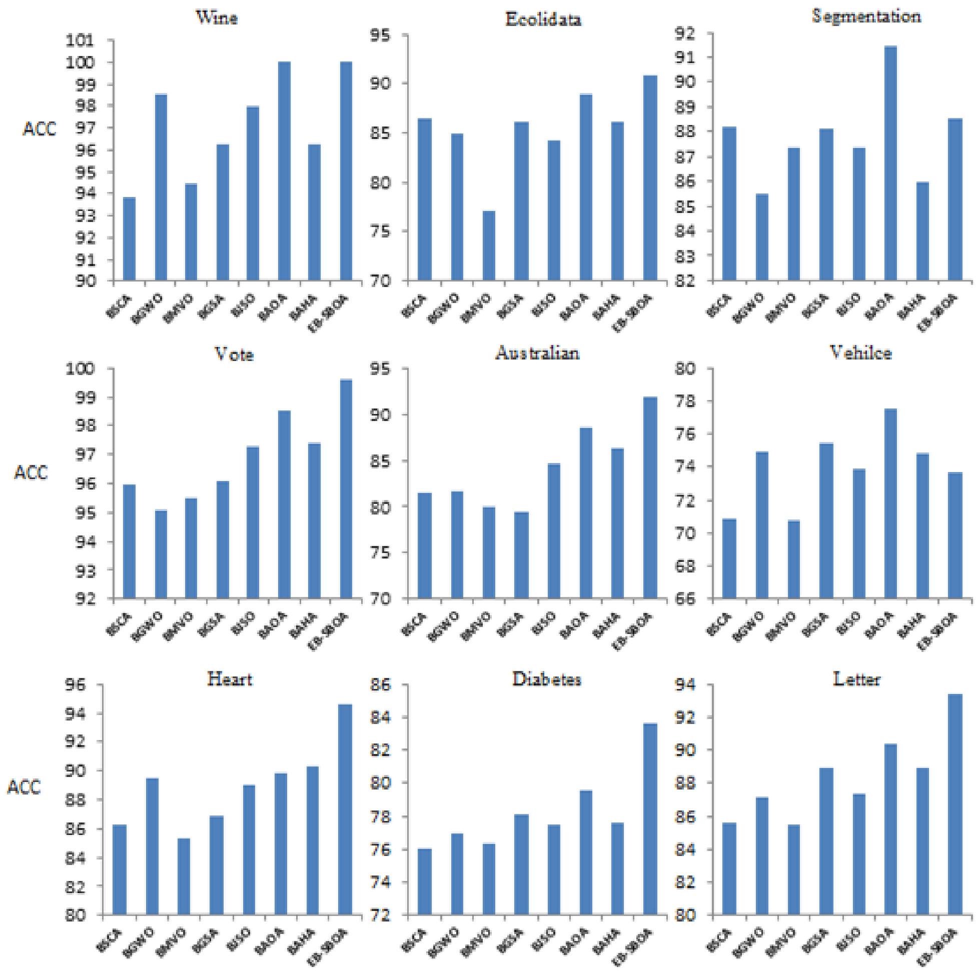

Figure 4. ACC of EB-SBOA compared with recent algorithms over Small-Scale datasets.

over Medium-Scale datasets show in Figure 5. Figure 6 depicts results of ACC of the proposed algorithm (EB-SBOA) compared with recent algorithms over Large-Scale datasets.

The Analysis of Variance (ANOVA) results reveal significant statistical differences between EB-SBOA and other algorithms. In the Wine dataset, the analysis shows highly significant differences between EB-SBOA and most algorithms, particularly with BAOA (p = 1.16E-05) and BGWO (p = 7.74E-04), while the comparison with BJSO shows no significant difference (p = 3.41E-01, p ¿ 0.05). For complex datasets such as Australian and Vehicle, the statistical significance is particularly pronounced. The Australian dataset shows extremely significant differences between EB-SBOA and BMVO (p = 8.02E-07) and BGWO (p = 6.07E-06). Similarly, the Vehicle dataset demonstrates highly significant differences with BMVO (p = 8.01E-07) and BAOA (p = 1.47E-03). The analysis of larger datasets like Letter and Waveform presents compelling evidence of algorithmic differences. In the Letter dataset, the differences are highly significant across most algorithms, with particularly strong significance against BGWO (p = 6.00E-07) and BMVO (p = 8.05E-06). The Waveform dataset shows consistent statistical significance across all algorithms, with p-values ranging from 6.00E-05 to 9.83E-03. In specialized datasets such as Hillvalley and LSVT, the statistical differences are notably strong. The Hillvalley dataset shows extremely significant differences with BSCA (p = 1.59E-08) and BGSA (p = 4.95E-04). The LSVT dataset demonstrates significant differences across all algorithms, with particularly strong significance against BMVO (p = 5.33E-05) and BSCA (p = 4.73E-04). The Receptor and Msplice datasets present interesting patterns of statistical significance.

