Shahid Bahonar University of Kerman

Journal of Mahani Mathematical Research

Print ISSN: 2251-7952 Online ISSN: 2645-4505

JMMR 2012

A NEW HYBRID FILTER-WRAPPER FEATURE SELECTION USING EQUILIBRIUM OPTIMIZER AND SIMULATED ANNEALING

M. Ansari Shiri, M.R. Omidi, and N. Mansouri ✉

Article type: Research Article

(Received: 02 March 2023, Received in revised form 24 July 2023)

(Accepted: 14 September 2023, Published Online: 15 September 2023)

Abstract. Data dimensions and networks have grown exponentially with the Internet and communications. The challenge of high-dimensional data is increasing for machine learning and data science. This paper presents a hybrid filter-wrapper feature selection method based on Equilibrium Optimization (EO) and Simulated Annealing (SA). The proposed algorithm is named Filter-Wrapper Binary Equilibrium Optimizer Simulated Annealing (FWBEOSA). We used SA to solve the local optimal problem so that EO could be more accurate and better able to select the best subset of features. FWBEOSA utilizes a filtering phase that increases accuracy as well as reduces the number of selected features. The proposed method is evaluated on 17 standard UCI datasets using Support Vector Machine (SVM) and K-Nearest Neighbors (KNN) classifiers and compared with ten state-of-the-art algorithms (i.e., Binary Equilibrium Optimizer (BEO), Binary Gray Wolf Optimization (BGWO), Binary Swarm Slap Algorithm (BSSA), Binary Genetic Algorithm (BGA), Binary Particle Swarm Optimization (BPSO), Binary Social Mimic Optimization (BSMO), Binary Atom Search Optimization (BASO), Modified Flower Pollination Algorithm (MFPA), Bar Bones Particle Swarm Optimization (BBPSO) and Two-phase Mutation Gray Wolf Optimization (TMGWO)). Based on the results of the SVM classification, the highest level of accuracy was achieved in 13 out of 17 data sets (76%), and the lowest number of selected features was achieved in 15 out of 17 data sets (88%). Furthermore, the proposed algorithm using class KNN achieved the highest accuracy rate in 14 datasets (82%) and the lowest selective feature rate in 13 datasets (76%).

Keywords: Feature selection, Equilibrium Optimizer, Simulated Annealing, Filter, Wrapper.
2020 MSC: 68T20

1. **Introduction**

Scientists and technologists have created more and more datasets due to technological advancements, such as the internet. The high-dimensional dataset creates a number of challenges, including slow model building, redundant data,

---

✉ Najme.mansouri@gmail.com, ORCID: 0000-0002-1928-5566
DOI: 10.22103/jmmr.2023.21150.1411
Publisher: Shahid Bahonar University of Kerman
How to cite: M. Anari Shiri, M.R. Omidi, N. Mansouri, *A New Hybrid Filter-Wrapper Feature Selection using Equilibrium Optimizer and Simulated Annealing*, J. Mahani Math. Res. 2024; 13(1): 293-332.

and reduced learning model performance [17]. The high predictive capability of supervised learning algorithms makes them popular in machine learning. Complex real-world classification problems often require relevant features. The learning model is improved by using a variety of candidate features to describe the resulting domain. It has been observed that many of these are irrelevant and redundant. There is a poor performance in coping with irrelevant and redundant features in learning models [11]. The application of classification methods is challenged by irrelevant and redundant features, which will dimensionality reduce the accuracy of the classifier and increase the spatial and temporal complexity. Consequently, data reduction has become a crucial part of the data preprocessing process. By removing redundant and irrelevant features from the original dataset and not changing its physical properties, feature selection reduces data dimensionality effectively. In order to increase the performance and interpretability of a classifier, it can extract the most valuable and representative features from the original dataset [8, 31]. Figure 1 shows the main process of feature selection, feature selection by meta-heuristics and feature selection methods based on evaluation criteria.

In FS, four main processes are carried out: searching for feature subsets, evaluating them, validating them, and setting stopping criteria. According to their evaluation criteria, FS methods can be classified as filter, wrapper, and hybrid. Statistical functions are used to select and rank feature subsets using filter-based methods. Wrapper-based methods communicate directly with the classifier. Several feature selection algorithms use wrapper-based algorithms. Compared to the filter-based method, it is computationally more expensive [14]. By contrast, hybrid models combine the advantages of filters and wrappers [39]. The most common combinatorial method involves ranking features and reducing the number of candidates via filters and wrappers. In a sequential approach, the first step involves filtering features to reduce the number to be considered in the next step. Following this, the wrapper method is used in the second step to select the desired number of features. Meta-heuristic feature selection starts with generating an initial population and calculating its fitness value. Exploration and exploitation must be conducted until the maximum repetition level is reached in order to detect the global and local optimum. Furthermore, the fitness value is calculated again, and the population is updated, and finally, the best solution has been found after reaching the stopping condition. It can be challenging to identify the optimal subset of features in general. Researchers in data mining and machine learning have increasingly focused on FS [1, 12]. One of the main challenges in feature selection is removing preprocessed and prepared data without compromising its quality. There are many different approaches and solutions to this problem [26]. To find the best subset for feature selection problems, several methods have been used, including exhaustive search, greedy search, and random search. It is common for most of these methods to converge prematurely, have a high computational complexity, and require a significant amount of computing power [5]. The

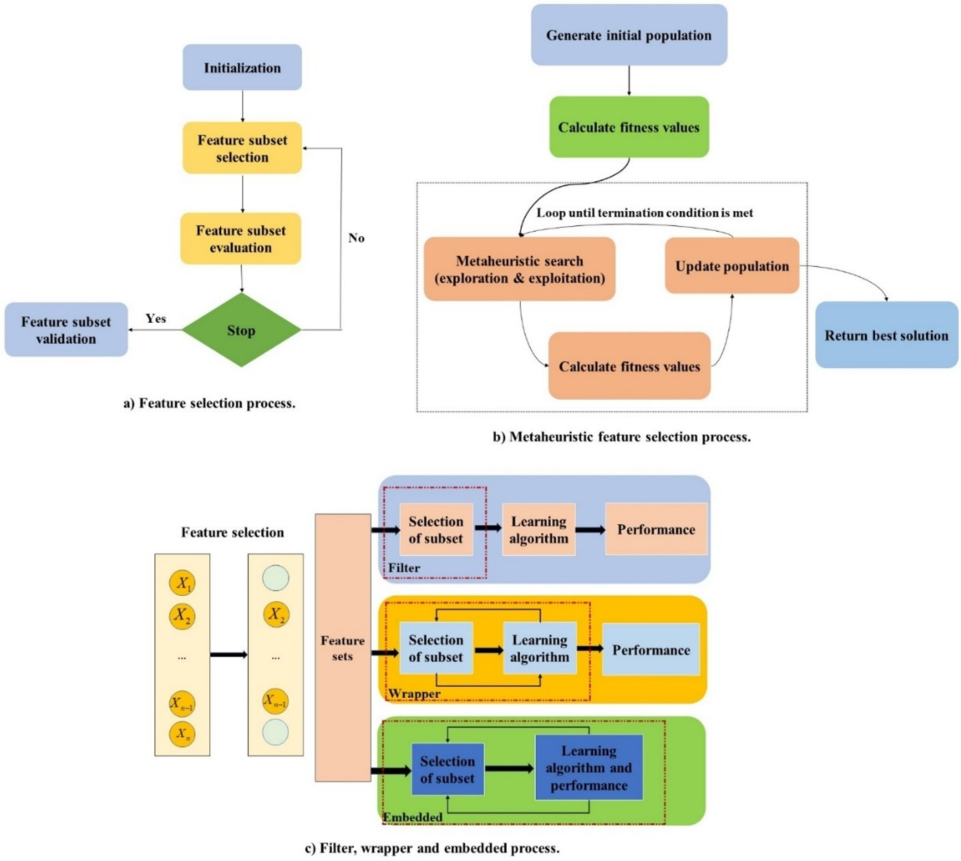

FIGURE 1. Feature selection process.

use of meta-heuristics has proven useful for the solution of a variety of optimization problems [30], including machine learning, generation problems, and feature selection. Meta-heuristic algorithms have become increasingly popular in recent years for solving complex optimization problems. Compared to traditional methods, they are more efficient and effective. Traditional optimization methods, such as gradient search, cannot easily solve structural optimization, economic optimization, or engineering design problems. The strength, flexibility, and simplicity of meta-heuristic algorithms has led to their application in a variety of fields in science and engineering. Throughout the optimization process, exploration and exploitation are both essential conditions. In exploration, the search space is explored further to find the global optimum, and in exploitation, the obtained solutions are refined to find the local optimum.

Therefore, both stages must be approached with a balanced approach. In order to avoid getting stuck in local optima or converge too early, meta-heuristic algorithms need to have a balance between exploration and exploitation. The controlling of this condition has been done by a variety of methods in different studies [6, 29]. We developed a filter-wrapper-based feature selection method using EO and SA to address the problems presented in this paper. The EO method is capable of balancing exploration and exploitation, as is SA, which can quickly find local optima. Consequently, it combines the efficiency of filters and the precision of wrappers.
Following are the main contributions of the paper:

- The filter phase considers two filter metrics: relevance (relevance of features and class label features) and redundancy (correlation between each feature and other features).
- BEO is used to solve the FS problem due to its fast convergence to the optimal solution, as well as its strong exploration capabilities.
- For exploitation enhancement, BEOs have been enhanced with SA mechanisms, called FWBEOSA. With these mechanisms, the optimal feature subsets are found and local optima are avoided.
- The proposed algorithms are evaluated over 17 well-regarded datasets obtained from the UCI repository to evaluate their performance. FW-BEOSA outperforms BEO and other competitive algorithms in most cases.

This paper is organized as follows. Section 2 explains the background of meta-heuristic algorithms. Section 3 provides an overview of works related to feature selection using evolutionary algorithms. A brief overview of the proposed method is provided in Section 4. Section 5 describes the experimental setup and results. Section 6 discusses conclusions and future work.

Table 1 shows abbreviations and their definitions.

2. Background

2.1. **Simulated Annealing (SA).** Metals and materials science anneal solids by heating them to high temperatures then gradually lowering them. SA is an improved version of hill climbing based on the use of a single solution. The SA accepts a certain probability of making a bad move in order to avoid getting stuck in locally optimal solutions. Using the fitness function, neighboring solutions are generated for a given solution (agent). A neighbor's fitness value is compared with the current solution, and if the neighbor's value is better, the current solution is replaced. By using the Boltzman equation ($p = e^{\frac{-\lambda}{T_k}}$), the neighbor's fitness value is calculated in case it is worse than the current solution's fitness value [16, 24]. Thus, acceptance probability can be expressed

Table 1. Symbols and descriptions.

Symbol	Description
SVM	Support Vector Machine
KNN	K-Nearest Neighbor
DT	Decision Tree
RF	Random Forest
NB	Naïve Bayes
ANN	Artificial Neural Network
FRB	Fuzzy Rule Based
EO	Equilibrium Optimizer
SA	Simulated Annealing
BEO	Binary Equilibrium Optimizer
BGWO	Binary Gray Wolf Optimization
BSSA	Binary Swarm Slap Algorithm
BGA	Binary Genetic Algorithm
BPSO	Binary Particle Swarm Optimization
BSMO	Binary Social Mimic Optimization
BASO	Binary Atom Search Optimization
MFPA	Modified Flower Pollination Algorithm
BBPSO	Bar Bones Particle Swarm Optimization
TMGWO	Two-phase Mutation Gray Wolf Optimization
GR	Generation Rate
GA	Genetic Algorithms
CSO	Competitive Swarm Optimization
SCA	Sine-Cosine Algorithm
HHO	Harris-Hawks Optimization
SCHHO	Sine-Cosine Harris-Hawks Optimization
HHOBSA	Harris-Hawks Optimization Binary Simulated Annealing
SCAGA	Sine-Cosine Algorithm Genetic Algorithm
DA	Dragonfly Algorithm
FS	Feature Selection
EPD	Evolutionary Population Dynamics
DBOA	Dynamic Butterfly Optimization Algorithm
FIM	Feature Interaction Maximization
FA	Firefly Algorithm
SCC	Spearman's Correlation Coefficients
GP	Generation Probability
GOQRFA	Genetic Operators Quasi-Reflected Firefly Algorithm
QRBL	Quasi-Reflection-Based Learning mechanism
GO	Genetic Operators

as follows:

(1)
$$p = \begin{cases} e^{\frac{-\lambda}{I_k}} & \text{if } \lambda > 0 \\ 1 & \text{if } \lambda \leq 0 \end{cases}$$

A fitness value $\lambda = \text{ fit } (neighbor) - \text{ fit } (\text{ current.sulotion }^{n})$ can be calculated by comparing the generated neighbor with the one of the current solutions. As new solutions are accepted, I_k indicates the temperature at the $k_t h$ instance. A problem of dimension $|D|$ is considered at first, where $I_k = 2 * |D|$ represents the original dimension. As a result of more iterations, $I_k + 1 = I_k$ is used with a cooling coefficient of [16, 24].

2.2. **Equilibrium Optimization (EO).** This section describes briefly the original EO algorithm. *Inspiration* : Mass control balances were used to estimate equilibrium and dynamics in the original EO. Sources and sinks of non-reactive constituents are accounted for by their concentrations in the control volume. An energy-generating control volume produces energy based on the mass balance equation, which conserves mass entering, supports physics, and provides support for underlying physics. In order to calculate mass in time, you subtract mass coming in from mass leaving the system, using the generic mass balance equation [15, 35]. $Mathematicalmodel$: As solutions, particles are treated by the EO as solutions, and their concentrations provide information about their location. Optimizing concentration involves identifying equilibrium candidates. An equilibrium state (optimal solution) is reached when particles (search agents) repeat the process. The update process involves three terms that each represent a rule. Alternatively, equilibrium concentration is a solution chosen by chance from a pool of equilibrium solutions. Secondly, we have the concentration difference between the particle and the equilibrium state. By using this term, particles are able to explore more (global search) efficiently. Small steps are exploited or refined using the last term. This term refers to the rate at which a product is generated. Each term is represented mathematically according to how it influences the search mechanism [15, 35].

$$X_{k+1} = X_{eq} + (X_k - X_{eq}) \cdot E + \frac{G}{\lambda V}(1 - E) \tag{2}$$

X_eq is the equilibrium concentration at an equilibrium point, λ is the residence time, V is the control volume, and k is the current iteration. Eq. 3 presents a mathematical definition of L [15, 35].

$$\lambda = \frac{Q}{V} \tag{3}$$

2.2.1. *Exponential Term (E).* Flow rates can be calculated by subtracting Q out and into a control volume. In addition, an exponential term E is used to update the main concentration rule. Exploration and exploitation phases are balanced using this term. According to [15, 35], the definition is as follows:

$$E = e^{-\lambda(t - t_0)} \tag{4}$$

Where:

$$t_0 = \frac{1}{\lambda} \ln\left(-a_1 \operatorname{sign}(r - 0.5)\left[e^{-\lambda t} - 1\right]\right) + t \tag{5}$$

Exploration and exploitation capabilities are controlled by constants a_1 and a_2. Higher a_1 values indicate a stronger exploration capability compared to a worse exploitation capability. Higher values of a_2 indicate stronger exploitation capabilities and lower values indicate poorer exploration capabilities. According to the literature, a_1 and a_2 are both equal to 2. In this case, r is a random variable. Exploration and exploitation are indicated by $sign(r - 0.5)$ [15, 35].

$$t = \left(1 - \frac{k}{Max_k}\right)^{a_2 \frac{k}{Max_k}} \tag{6}$$

A current iteration is k, while a maximum iteration is Max_k. Higher a_2 indicates better exploitation abilities and lower exploration [15, 35].

2.2.2. *Generation Rate (G).* A key part of the EO algorithm is the mass generation rate. The definition is as follows [15, 35]:

$$G = G_0 E \tag{7}$$

Where:

$$G_0 = GCP\,(X_{eq} - \lambda X) \tag{8}$$

$$GCP = \begin{cases} 0.5 r_1 \times 0 \text{ enes } (1, \dim) & r_2 \geq GP \\ 0 & r_2 < GP \end{cases} \tag{9}$$

The control parameter GCP determines the generation rate, and GP determines the generation probability. $Ones(1, dim)$ consists of a vector of length $1dim$, initialized with one. Dimensions of a vector are determined by dim. A random parameter r_1, r_2, and r_3 are generated between 0 and 1.

2.2.3. *Exploration ability of EO.* EO leads to exploration through several parameters and mechanisms, as follows:

- a_1 : This parameter controls how much exploration (magnitude) the algorithm performs. This measure determines how far away the new position is from equilibrium. Exploration ability increases with a1 value. The performance of exploration would be considerably degraded by numbers over three. Exploration ability can be expanded through the use of large a1 values, which magnify concentration variations. The agents are forced to search on boundaries when the value exceeds three, according to empirical testing.
- $sign(r0.5)$: Basically, it determines which direction exploration will take. A uniform distribution of r with a [0, 1] mean indicates equal probabilities of negative and positive results.
- GP : By controlling the generation rate, it controls the participation probability of concentration updating. In the optimization process, $GP = 1$ means there is no generation rate term. High exploration capability leads to inaccuracy in solutions. When $GP = 0$, a local optimum has a greater chance of stagnating because generation rate is always present. An empirical test indicates that $GP = 0.5$ provides a good balance between exploration and exploitation [15, 35].

2.2.4. *Exploitation ability of EO.* EO exploitation and local search are based on the following parameters and mechanisms:

- a_2 : This parameter controls exploitation. This algorithm digs around until it finds the best solution in order to determine the quantity of exploitation (magnitude).
- $sign(r0.5)$: It also controls the quality of exploitation (direction). It specifies the direction in which a local search should be conducted.

- *Memory saving:* This method saves a number of good particles and substitutes them for worse particles. The EO's exploitability is directly improved by this feature.
- *Equilibrium pool:* By the end of an iteration, exploration fades away and exploitation takes over. Therefore, in the last iterations, when equilibrium candidates are close to each other, the concentration updating process facilitates local search around them, leading to exploitation [15, 35].