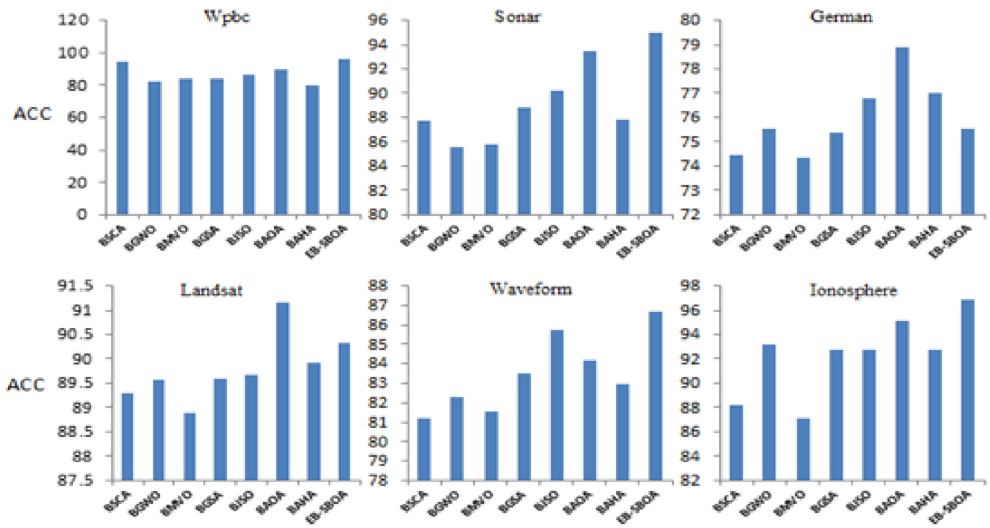

Figure 5. ACC of EB-SBOA compared with recent algorithms over Medium-Scale datasets.

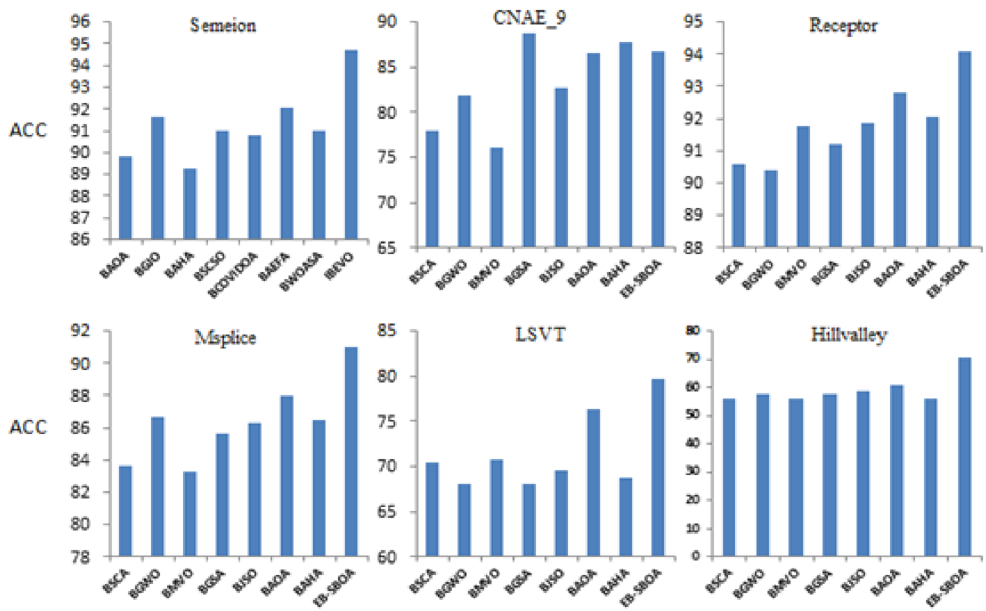

Figure 6. ACC of EB-SBOA compared with recent algorithms over Large-Scale datasets.