3. Related Works

According to Ding et al., it is possible to select features with both Genetic Algorithms (GA) and Competitive Swarm Optimization (CSO) [13]. By using crossover and mutation operators, this paper aims to speed up generation time and prevent premature population growth. Using Sine-Cosine Algorithm (SCA) and Harris-Hawks Optimization (HHO), Hussain et al. [18] proposed a hybrid optimization method that integrates SCA with HHO. Through SCA integration, inefficiencies in exploration were solved, as well as a dynamic adjustment of candidate solutions was achieved to prevent solutions from stagnating. A numerical optimization test suite for CEC'17 was used to evaluate the proposed method, SCHHO, with 16 datasets with dimensions exceeding 15000 attributes. According to Abdelbasset et al. [3], simulated annealing is based on Harris Hawks optimization. As part of its approach to solving the FS problem, the HHOBSA algorithm simulated annealing to solve Harris Hawks Optimization by means of bitwise operations. A technique known as SA is used to improve the performance of HHOBSA. An SCAGA algorithm is proposed by Abualigah and Dulaimi [2], which integrates Sine Cosine Algorithms and Genetic Algorithms. In optimization methods, exploration of the search space and exploitation of the search space are the two main search strategies. The proposed SCAGA achieved better results when balancing exploration and exploitation of the search space. A standard deviation test has also been conducted on the proposed SCAGA, along with testing classification accuracy, worst fitness, mean fitness, and best fitness. A hybrid GWO/HHO approach was created by the authors based on Al-Wajih et al. [7], achieving a good balance between these two approaches. To meet the feature selection nature requirement, the continuous search space is transformed into a binary one using the sigmoid transfer function. To determine the quality of selected features, a KNN wrapper is used. The proposed method was tested on 18 standard benchmark datasets from UCI. An improved version of Dragonfly Algorithm (DA) is proposed by Chantar et al. [12] by combining it with SA. DA's local optima problem can be solved and its ability to select the best subset of features can be improved. For the purpose of evaluating the proposed FS approach, a set of frequently used data sets from the UCI repository was used. A comparison between the proposed hybrid approach and wrapper-based FS methods using a basic version of Binary Dragonfly Algorithm shows superior performance. Thaher et al. [38] proposed an efficient feature selection approach based on a Boolean variant of Particle Swarm Optimization (BPSO) boosted with Evolutionary Population Dynamics (EPD). It enhances the BPSO's exploration capability so that local optima obstacles can be avoided. The worst half of the solutions are discarded in the BPSO-EPD by repositioning them around the best solutions. Based on experimental results, BPSO-TEPD outperformed conventional BPSO and other EPD variants, especially when using EPD-based feature selection approaches. Using a Dynamic Butterfly Optimization Algorithm based on Interaction Maximization (IFS-DBOIM), Tiwari and Chaturvedi [33] developed a new hybrid feature selection method. This algorithm combines Dynamic Butterfly Optimization Algorithm (DBOA) and Mutual Information-Based Feature Interaction Maximization (FIM) to select the most optimal set of features. Furthermore, it avoids redundancies and irrelevant features, and tries to improve tradeoffs between exploration and exploitation phases. While DBOA provides better exploration, exploitation, and avoidance of local optima entrapment than FIM, the latter scores maximum relevance with the least redundant new features. Using an algorithm from the swarm intelligence branch of machine learning, Zivkovic et al. [42] improved feature selection by employing a novel algorithm. Combining machine learning and metaheuristics has created a new branch of artificial intelligence known as learn heuristics. The approach utilizes both the capability of feature selection to select the best solutions for accuracy and performance and the characteristic of swarm intelligence algorithms to efficiently comb through large search spaces. Feature selection is performed using this method as a wrapper, and the improvements are significant. The authors proposed a modified version of the Salp Swarm Algorithm for selecting features. A classification model based on K-Nearest Neighbors is used to verify this solution with 21 datasets. Bacanin et al. [9] proposed the Genetic Operators Quasi-Reflected FA (GOQRFA) for feature selection problem. By combining solutions from novel regions of the search space in early iterations, GO reduces the likelihood of being trapped in suboptimal domains and increases efficiency. Conversely, GOs enable the problem to be fine-tuned around optimum domains, resulting in higher-quality solutions. There are two implications of QRBL for algorithm robustness. If QRBL mechanism is applied, solutions diversity and convergence speed can be dramatically boosted in early iterations as well as the later stages of the execution process. A dynamically adjustable step size parameter is also included in GOQRFA. An evaluation of the proposed algorithm demonstrated its robustness and efficiency in practical simulations. Most of these methods have a high complexity because they are one-dimensional and do not use filters, in addition to the possibility of getting caught in the local optimum. This research combined filter methods with equilibrium optimization and simulated annealing to solve these problems. Table 2 summarizes the related works on feature selection.

TABLE 2. Related works on feature selection.

Ref.	Algorithm(s)	Compared methods	Objective function(s)	Learning algorithm	Dataset used	Disadvantage(s)
(Y. Ding et al., 2020) [13]	HBCSO	CSO, GA	Accuracy, Number of selected features	KNN	5	Number of datasets is low, High running time, in contrast to more accurate classification methods, it can only be used with simple classifiers
(K. Hussain et al. 2021) [18]	CSHHO	CSA, HHO	Accuracy, Number of selected features, Convergence speed, solution fitness	KNN	16	Not adaptively updating the parameter in SCHHO, not suitable for large-scale optimization.
(M. Abdel-Basset et al., 2021)[3]	HHOBSA	HHO, SA	Accuracy, Number of selected features	KNN	24	High complexity
(L. Abualigah and A. J. Dulaimi 2021) [2]	SCAGA	CSA, GA	Classification accuracy, Worst fitness, Mean fitness, Best fitness, the Average number of features, Standard deviation	KNN	16	High computational complexity, ambiguous results
(R. Al-Wajih et al. 2021) [7]	HBGWOHHO	GWO, HHO	Accuracy, Number of selected features, and computational time	KNN	18	Irrational numbers in the best fitness metric
(H. Chantar et al. 2021) [12]	BDA-SA	DA, SA	Accuracy, Best fitness value, Number of selected features	KNN	18	The selection of a relatively large number of features
(Thaher et al. 2022) [38]	BPSO-EPD	PSO, EPD	Accuracy, Number of selected features	KNN, DT	22	High computational complexity, local optima are poorly exploited and may be trapped
(Tiwari and Chaturvedi 2022) [33]	IFS-DBOIM	DBOA	Accuracy, Number of selected features	SVM, NB, DT	20	is a nondeterministic algorithm, it suffers from a lack of generalization and relies on the characteristics of applied datasets
(M. Zivkovic et al. 2022) [42]	SSARM-SCA	SSA, CSA	Accuracy, Number of selected features, benchmark function	KNN	21	There is no guarantee that it would work well for other optimization tasks
(Bacanin et al. 2023) [9]	GOQRFA	FA	convergent, quality of solutions, and classification accuracy	KNN	22	High complexity, GOQRFA's potential has not been fully explored on other real-world datasets

4. Methodology

The proposed method is presented in two phases: filter and wrapper.

4.1. **Filter phase.** Using variable ranking techniques as the primary criterion for variable selection, filter methods select variables based on their order of importance. Due to their simplicity, ranking methods have been successful in practical applications. Scored variables are ranked according to an appropriate ranking criterion, and variables below the threshold are removed. As a filter method, ranking removes variables that are less relevant before classifying them. The proposed method uses two filter methods [11].

4.1.1. *Redundancy computation.* Redundancy describes the degree to which two or more features are interdependent. In simple terms, the MI provides a way of measuring the dependency of a feature on a subset of features. The feature sets are symmetric, nonlinear, nonnegative, and do not diminish as features are added. Nevertheless, it is difficult to determine precisely which features of S are redundant based on this measure. It would be wise to develop more elaborate redundancy criteria, such as Markov blankets and total correlations, in order to reduce redundancy. An analysis of data-driven correlations can be used to investigate the relationship between numerical features and materials knowledge. By calculating correlation coefficients between two features using data-driven methodology, it is possible to calculate them quickly. A highly correlated feature must have a correlation coefficient that exceeds a certain threshold in order to qualify for Spearman correlation coefficient calculation. Between two features, we used Spearman's Correlation Coefficients (SCC) to estimate the correlation coefficients. Based on Eq. 10, correlation analysis uses a different trigger condition. With the SCC method, nonlinear correlations can also be measured along with linear correlations. SCC measures how closely two features are related. Stronger the correlation, the higher the SCC value. In the presence of high correlation, SCC between f_i and f_j should exceed k_1 [25, 40, 41].

$$\operatorname{corr}(f_i, f_j) = |\mathrm{SCC}(f_i, f_j)|, n \geq k_1 \tag{10}$$

4.1.2. *Relevance computation.* As a general rule of thumb, a feature is relevant if it provides information about the Class (C) label feature alone or when it is used in conjunction with other variables. Various methods have been used to define relevance, including weakly relevant, strongly relevant, and irrelevant features. When a feature is considered to be strongly relevant to C, it cannot be replaced by any other feature without removing the information they provide. Alternatively, weakly relevant features provide information about C, but it is possible to substitute them with other relevant features without losing any significance. Taking irrelevant features out of C can result in a loss of information about the C [25, 40, 41]. Table 3 shows the levels of relevance for feature f_i.

TABLE 3. Levels of relevance for feature f_i.

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

4.2. **Wrapper phase.**

4.2.1. *Binary Equilibrium Optimizer and its Hybridization with Simulated Annealing.* Now that the FS problem has been solved, finding the right features will be the main challenge. It can be quite time-consuming to figure out what feature subset is the best in wrapper-based problems, since each feature subset has to be evaluated by means of a classifier algorithm. Thus, it is necessary to minimize the number of subset evaluations by using an optimization technique. In order to suggest this algorithm as a search method for a wrapper-based FS method, the EO results were compared to those of other methods in the literature. EO was originally designed for continuous search spaces, but since FS uses binary search spaces, a Binary version of EO (BEO) has been created. EO was exploited better through simulated annealing and computation complexity was reduced. There are only binary 0, 1 solutions in binary search space. Binary versions consist of a vector of 0's and 1's, where 1 indicates that the corresponding feature is selected, and 0 indicates that it is not selected. Continuously updating the concentration of the solution is Eq. 2, which is used in EO. EO's binary version requires a transfer function. As depicted in Fig. 2, sigmoid transfer functions are given by Eq. 11.

$$T(x) = \frac{1}{1+e^{-x}} \tag{11}$$

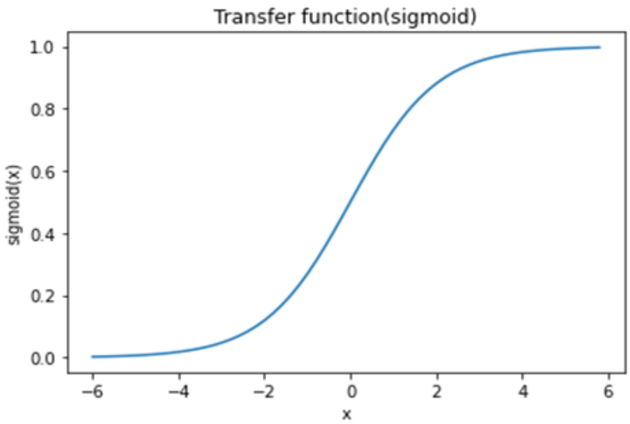

Figure 2. Transfer function for converting continuous search space to binary.