In the Receptor dataset, the strongest differences are observed with BGWO (p = 2.17E-06) and BSCA (p = 4.68E-05), while the Msplice dataset shows highly significant differences with BAOA (p = 4.07E-04) and BSCA (p = 8.14E-06).

The Wilcoxon rank sum test results reveal significant statistical differences in classification accuracy between EB-SBOA and other algorithms. For the Wine dataset, the analysis shows highly significant differences across all algorithms, with particularly strong significance against BAOA (p = 4.70E-04) and BJSO (p = 4.54E-04), indicating substantial performance differences in this classification task. When examining complex datasets such as Ecolidata and Segmentation, the statistical significance varies notably. In Ecolidata, the strongest significance is observed with BMVO (p = 1.01E-04), while BSCA shows no significant difference (p = 4.61E-01, p ¿ 0.05). The Segmentation dataset demonstrates significant differences with BAHA (p = 3.08E-03) and BGSA (p = 3.01E-03), suggesting distinct performance characteristics.

Table 4. p-values of Analysis of Variance for the classification accuracy of EB-SBOA and recent algorithms.

Dataset	BAHA	BAOA	BJSO	BGSA	BMVO	BGWO	BSCA
Wine	9.12E-03	1.16E-05	3.41E-01	5.00E-03	1.04E-02	7.74E-04	2.51E-03
Ecolidata	5.17E-02	1.23E-03	4.86E-03	5.28E-02	8.00E-03	1.57E-01	4.39E-02
Segmentation	2.12E-02	3.47E-02	3.94E-01	4.27E-01	5.70E-02	6.36E-03	3.05E-01
Vote	3.67E-02	3.66E-02	5.43E-02	5.13E-03	8.22E-05	6.55E-01	3.22E-03
Australian	3.95E-01	1.06E-02	1.09E-02	5.00E-03	8.02E-07	6.07E-06	4.11E-03
Vehicle	2.07E-02	1.47E-03	2.65E-03	4.01E-02	8.01E-07	9.30E-03	8.00E-03
Heart	2.87E-01	3.18E-02	4.99E-04	4.14E-05	2.00E-03	6.39E-01	1.01E-02
Diabetes	9.45E-02	5.55E-03	3.05E-02	1.48E-02	3.93E-02	2.86E-02	1.27E-02
Letter	2.00E-02	3.40E-02	7.38E-03	6.70E-03	8.05E-06	6.00E-07	4.01E-03
Wpbc	9.47E-03	2.11E-05	4.22E-02	1.13E-02	1.95E-05	2.11E-03	1.65E-06
Sonar	1.50E-02	6.00E-02	1.19E-01	1.14E-02	8.18E-03	8.36E-04	1.90E-04
German	2.70E-02	1.40E-03	1.19E-04	8.40E-03	8.26E-02	1.43E-01	3.12E-03
Landsat	2.85E-02	1.35E-02	3.42E-01	1.11E-02	8.09E-01	8.90E-03	6.60E-04
Waveform	9.83E-03	7.66E-03	1.07E-02	1.07E-02	8.03E-03	6.00E-05	2.00E-02
Ionosphere	4.97E-02	7.37E-03	1.44E-02	1.16E-04	8.02E-02	1.40E-02	4.09E-04
Semeion	3.28E-02	6.71E-02	2.09E-01	4.81E-02	8.14E-02	4.57E-06	2.66E-03
CNAE_9	3.28E-02	6.33E-03	1.23E-02	2.06E-01	8.00E-04	6.39E-06	1.00E-01
Receptor	2.63E-01	3.44E-02	8.64E-01	4.98E-01	2.43E-04	2.17E-06	4.68E-05
Msplice	1.04E-02	4.07E-04	5.09E-03	5.39E-03	8.00E-01	1.17E-01	8.14E-06
LSVT	3.90E-02	1.10E-02	4.70E-02	4.71E-02	5.33E-05	4.16E-04	4.73E-04
Hillvalley	2.54E-02	2.67E-03	5.69E-03	4.95E-04	1.24E-02	2.82E-03	1.59E-08