EO primarily exploits the Equilibrium pool [15]. The exploration and exploitation of this pool are controlled by this pool. Initial iterations are characterized by high distances among equilibrium candidates, which helps perform global search by updating concentrations. Iteratively increasing the number of iterations leads to the equilibrium candidates approaching each other, allowing the concentrations to be updated through the use of the equilibrium candidates, thus facilitating local searches surrounding the equilibrium candidates.

Generation Probability (GP, Eq. (9)) is used exclusively for exploration. Nevertheless, there is nothing specific about exploitation to consider. The concept of SA is used for performing local searches, i.e., exploitation. In this work, we aim to improve the performance of BEO for feature solution problems. In order to accomplish that, the best solution obtained through BEO is passed on to the SA algorithm instead of being generated at random. Thus, SA will search locally for a better solution based on the optimal solution found by BEO so far. At the beginning of the search process, diversification tends to be more important than intensification for exploring potential useful areas of the feature space. Toward the end of the exploration phase, exploitation becomes more important, since it requires the search for better solutions around those found by exploration [38]. The search space can be explored and exploited through hybrid approaches, such as BEO-SA in our case. However, hybrid approaches incur a higher computational cost than classic wrapper approaches, which use a heuristic and an evaluator. BEO and SA are combined by integrating the SA process into the BEO algorithm. The process is as follows:

a. Temperature (T): A temperature parameter (T) is introduced in BEO-SA to control the probability of rejecting a worse solution. Temperatures are initially set high, increasing the likelihood of accepting poorer solutions. By exploring solution space in this manner early on, the algorithm avoids local optima in the latter stages.

b. Cooling Schedule: Depending on the cooling schedule, the temperature will decrease over time. Gradually lowering the temperature with the BEO-SA algorithm greatly reduces the probability of accepting worse solutions. Temperature decreases quickly according to the cooling rate. Slower cooling rates enable the algorithm to explore more of the solution space.

c. SA in BEO: BEO generates new candidate solutions (particles) by randomly perturbing the temperature T. The perturbation determines the type of features that are selected for the new particle. The higher temperature in the early iterations facilitates exploration by enhancing perturbation. The perturbation decreases with each iteration, which leads to a better solution as the iterations progress.

d. Combining Diverse Solutions: By accepting the worst solutions early on, the SA process helps the BEO-SA algorithm combine diverse solutions. As a result, better feature subsets are discovered through a more comprehensive exploration of the feature space.

BEO-SA combines Equilibrium Optimization with Simulated Annealing to enhance its search capability, enabling it to explore the solution space more effectively, escape local optimum, and perhaps discover better solutions. Exploration and exploitation are balanced by temperature control and cooling schedules in the optimization process. Multi-objective optimization occurs when two criteria are considered when evaluating FS (classification accuracy and number of selected features) [27]. In more specific terms, the goal of FS is to achieve maximum classification accuracy with the fewest features possible. Classification error rate has been used instead of accuracy as a criterion to avoid conflicts. The FS problem is transformed into a single objective problem using Eq. 12.

$$\text{Fïness } = \omega\gamma(S) + (1-\omega)\frac{|S|}{D} \tag{12}$$

Considering the subset of features selected as S, $|S|$ represents the number of features selected, $\gamma(S)$ is classification error rate of S, the dataset's original dimension is D, weights (ω) are represented by the values 0 and 1. In order to compute the classification error ($\gamma(S)$), we used SVM (Support Vector Machine) [37] and KNN (K-Nearest Neighbors) classifiers.

4.3. **Proposed Filter-Wrapper Binary Equilibrium Optimizer Simulated annealing (FWBEOSA).** Exploration and exploitation phases must be balanced to ensure success. It is possible for exploration and exploitation to become prematurely convergent or to become trapped in local optima if they are not balanced. By combining BEO and SA, we have developed a filter-wrapper-based feature selection method to address premature convergence, local optima trapping, and computational complexity. The goal of feature selection is to select a subset of features that have a maximum relationship between them and the class label feature so that they are minimally redundant. In the proposed method, the first step uses a hybrid filter method to reduce the number of features to be considered in the next step. A Spearman correlation coefficient between the features is calculated as the first step of the hybrid filter in accordance with section 4.1.1. If two features have a higher correlation score than a predetermined threshold, one is eliminated from consideration. A relevance value is calculated between the remaining features and the class label feature in the next step of the filter method, according to section 4.1.2. When the relevance between a feature and the class label feature is less than a certain threshold, we can discard the attribute without losing valuable information, because it adds no information to our assets. In the second stage, a hybrid wrapper method (BEO with SA) is used to identify the desired number of features. Wrapper steps are as follows:

Initialize the population: A binary feature mask is created for each individual (particle) in the population. Feature inclusions or exclusions are selected randomly (1) or (0).

Evaluate fitness: The fitness function evaluates classification performance on a subset of features by using SVM and KNN as classifiers. Using the weight parameter omega, fitness functions control classification accuracy and number of selected features.

Binary Equilibrium Optimization with Simulated Annealing (BEO-SA): It runs the Binary Equilibrium Optimization (BEO) and Simulated Annealing (SA) processes simultaneously. The BEO process updates feature masks based on

probabilistic rules. The SA process improves the search capability of the algorithm by escaping local optima. To balance exploration and exploitation, parameters such as cooling rate (t), temperature (T), and others are used.
Feature selection and classification: In BEO-SA, fitness values are used to select the most appropriate feature mask. Based on these features, SVM and KNN are used to classify the data, and classification accuracy is measured.
Performance evaluation: During the iteration process, BEO-SA tracks which features are selected, as well as the accuracy of the classification. Performance is summarized across all runs.
This method combines the efficiency and speed of the filter method with the precision of the wrappers. The proposed method is shown in Fig. 3.

5. Experimental Setup and Results

The inherent characteristics of SVMs and KNN classifiers make them suitable for feature selection. Features are selected using these classifiers for the following reasons:
Intrinsic Feature Importance: Both SVM and KNN can rank features implicitly based on their contribution to classification accuracy. Model coefficients (weights) can be used to determine the importance of features in SVM. Feature contributions that significantly influence the neighbors' selection are considered important in KNN.
Robustness to Irrelevant Features: There is generally less effect on SVM and KNN from irrelevant or noisy features. KNN uses the distance between data points to determine which hyperplane best separates the classes. SVM searches for the hyperplane that best separates the classes. Classification algorithms may be less sensitive to irrelevant features, resulting in more robust feature selection.
Non-Parametric Nature: In KNN, no strong assumptions are made regarding the underlying distribution of the data. Unlike traditional linear models, it is more suitable for datasets with complex relationships between features and target variables.
Flexibility in Distance Metrics: KNN allows for the use of different distance metrics, such as Euclidean distance, Manhattan distance, or cosine similarity. Availability of flexible distance functions enables KNN to handle different types of data effectively. Feature Interactions: There is no difference between SVM and KNN when it comes to capturing interactions between features implicitly. SVM's kernel trick can be used to find nonlinear decision boundaries, while KNN's distance-based approach can adapt to feature interactions.
No Feature Assumption: KNN and SVM assume no particular relationship between features and the target variable, unlike linear regression, which does. The lack of assumptions allows them to find patterns in data that specific functional models may have missed. Suitability for High-Dimensional Data: It is widely known that SVMs and KNNs perform relatively well on high-dimensional data,

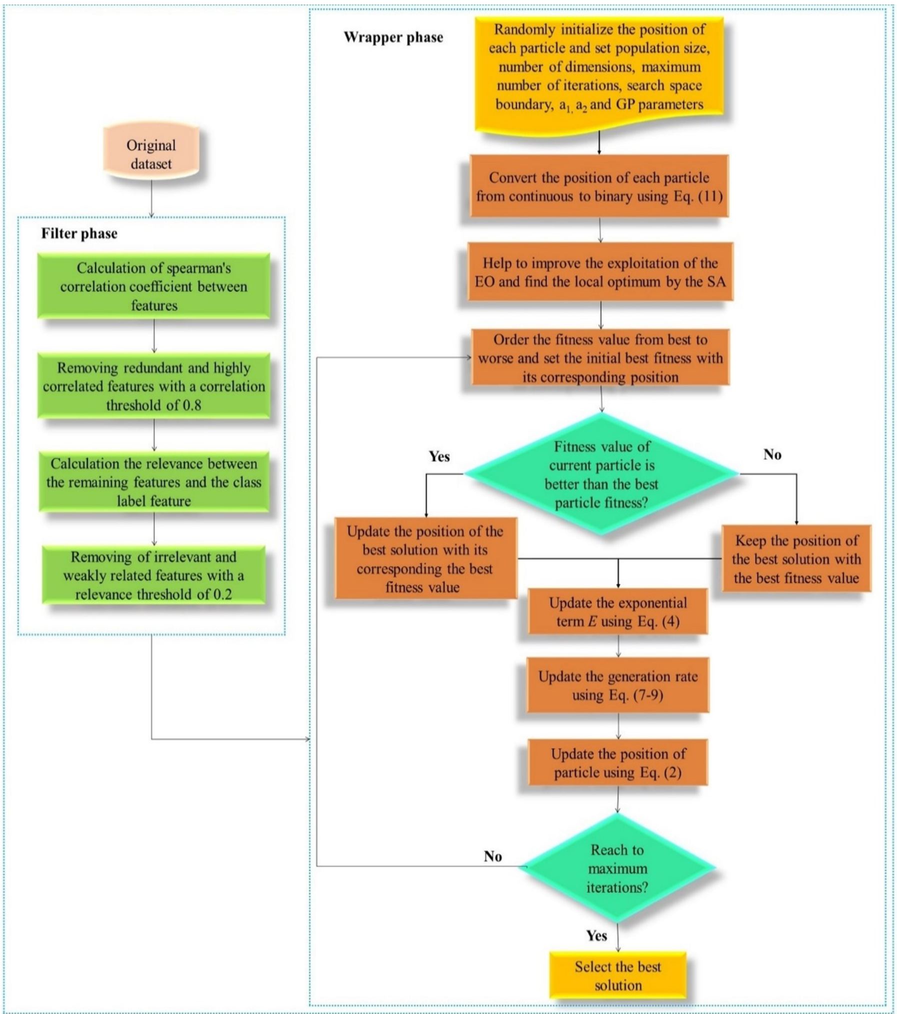

FIGURE 3. Flowchart of FWBEOSA.

even if the number of features exceeds the number of samples. The high dimensionality of feature spaces makes them suitable for feature selection tasks. *Interpretability:* It is possible to interpret features selected using SVM and KNN. KNN analyzes the distance-weighted contributions of features to increase understanding of their importance based on the non-zero coefficients in the model.

However, SVM and KNN's suitability for feature selection can vary according to the dataset and problem. To find the most appropriate approach for a given problem, it is always recommended to experiment with different classifiers and feature selection techniques. Due to the data set used and the most popular classifiers (SVM and KNN), we used them for feature selection and classification [5, 28, 34]. According to Fig. 4, metaheuristic algorithms for feature selection use a variety of classifiers.

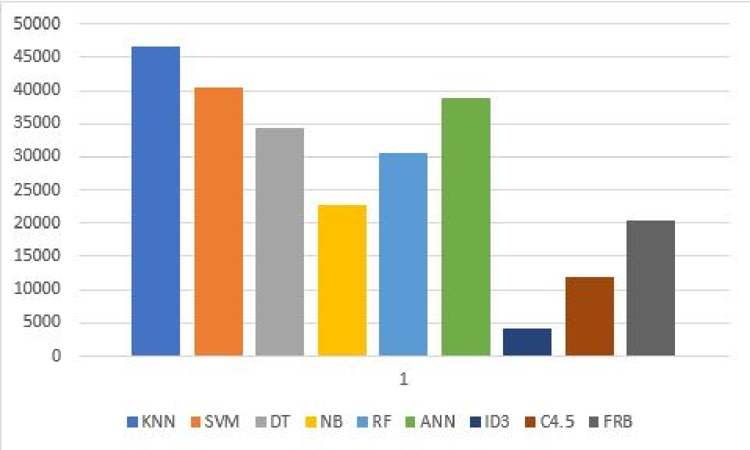

FIGURE 4. Classifier rate in metaheuristic algorithms for feature selection [28].

In the training dataset, 80% of the instances are used, while 20% are used in the testing dataset. Based on train data, FS methods are used to select a subset of features. Graphs are plotted and implemented using Matplotlib and Python 3. The performance assessment of FWBEOSA took into account 17 UCI datasets.

5.1. **Dataset description and control parameters setup.** Table 4 shows 17 evaluation datasets. The number of features in each dataset can be determined by comparing it before and after filtering. Duplicate features are removed by filtering based on whether they are redundant. In addition, features that are generally unrelated can also be ignored if correlations between features and category features are considered. Furthermore, the proposed method is computationally simpler than other methods, making it more accurate for selecting a subset of features. There are a variety of attributes (features) and instances in these datasets. This variance makes the proposed method robust. Multiagent evolutionary algorithms require a population size and an iteration number that are both very important parameters. As agents learn from each other's experiences, population size determines how they learn, while iterations determine how agents evolve step by step. In BEO and FWBEOSA, the population size is varied as [5, 10, 20, 30, 50]. Experiments have determined that a maximum of 50 iterations should be used. Table 5 shows some parameters of the compared methods.

Table 4. Description of datasets before and after filter phase.

Dataset	No of samples	No of features before applying filter	No of features after applying filter	No of classes	Domain
Algerian forest fires	244	14	8	2	Life
BreastCancer	698	11	9	2	Life
BreastEW	568	31	15	2	Life
CongressEW	434	17	15	2	Social
DataR2	116	10	4	2	Life
HeartEW	270	14	10	2	Life
Ionosphere	351	35	13	2	Physical
Lung-cancer	32	57	31	2	Life
Lymphography	148	19	12	2	Life
M-of-n	1000	14	7	2	Life
sobar-72	72	20	16	7	Physical
Sonar	208	61	18	2	Life
SpectEW	267	23	14	2	Physical
Vote	300	17	15	2	Social
Wholesale customers data	440	8	4	2	Business
Wine	178	14	12	3	Physical
Zoo	101	17	10	6	Life

Table 5. Control parameters.

Algorithm	Parameters
BGA	Pop-Size = 10, Max-Iter = 20, Crossover-prob = 0.6, Muprob-min = 0.01, Muprob-max = 0.3
BBPSO	Number of particles = 5, Max-Iter = 30
BPSO	Pop-Size = 20, Max-Iter = 30, C1, C2 = 2, WMAX = 0.9, WMIN = 0.4
BSMO	Pop-Size = 10, Max-Iter = 20
BASO	Pop-size = 10, Max-Iter = 30, α = 50, β = 0.2
BGWO	Number of particles = 5, Max-Iter = 30
TMGWO	Mutation Probability = 0.5, Number of particles = 5, Max-Iter = 30
MFPA	Switch Probability = 0.8, Number of particles = 5, Max-Iter = 30
BSSA	Number of particles = 5, Max-Iter = 30

5.2. **Implementation of filter methods.** An information gain is a measure of the reduction in entropy resulting from the transformation of a dataset. Information gain is calculated by comparing each variable with the target variable in context. Correlation measures the linear relationship between two or more variables. Predicting one variable based on the other allows us to predict the other variable. When selecting features, correlation is used because good variables are highly correlated. Additionally, variables should be correlated with the target but not with each other. The relationship between two variables can be predicted if they are correlated. A model can therefore use only one feature if two features are correlated, as the second does not add any additional information to the model. The Spearman Correlation will be used here. As shown in Table 6, feature selection can be done using 3 filter methods, namely redundancy-relevance (filter phase of the proposed method), information gain, and correlation coefficient alone. With information gain, all features except the target feature are left out in datasets such as DataR2 and HeartEW, which is unacceptable, and a poor feature selection is evident. With the correlation coefficient, more features are selected in all data sets than with the filter method used in the proposed method. As a result, the proposed method has a better filter phase than the other two methods and selects the most informative features.

Table 6. Feature selection with filter methods.

Dataset		**Filter methods**	
	Redundancy-Relevance	**Information Gain**	**Correlation Coefficient**
Algerian forest fires	8	9	11
BreastCancer	9	9	10
BreastEW	15	15	20
CongressEW	15	6	17
DataR2	4	1	9
HeartEW	10	1	14
Ionosphere	13	21	35
Lung-cancer	31	48	54
Lymphography	12	5	19
M-of-n	7	1	14
sobar-72	16	4	19
Sonar	18	1	39
SpectEW	14	1	23
Vote	15	9	17
Wholesale customers data	4	4	7
Wine	12	12	13
Zoo	10	16	16

5.3. **Performance analysis of BEO and FWBEOSA.** BEO and FWBEOSA have both been compared against the datasets discussed in Section 5.1. As FWBEOSA uses two powerful search algorithms, BEO, which is efficient in exploration, and SA, which is robust in exploitation, it is expected that it will perform better in terms of classification accuracy. Utilizing BEO allows the exploration of highly relevant regions in the feature space without falling into the trap of local optimizations. The SA algorithm intensifies the nearby regions based on the best subset of features found by BEO. Because FWBEOSA used filter methods and discarded redundant and irrelevant features, they have chosen a much smaller number of features. This means FWBEOSA was able to identify informative and relevant features ignored by BEO. Based on Table 7, it is clear that BEO and FWBEOSA perform FS efficiently. The accuracy of 14 datasets in BEO exceeds or equals 90%, while 16 datasets in FWBEOSA do. Furthermore, BEO has achieved 100% accuracy on five datasets, while FWBEOSA has achieved 100% accuracy on nine datasets. In addition to their classification accuracy, FWBEOSA and BEO performed exceptionally well when it came to the number of features they used. On most datasets, BEO uses less than 50% or even fewer features for classification, while FWBEOSA uses less than 30% or even fewer features for classification, except on a few datasets. Table 8 displays the accuracy of classification and the number of features selected by FWBEOSA and BEO with KNN classifiers. FWBEOSA has 14 datasets with accuracy greater than 90%, as opposed to 12 for BEO. Additionally, BEO has produced 100% accuracy on two datasets as compared to FWBEOSA, which has produced 100% accuracy on seven datasets. A FWBEOSA selection of 16 datasets included fewer or the same number of features as a BEO selection. The discussion has concluded that both BEO and FWBEOSA are able to select the best set of features from datasets. In terms of financial simulation, both of these models are very competitive because they are accurate and use a limited number of features. Even though both BEO and FWBEOSA performed well in FS, FWBEOSA's results were superior. In FWBEOSA in order to improve the performance of BEO, SA and filter phase are crucial. Therefore, BEO utilizes the equilibrium pool's four particles to guide exploration. A pool consists of five particles averaged from four others. By contrast with exponential terms, it maintains a balance between exploration and exploitation. A particle may belong to the same search space in some cases when it has similar characteristics as those in the equilibrium pool. This causes that specific area to be massively exploited. To find the global optimum, the algorithm explores the entire search space. SA assists the BEO in exploring the search space and therefore aids the BEO in performing well in this context. BEO's exploration capabilities are improved through the use of exponential terms while exploration-exploitation trade-offs are maintained. A feature selection problem is modeled as an optimization problem aimed at reducing the number of features, the classification error and increase the rate of convergence. Therefore, the convergence graphs that have a downward slope (and reach convergence faster) indicate that the method is more likely to achieve convergence. Figure 5, 6 and 7 shows a comparison between FWBEOSA and BEO with SVM classifiers. According to the results in 14 datasets (82%) the proposed method FWBEOSA is better at finding an initial solution than BEO, and it is also better at finding a final solution in 13 datasets (76%). Compared to BEO, the proposed method has performed better or similar in 15 datasets (88%), based on the convergence rate. FWBEOSA and BEO with KNN classifiers are compared in Fig. 8, 9 and 10. Compared to BEO, the proposed method gets better initial solutions in 15 datasets (82%), and also has better final solutions in 15 datasets (88%).