The analysis of larger datasets such as Letter and Waveform presents compelling evidence of algorithmic differences. The Letter dataset shows consistent statistical significance across all algorithms, with particularly strong differences against BGSA (p = 2.47E-05) and BSCA (p = 4.09E-04). In the Waveform dataset, significant differences are observed with BAOA (p = 1.08E-04) and BGSA (p = 1.06E-03). For specialized datasets like CNAE_9 and Receptor, the test reveals interesting patterns. CNAE_9 shows highly significant differences with BGSA (p = 1.06E-04) and BAOA (p = 2.70E-04), while showing no significant difference with BSCA (p = 9.00E-01). The Receptor dataset demonstrates strong significance particularly with BSCA (p = 1.56E-04), while showing less significant differences with other algorithms. In the more challenging datasets of LSVT and Hillvalley, the statistical differences remain notable. LSVT shows strong significance with BAOA (p = 3.51E-04) and BGSA (p = 4.50E-04), while Hillvalley demonstrates particularly significant differences with BAHA (p = 4.12E-04) and notable differences across other algorithms.

.

6. Conclusion

This research introduces a novel version of the Secretary Bird Optimization Algorithm, called EB-SBOA, specifically designed to address feature selection challenges. The enhancement of the original SBOA incorporates two key modifications: a refracted opposition-based learning technique during the initialization phase to enhance population diversity, and an innovative random replacement strategy to achieve more precise convergence. The implementation utilizes either kNN or SVM classifiers, which have demonstrated remarkable effectiveness in generating high-quality solutions and extracting meaningful patterns from training data. To mitigate over fitting

Table 5. p-values of Wilcoxon rank sum test for the classification accuracy of EB-SBOA and recent algorithms.

Dataset	BAHA	BAOA	BJSO	BGSA	BMVO	BGWO	BSCA
Wine	1.87E-02	4.70E-04	4.54E-04	9.20E-03	1.61E-02	4.19E-02	1.12E-02
Ecolidata	3.73E-02	3.33E-03	2.87E-02	2.48E-02	1.01E-04	4.47E-02	4.61E-01
Segmentation	3.08E-03	4.48E-01	5.39E-02	3.01E-03	2.95E-02	1.92E-02	3.51E-02
Vote	5.43E-03	5.53E-01	3.43E-03	1.54E-01	1.24E-02	3.62E-02	1.63E-05
Australian	3.08E-02	2.98E-03	1.48E-01	8.10E-03	5.12E-01	1.49E-02	4.18E-03
Vehicle	1.18E-04	4.08E-02	8.60E-03	6.50E-04	5.10E-03	9.70E-03	4.09E-01
Heart	2.29E-04	4.09E-03	1.39E-05	5.26E-02	3.33E-02	8.24E-03	1.05E-02
Diabetes	5.28E-02	1.38E-04	3.54E-02	2.48E-02	1.87E-04	3.64E-02	5.40E-03
Letter	9.36E-03	7.62E-03	2.84E-03	2.47E-05	5.10E-03	8.09E-03	4.09E-04
Wpbc	1.83E-02	1.40E-02	2.14E-03	1.23E-02	6.30E-03	2.97E-02	4.51E-03
Sonar	1.58E-05	7.70E-04	3.52E-02	4.93E-01	1.25E-02	2.05E-02	4.61E-02
German	5.54E-02	5.65E-03	5.64E-04	4.44E-02	5.63E-03	2.02E-02	4.13E-03
Landsat	5.11E-01	3.71E-01	3.91E-04	3.50E-03	2.27E-02	4.02E-02	7.80E-04
Waveform	1.98E-01	1.08E-04	1.63E-02	1.06E-03	1.19E-02	1.31E-02	9.00E-03
Ionosphere	5.84E-02	4.44E-03	2.16E-03	4.06E-02	1.19E-04	2.97E-02	1.64E-05
Semeion	3.74E-03	3.04E-01	3.15E-01	4.99E-01	1.19E-02	4.62E-02	2.06E-02
CNAE_9	2.61E-03	2.70E-04	1.20E-02	1.06E-04	1.01E-01	1.29E-02	9.00E-01
Receptor	4.52E-01	2.33E-01	9.40E-01	3.84E-02	2.56E-02	5.69E-02	1.56E-04
Msplice	9.33E-03	1.55E-03	9.97E-02	9.90E-03	5.09E-03	9.60E-03	4.09E-03
LSVT	8.40E-02	3.51E-04	9.70E-03	4.50E-04	3.16E-02	1.93E-02	5.40E-02
Hillvalley	4.12E-04	7.72E-03	4.37E-02	3.97E-02	3.33E-03	5.49E-02	2.58E-02

concerns, the methodology employs k-fold cross-validation. The algorithm's performance was rigorously evaluated using twenty-one distinct datasets and benchmarked against established feature selection algorithms. The EB-SBOA effectiveness is particularly notable in its balanced approach to exploration and exploitation, enhanced by the incorporation of refracted opposition-based learning and local search mechanisms. The proposed algorithm (EB-SBOA) demonstrates superior performance in feature selection and classification tasks across multiple benchmark datasets. The algorithm's performance is particularly strong in handling complex datasets, demonstrating its scalability and practical applicability in real-world scenarios. The statistical validation through p-values confirms the significance of these improvements, with many comparisons showing p-values < 0.05, establishing EB-SBOA as a valuable contribution to the field of metaheuristic optimization and feature selection. These results suggest that EB-SBOA offers a promising approach for addressing the challenges of high-dimensional data classification while maintaining computational efficiency.

REFERENCES

1. Jing, Li-Ping, Hou-Kuan Huang, and Hong-Bo Shi. "Improved feature selection approach TFIDF in text mining." *Proceedings. International Conference on Machine Learning and Cybernetics*. Vol. 2. IEEE, 2002.
2. Al-Sharo, Yasser Mohammad, Khalid Al Smadi, and Takialddin Al Smadi. "Optimization of Stable Energy PV Systems Using the Internet of Things (IoT)." *Tikrit Journal of Engineering Sciences* 31.1: 127-137, 2024.
3. Elmanakhly, Dina A., Mohamed Mostafa Saleh, and Essam A. Rashed. "An improved equilibrium optimizer algorithm for features selection: methods and analysis." *IEEE Access* 9: 120309-120327, 2021.
4. K. Huang, S. Aviyente, "Wavelet feature selection for image classification," *IEEE Trans. Image Process.* 17 (9): 1709–1720, 2008.
5. Ghaddar, Bissan, and Joe Naoum-Sawaya. "High dimensional data classification and feature selection using support vector machines." *European Journal of Operational Research* 265.3: 993-1004, 2018.
6. Faris, Hossam, et al. "An efficient binary salp swarm algorithm with crossover scheme for feature selection problems." *Knowledge-Based Systems* 154: 43-67, 2018.
7. Jain, Anil K., Robert P. W. Duin, and Jianchang Mao. "Statistical pattern recognition: A review." *IEEE Transactions on pattern analysis and machine intelligence* 22.1: 4-37, 2000.
8. Dasarathy, Belur V. "Nearest neighbor (NN) norms: NN pattern classification techniques." *IEEE Computer Society Tutorial*, 1991.
9. Emary, Eid, Hossam M. Zawbaa, and Aboul Ella Hassanien. "Binary ant lion approaches for feature selection." *Neurocomputing* 213: 54-65, 2016.
10. Kuzudisli, Cihan, et al. "Review of feature selection approaches based on grouping of features." *PeerJ* 11: e15666, 2023.