TABLE 7. Performance of BEO and FWBEOSA (SVM Classifier).

No.	Dataset	BEO(Accuracy)	BEO(Features)	FWBEOSA(Accuracy)	FWBEOSA(Features)
1	Algerian forest fires	95.92	2	100	1
2	Breast cancer	98.57	2	100	3
3	BreastEW	100	6	100	3
4	CongressEW	98.85	1	100	2
5	DataR2	83.33	2	83.33	1
6	HeartEW	92.59	4	94.44	4
7	Ionosphere	95.77	11	95.77	2
8	lung-cancer	100	5	100	2
9	Lymphography	96.67	6	96.67	3
10	M-of-n	87.50	4	90.50	4
11	sobar-72	100	2	100	2
12	Sonar	92.85	16	95.23	5
13	SpectEW	90.74	15	94.44	8
14	Vote	98.33	1	100	1
15	Wholesale customers data	76.14	1	94.32	2
16	Wine	100	3	100	2
17	Zoo	100	6	100	4

TABLE 8. Performance of BEO and FWBEOSA (KNN Classifier).

No.	Dataset	BEO(Accuracy)	BEO(Features)	FWBEOSA(Accuracy)	FWBEOSA(Features)
1	Algerian forest fires	98.95	2	100	1
2	Breast cancer	97.85	3	100	3
3	BreastEW	98.24	3	98.24	2
4	CongressEW	98.85	3	100	3
5	DataR2	83.33	2	83.33	1
6	HeartEW	88.88	5	94.44	4
7	Ionosphere	95.77	6	98.59	3
8	lung-cancer	100	15	100	7
9	Lymphography	93.33	6	96.66	6
10	M-of-n	84.50	5	89.00	4
11	sobar-72	100	3	100	2
12	Sonar	92.85	8	95.23	5
13	SpectEW	87.04	7	96.30	8
14	Vote	96.66	2	100	2
15	Wholesale customers data	94.31	1	89.77	1
16	Wine	97.22	4	100	3
17	Zoo	81.25	2	93.75	2

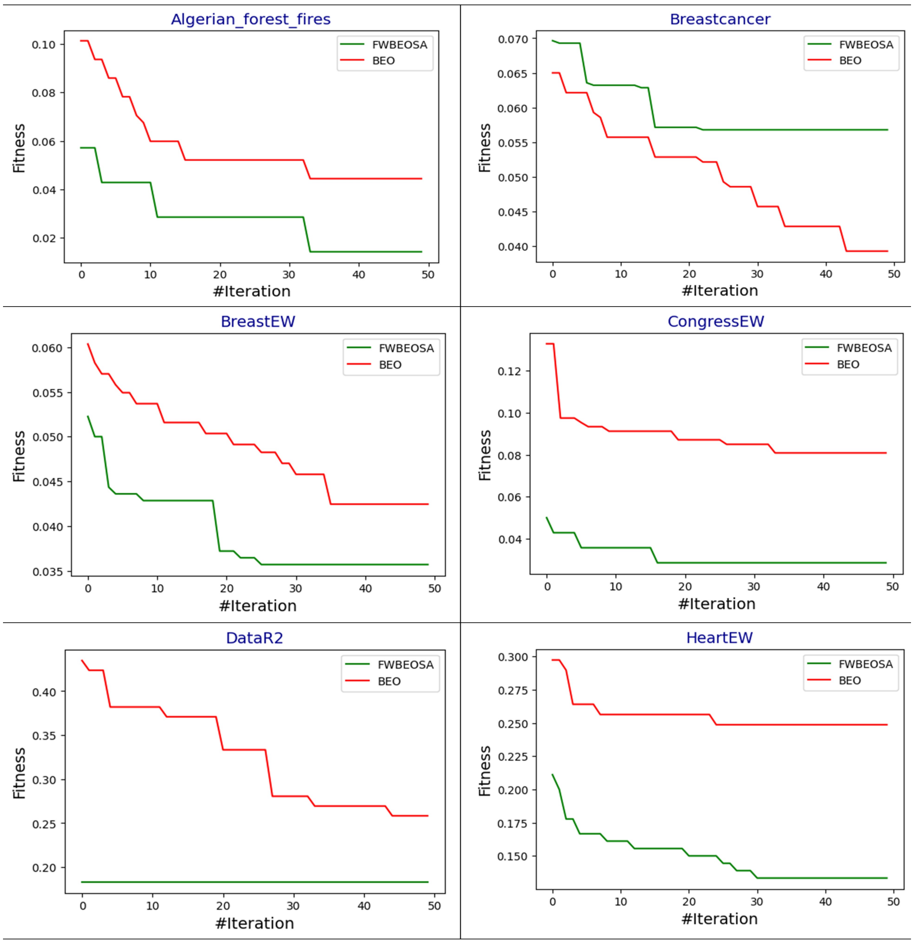

Figure 5. Convergence graphs for 17 UCI datasets (SVM Classifier).

5.4. **Definition of comparison methods.** This section presents the results of FWBEOSA's proposed algorithm. FWBEOSA has been compared with BEO and nine binary FS works as a result of our comparison. The list includes:

- *GWO:* Gray wolves live in packs, a predatory lifestyle that is derived from their ways of life. The predators are arranged in an alphabetical hierarchy from alpha to gamma. An optimizer model of gray wolves simulates their tracking, encirclement and attack phases [19].
- *ASO:* The atom search optimization algorithm combines potential function, interaction force, and geometric constraint with the movement

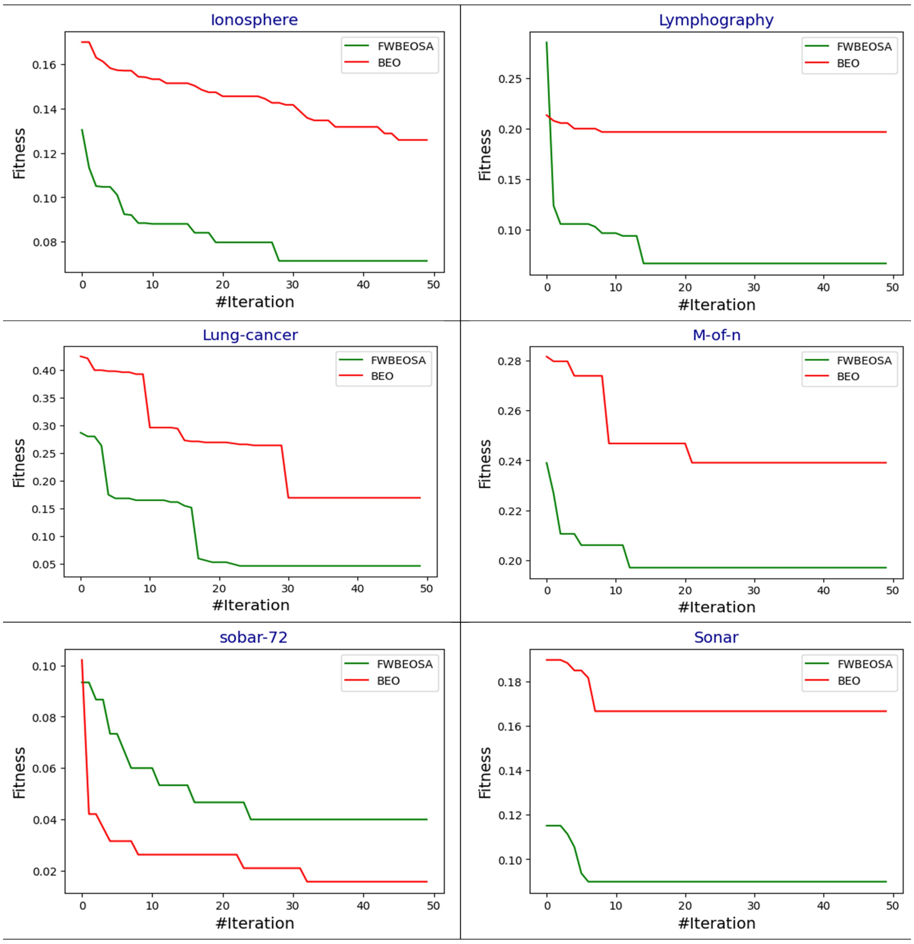

FIGURE 6. Convergence graphs for 17 UCI datasets (SVM Classifier).

principle of atoms. Metaheuristic algorithms generate atoms of solutions at first [36].

- *SMO:* In SMO, people mimic the behaviors of others in society. It is through imitation of famous people that individuals try to assimilate themselves to them. By imitating the parameters of optimal solutions, each solution in optimization problems can reach global optimality. It is possible to model imitation behavior as randomly searching solution spaces in optimization problems. In each iteration, the optimal value of each solution is compared to the global optimal value obtained in