11. Alirezapour, Hanie, Najme Mansouri, and Behnam Mohammad Hasani Zade. “A Comprehensive Survey on Feature Selection with Grasshopper Optimization Algorithm.” *Neural Processing Letters* 56.1: 28, 2024.
12. Emary, Eid, Hossam M. Zawbaa, and Aboul Ella Hassanien. “Binary grey wolf optimization approaches for feature selection.” *Neurocomputing* 172: 371-381, 2016.
13. Fu, Youfa, et al. “Secretary bird optimization algorithm: a new metaheuristic for solving global optimization problems.” *Artificial Intelligence Review* 57.5: 1-102, 2024.
14. G. Bissan, and J. Naoum. “High dimensional data classification and feature selection using support vector machines,” *European Journal of Operational Research* 265(3): 993-1004, 2018.
15. F. Hossam “An efficient binary salp swarm algorithm with crossover scheme for feature selection problems,” *Knowledge-Based Systems* 154: 43-67, 2018.
16. J. K., Robert, and J. Mao. “Statistical pattern recognition: A review,” *IEEE Transactions on pattern analysis and machine intelligence* 22.1: 4-37, 2000.
17. Belur V., and Dasarathy, “Nearest neighbor (NN) norms: NN pattern classification techniques,” *IEEE Computer Society Tutorial*, 1991.
18. K. Hosny, and A Mohamed, “BCOVIDOA: a novel binary coronavirus disease optimization algorithm for feature selection,” *Knowledge-based systems* 248: 108789, 2022.
19. E. Eid, and A. Hassanien. “Binary grey wolf optimization approaches for feature selection,” *Neurocomputing* 172: 371-381, 2013.
20. K. Ali, and N. Farhoudi. “A new optimization method: Dolphin echolocation,” *Advances in Engineering Software* 59: 53-70, 2023.
21. R. Venkata, and D. Vakharia. “Teaching–learning-based optimization: a novel method for constrained mechanical design optimization problems,” *Computer-aided design* 43.3: 303-315, 2011.
22. R. Esmat, and S. Saryazdi. “GSA: a gravitational search algorithm,” *Information sciences* 179.13: 2232-2248, 2009.
23. H. Nikolaus, and D. Müller, “Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (CMA-ES),” *Evolutionary computation* 11.1: 1-18, 2003.
24. R. Douglas, “Binary flower pollination algorithm and its application to feature selection,” *Recent advances in swarm intelligence and evolutionary computation*: 85-100, 2015.
25. E. Dina, “BinHOA: Efficient binary horse herd optimization method for feature selection: Analysis and validations,” *IEEE Access* 10: 26795-26816, 2002.
26. M. Majdi, “Binary dragonfly algorithm for feature selection,” *International conference on new trends in computing sciences (ICTCS)*, IEEE, 2017.
27. K. Ali, and N. Farhoudi. “A new optimization method: Dolphin echolocation,” *Advances in Engineering Software* 59: 53-70, 2013.
28. X. Bing, and N. Browne. “Particle swarm optimization for feature selection in classification: Novel initialization and updating mechanisms,” *Applied soft computing* 18: 261-276, 2014.
29. Pernkopf, Franz. “Bayesian network classifiers versus selective k-NN classifier.” *Pattern recognition* 38.1: 1-10, 2005.
30. A. Prachi, and A. Mohamed, “A novel binary gaining–sharing knowledge-based optimization algorithm for feature selection,” *Neural Computing and Applications* 33.11: 5989-6008, 2021.
31. Hashim, Fatma A., et al. “Henry gas solubility optimization: A novel physics-based algorithm.” *Future Generation Computer Systems* 101: 646-667, 2019.
32. H. Seyedmohsen, and A. Khaled. “A survey on the imperialist competitive algorithm metaheuristic: implementation in engineering domain and directions for future research,” *Applied Soft Computing* 24: 1078-1094, 2024.
33. K. Cio, and C. Lin. “Cultural evolution algorithm for global optimizations and its applications,” *Journal of applied research and technology* 11.4: 510-522, 2013.
34. M. Reza, and K. Salimifard, “Volleyball premier league algorithm,” *Applied Soft Computing* 64: 161-185, 2018.
35. R. Venkata, and D. Vakharia, “Teaching–learning-based optimization: a novel method for constrained mechanical design optimization problems,” *Computer-aided design* 43.3: 303-315, 2011.
36. Faramarzi, Afshin, et al. “Equilibrium optimizer: A novel optimization algorithm.” *Knowledge-Based Systems* 191: 105190, 2020.
37. Sh. Ibrahim and A. Mutlag. “Lightning search algorithm,” *Applied Soft Computing* 36: 315-333, 2015.
38. M. Seyedali, and A. Hatamlou. “Multi-verse optimizer: a nature-inspired algorithm for global optimization,” *Neural Computing and Applications* 27: 495-513, 2016.
39. H. Fatma, “Henry gas solubility optimization: A novel physics-based algorithm,” *Future Generation Computer Systems* 101: 646-667, 2019.
40. T. Mohammad, “An evolutionary gravitational search-based feature selection,” *Information Sciences* 497: 219-239, 2019.
41. H. Sajedi “Flash-flood hazard assessment using ensembles and Bayesian-based machine learning models: Application of the simulated annealing feature selection method,” *Science of the total environment* 711: 135161, 2020.
42. Elmanakhly, Dina A., Mohamed Mostafa Saleh, and Essam A. Rashed. “An improved equilibrium optimizer algorithm for features selection: methods and analysis.” *IEEE Access* 9: 120309-120327, 2021.
43. K. Manoj, “Genetic algorithm: Review and application,” *Available at SSRN* 3529843, 2010.
44. Y. Gao, Y. Zhou, and Q. Luo, “An efficient binary equilibrium optimizer algorithm for feature selection,” *IEEE Access*, vol. 8, pp. 140936–140963, 2021.
45. R. A. Khurma, I. Aljarah, and A. Sharieh, “An efficient moth flame optimization algorithm using chaotic maps for feature selection in the medical applications,” in *Proc. ICPRAM*, 2020, pp. 175–182.
46. Z. Yong, “Binary differential evolution with self-learning for multi-objective feature selection,” *Information Sciences* 507: 67-85, 2020.
47. S. Dan, “Biogeography-based optimization,” *IEEE transactions on evolutionary computation* 12.6: 702-713, 2008.
48. K. Soheyl and S. Khalilpourazary “Multi-objective stochastic fractal search: A powerful algorithm for solving complex multi-objective optimization problems,” *Soft Computing* 24: 3037-3066, 2020.
49. Larson ER, Olden JD “the state of crayfsh in the Pacifc Northwest.” *Fisheries* 36(2): 60–73, 2011.

50. E.-S. M. El-Kenawy, M. M. Eid, M. Saber, and A. Ibrahim, “MbGWOSFS: Modified binary grey wolf optimizer based on stochastic fractal search for feature selection,” *IEEE Access*, vol. 8, pp. 107635–107649, 2020.
51. Zhao, F., Zhang, L., Zhang, Y., Ma, W., Zhang, C., & Song, H. “An improved water wave optimisation algorithm enhanced by CMAES and opposition-based learning.” *Connection Science* 32(2): 132–161, 2020.
52. M. Seyedali, L. Andrew, “S-shaped versus V-shaped transfer functions for binary particle swarm optimization,” *Swarm Evol. Comput.* 9: 1–14, 2013.
53. Lichman M “UCI machine learning repository.” http://archive.ics.uci.edu/ml, 2013.
54. Too, Jingwei, Abdul Rahim Abdullah, and Norhashimah Mohd Saad. “A new quadratic binary harris hawk optimization for feature selection.” *Electronics* 8.10: 1130, 2019.
55. Reddy, K. Srikanth, et al. “A new binary variant of sine–cosine algorithm: development and application to solve profit-based unit commitment problem.” *Arabian Journal for Science and Engineering* 43: 4041-4056, 2018.
56. Hu, Pei, Jeng-Shyang Pan, and Shu-Chuan Chu. “Improved binary grey wolf optimizer and its application for feature selection.” *Knowledge-Based Systems* 195: 105746, 2020.
57. Al-Madi N, Faris H, Mirjalili S “Binary multi-verse optimization algorithm for global optimization and discrete problems.” *Int J Mach Learn Cybern* 10(12): 3445–3465, 2019.
58. Taradeh, Mohammad, et al. “An evolutionary gravitational search-based feature selection.” *Information Sciences* 497: 219-239, 2019.
59. Chou JS, Truong DN “A novel metaheuristic optimizer inspired by behavior of jellyfish in ocean.” *Appl Math Comput* 389: 125535, 2021.
60. Zhao W, Wang L, Mirjalili S “Artificial hummingbird algorithm: a new bio-inspired optimizer with its engineering applications.” *Comput Methods Appl Mech Eng* 388: 114194, 2022.
61. Fathi, Islam S., and Mohammed Tawfik. “Enhancing IoT Systems with Bio-Inspired Intelligence in fog computing environments.” *Statistics, Optimization & Information Computing* 13.5: 1916-1932, 2025.
62. Tawfik, Mohammed. “Optimized intrusion detection in IoT and fog computing using ensemble learning and advanced feature selection.” *PLoS ONE* 19.8: e0304082, 2024.
63. G. Hassan, K. M. Hosny, M. M. Fouda, and I. S. Fathi, “Efficient compression of fetal phonocardiography bio-medical signals for Internet of Healthcare Things,” *IEEE Access*, vol. 11, pp. 122991–123003, Nov. 2023.
64. I. S. Fathi, M. A. Ahmed, and M. A. Makhlouf, “An efficient compression technique for foetal phonocardiogram signals in remote healthcare monitoring systems,” *Multimedia Tools Appl.*, vol. 82, pp. 19993–20014, 2023.
65. E. A. Aldakheel, D. S. Khafaga, I. S. Fathi, K. M. Hosny, and G. Hassan, “Efficient analysis of large-size bio-signals based on orthogonal generalized Laguerre moments of fractional orders and Schwarz–Rutishauser algorithm,” *Fractal Fract.*, vol. 7, no. 11, p. 826, Nov. 2023.