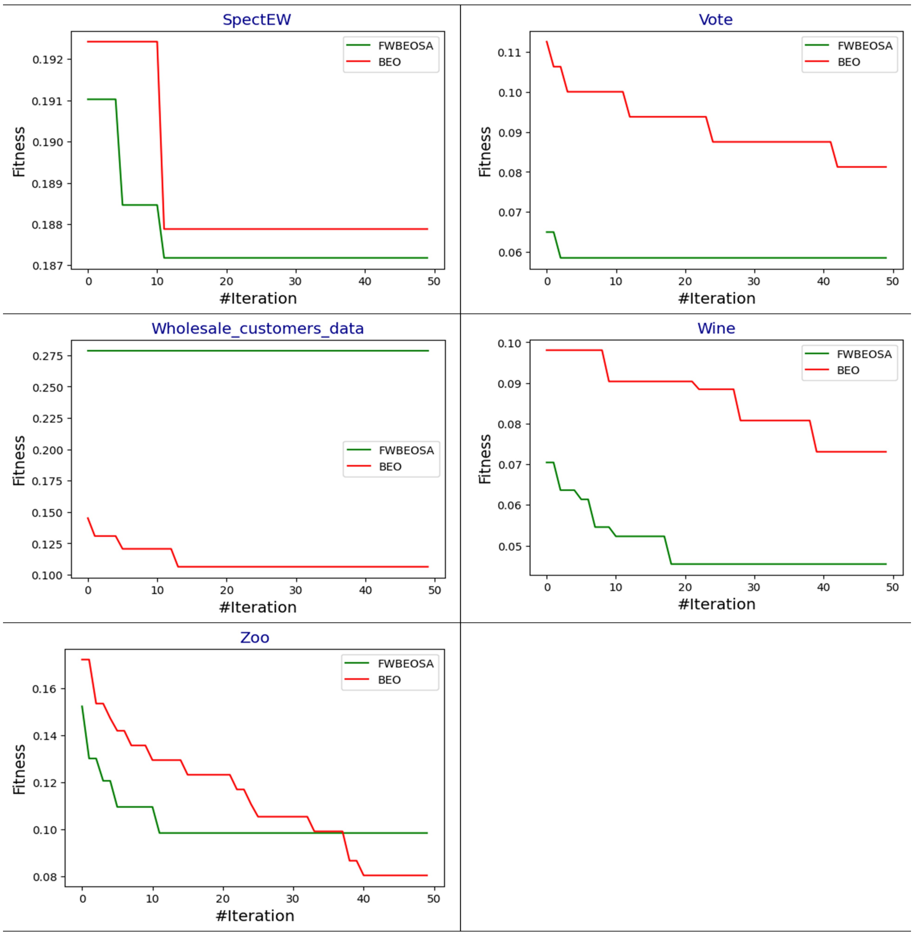

Figure 7. Convergence graphs for 17 UCI datasets (SVM Classifier).

the previous iteration in order to determine its difference from the global optimal value. Based on this difference value, a random search is conducted to find the best solution [10].

- *PSO:* Birds and fish flocking behavior inspired PSO algorithm. The algorithm is considered a swarm intelligent algorithm since it is based on population [19].
- *SSA:* Planktonic tunicates called salps belong to the Salpidae family and mimic the behavior of these animals. Also, their tissues resemble

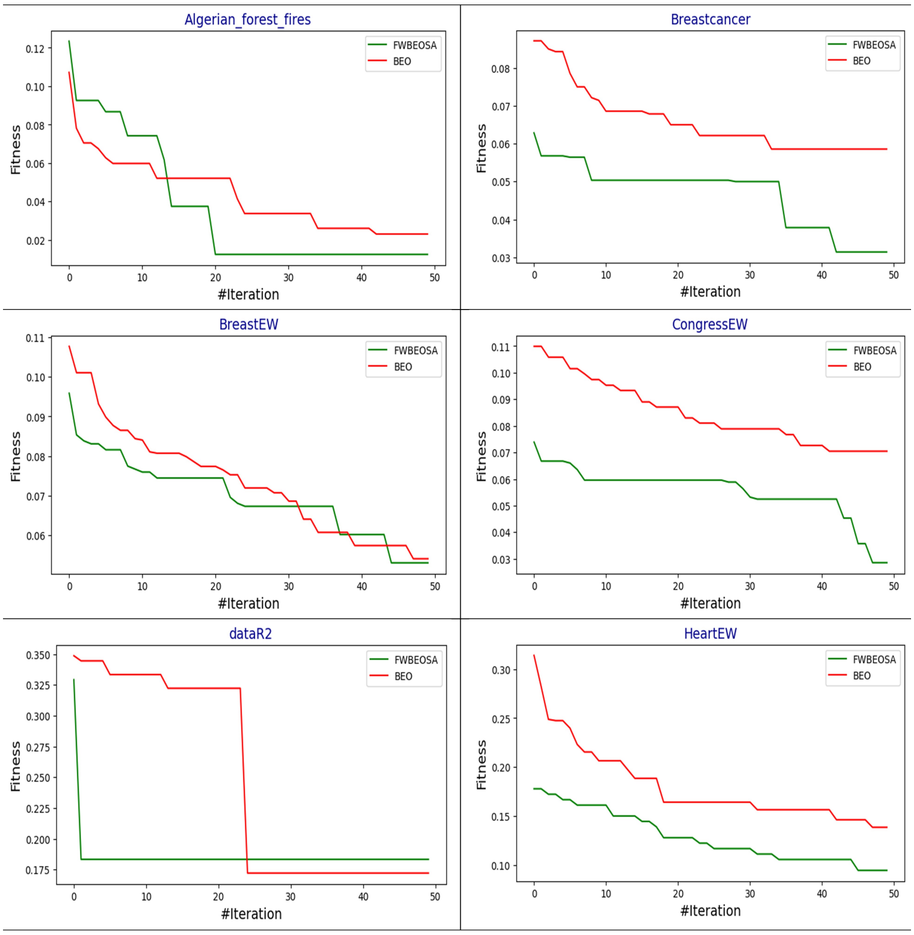

FIGURE 8. Convergence graphs for 17 UCI datasets (KNN Classifier).

those of jellyfishes, and their movement and weight are heavily water-based. Water is pumped through their jelly bodies to propel them forward. It appears that salps use a fast harmonious change to move and forage in oceans, which may help them do better foraging and movement [20].

- *GA:* Based on Darwin's notion of natural selection and evolution, optimization search strategies are used. Under the "selective pressure" of the object function, a set of trial solutions is selected and "evolved" toward an optimal solution [22].

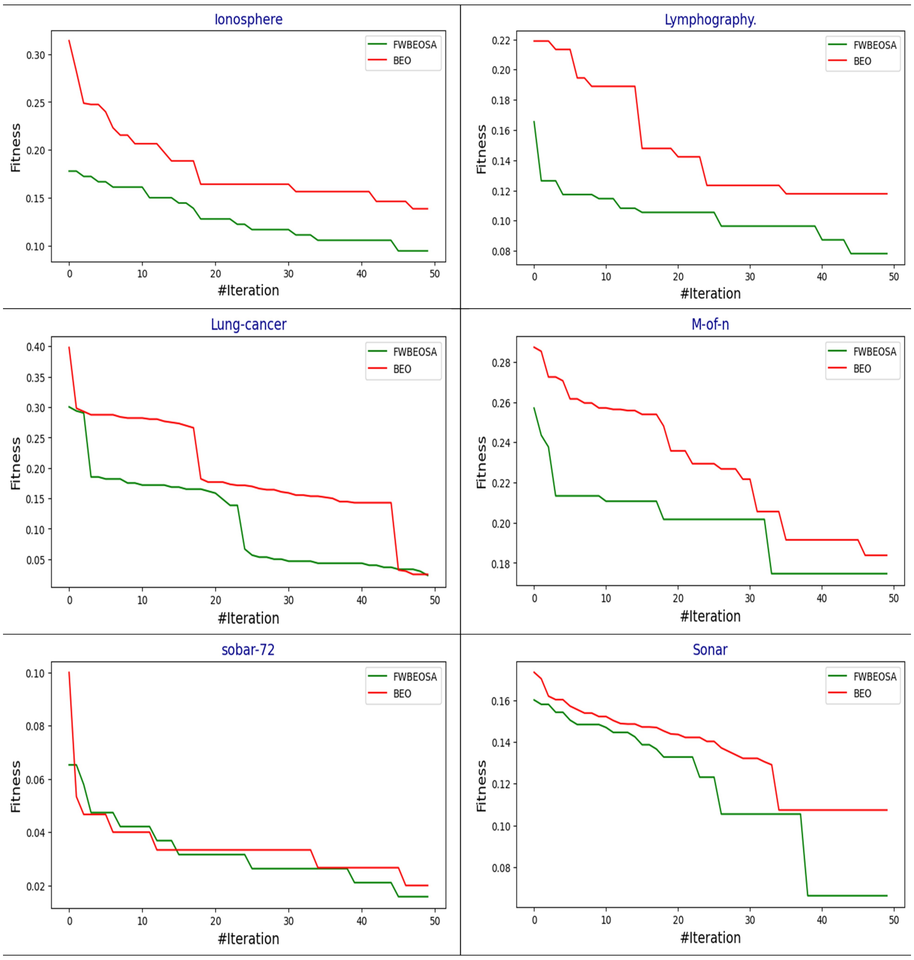

FIGURE 9. Convergence graphs for 17 UCI datasets (KNN Classifier).

- *MFPA:* Researchers have been increasingly interested in Flower Pollination Algorithm (FPA) in the area of computational intelligence. In many optimization problems, it is efficient and simple to search for global optimality. MFPA aims to develop a new variant of FPA aimed at improving the convergence rate and solution quality. It will be called the Modified FPA. In order to better utilize existing solutions, the MFPA extracts their characteristics and directs the exploration process towards specific areas [32].

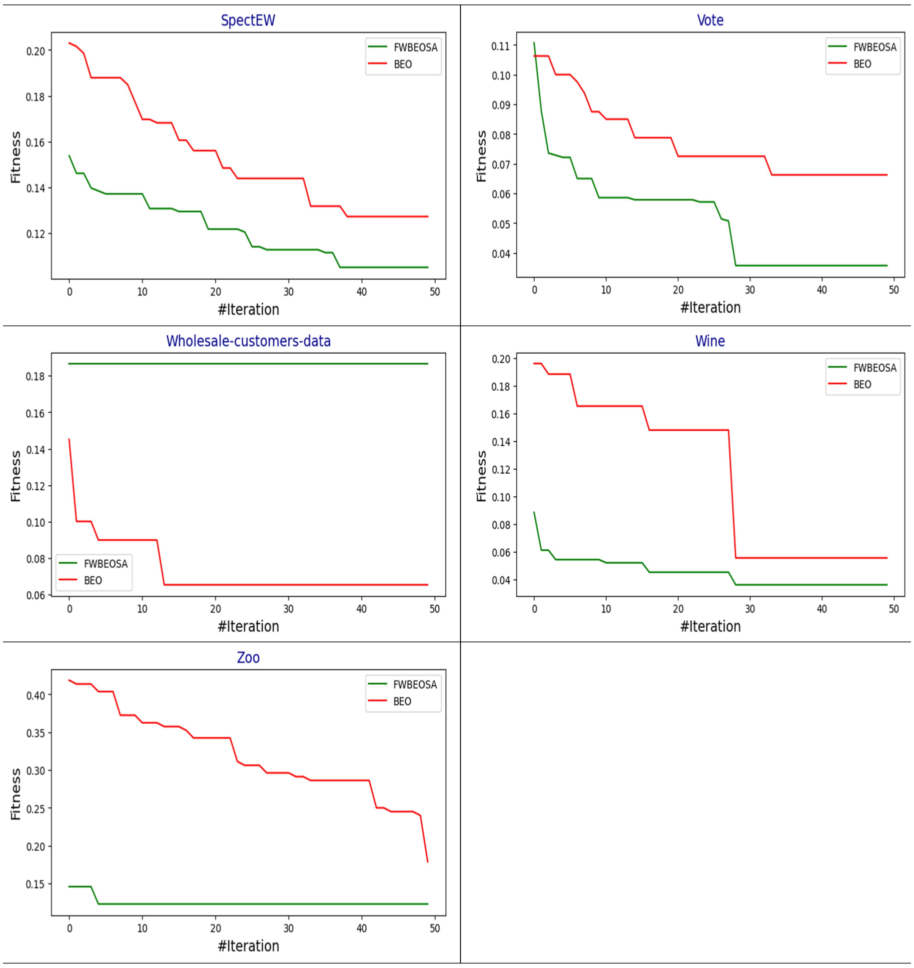

Figure 10. Convergence graphs for 17 UCI datasets (KNN Classifier).

- *BBPSO:* Particle swarm algorithms are just complicated enough to be difficult to understand. While the formula is very simple, it is even easier to describe how the algorithm works verbally, but it is very difficult to comprehend in one's mind how particles move around constantly shifting centers. By removing some conventional features from particle swarms, BBPSO discovers their operating principles. As a result, some of the algorithm's mysteries will be revealed, its similarities to other stochastic populations-based methods of solving problems will be explored, and suggestions or implied new directions will be explored [23].

- *TMGWO:* Two-phase Mutation Gray Wolf Optimization solves the classification problem using two-phase mutation. A two-phase mutation improves the algorithm's ability to be exploited. During the first mutation phase, features are reduced while high classification accuracy is maintained. The second mutation phase aims to increase the classification accuracy by adding more informative features. Two phase mutations are possible with a small probability due to the time-consuming mutation phase [4].

5.5. **Comparison of the proposed method and the other methods.** Table 9 shows that FWBEOSA achieves a good classification accuracy using SVM classifier. FWBEOSA performs the best in 13 cases (76%), due to its unique characteristics in finding optimal solutions and balancing exploration and exploitation. The SA algorithm succeeded in enhancing the best solution determined by the BEO algorithm. There are bold numbers in all tables which represent the best performances. To display differences accurately and reduce errors, four decimal places are used. On the basis of the number of features selected by the SVM classifier, FWBEOSA performs well due to the advantages of the filter method along with the wrapper method. According to Table 10, FWBEOSA performed the best in 15 datasets (88%). It performed well in both other datasets, demonstrating its superiority over existing methods. In six cases, BEO achieved highest and chose fewer features. In addition to increasing classification accuracy, FWBEOSA reduced the selection of features and the objectives. According to Table 11, the proposed method provides higher classification accuracy than other methods within the KNN classifier. Based on 14 data sets out of 17 data sets, the proposed method achieved the highest level of classification accuracy (82%). According to Table 12, the proposed method selects fewer features than other methods. The proposed method selects the least number of features in 13 of 17 data sets (76%), and selects more features than other methods in just three data sets: Breastcancer, Lymphography, and SpectEW. In comparison to BEO and other methods, the proposed method has better classification accuracy and a greater number of features selected, and is significantly better than BEO in terms of performance. In data analysis, boxplots are common for summarizing quantitative and qualitative data. An upper and lower quartile boxplot is shown along with the minimum and maximum range values and the median [21]. Figure 11, 12 and 13 shows a boxplot of the proposed method (FWBEOSA) along with other methods in a classification using SVM classifier. Using this diagram, it is evident that the proposed method has a median equal to or higher than other methods in 13 datasets (76%) and its dispersion is lower in 12 datasets (71%). Furthermore, most datasets contain fewer outliers. Figure 14, 15 and 16 shows the boxplot of the proposed method (FWBEOSA) compared to other methods in classification with a KNN classifier. The proposed method has a median equal to or higher than other methods in 12 datasets (71%), and its dispersion in 11 datasets (65%) is less than other methods.

TABLE 9. Classification accuracy obtained by FWBEOSA and other methods (SVM Classifier).

Datasets	FWBEOSA	BEO	BGWO	BSSA	BGA	BSMO	BPSO	BASO	MFPA	BBPSO	TMGWO
Algerian forest fires	**1**	0.9592	0.9864	0.9864	0.9795	**1**	**1**	0.9836	0.9864	0.9795	0.9729
BreastCancer	**1**	0.9857	0.9857	0.9904	0.9785	0.9785	0.9785	0.9771	0.9857	0.9704	0.9761
BreastEW	**1**	**1**	0.9941	0.9941	0.9736	**1**	0.9736	0.9859	0.9815	0.9849	0.9909
CongressEW	**1**	0.9885	0.9847	0.9618	0.9540	0.9770	0.9885	0.9816	0.9847	0.9894	0.9847
DataR2	0.8333	0.8333	0.8571	**0.8857**	0.6666	0.8750	0.8333	0.8275	0.8570	0.8742	0.8557
HeartEW	**0.9444**	0.9259	0.9012	0.8765	0.8888	0.8888	0.8888	0.9117	0.8750	0.9012	0.9135
Ionosphere	0.9577	0.9577	**0.9907**	0.9622	0.9295	0.9436	0.9154	0.9318	0.9428	0.9639	0.9751
Lung-cancer	**1**	**1**	0.8000	0.8000	0.8571	0.8571	0.8571	0.8750	0.9523	0.9135	0.9295
Lymphography	**1**	0.9667	0.9111	0.9111	0.9000	0.9333	0.9666	0.9729	0.9888	0.9455	0.9333
M-of-n	0.9050	0.8750	**1**	**1**	**1**	**1**	**1**	**1**	0.9907	**1**	**1**
sobar-72	**1**	**1**	0.9545	0.9545	0.9333	0.9333	**1**	**1**	**1**	**1**	0.9545
Sonar	**0.9523**	0.9285	0.9047	0.9206	0.9285	0.8809	0.9047	0.9038	0.9465	0.9333	0.9365
SpectEW	**0.9444**	0.9074	0.8888	0.8888	0.8703	0.8888	0.9074	0.8955	0.9012	0.9035	0.9159
Vote	**1**	0.9833	0.9666	0.9666	0.9666	0.9833	0.9666	0.9733	0.9777	0.9888	0.9777
Wholesale customers data	0.7614	**0.9431**	0.9242	0.9318	0.8863	**0.9431**	0.8977	0.9090	0.9393	0.9431	0.9242
Wine	**1**	**1**	0.9259	0.8888	**1**	**1**	**1**	**1**	0.9888	0.9777	0.9529
Zoo	**1**	**1**	0.9677	0.9354	**1**	**1**	**1**	**1**	**1**	**1**	**1**

TABLE 10. The number of selected features by FWBEOSA and other methods (SVM Classifier).

Datasets	FWBEOSA	BEO	BGWO	BSSA	BGA	BSMO	BPSO	BASO	MFPA	BBPSO	TMGWO
Algerian forest fires	**1**	2	7	3	3	5	2	4	4	2	5
BreastCancer	3	**2**	6	4	**2**	3	3	4	4	3	6
BreastEW	**3**	6	18	12	11	7	12	16	5	8	5
CongressEW	2	2	9	4	3	**1**	4	4	3	3	8
DataR2	**1**	2	3	4	4	3	7	3	4	5	2
HeartEW	**4**	**4**	7	9	8	10	10	7	6	8	5
Ionosphere	**2**	11	15	24	14	21	14	19	10	11	13
Lung-cancer	**2**	4	18	23	13	42	11	22	8	9	15
Lymphography	**6**	**6**	9	10	11	13	12	11	**6**	10	7
M-of-n	**4**	**4**	8	7	10	9	10	8	5	9	6
sobar-72	**2**	**2**	6	5	5	6	4	3	7	**2**	5
Sonar	**5**	16	27	32	35	25	25	21	22	19	23
SpectEW	**8**	15	12	11	9	6	13	15	11	9	15
Vote	**1**	**1**	4	8	2	10	3	3	3	2	4
Wholesale customers data	**1**	2	4	4	2	4	2	2	5	3	2
Wine	**2**	3	7	6	5	8	4	6	3	4	5
Zoo	**4**	6	8	10	6	6	7	7	5	6	**4**

A FWBEOSA method uses the filter and wrapper techniques as a means of using the speed of the filter methods at the beginning of the feature selection process and discarding the extra and irrelevant features. A binary balance optimizer is also employed in the wrapping step to select the optimal set of features, since the method has weaknesses and may get locked into a local optimum. In addition to improving the speed of filter methods and the accuracy of wrapping methods, the thermal simulator combined with BEO is one of the most efficient ways to find the local optimum. FWBEOSA has achieved a higher accuracy

Table 11. Classification accuracy obtained by FWBEOSA and other methods (KNN Classifier).

Datasets	**FWBEOSA**	**BEO**	**BGWO**	**BSSA**	**BGA**	**BSMO**	**BPSO**	**BASO**	**MFPA**	**BBPSO**	**TMGWO**
Algerian forest fires	**1**	0.9895	0.9864	**1**	0.9795	0.9883	0.9795	0.9836	**1**	0.9864	0.9864
BreastCancer	**1**	0.9785	0.9904	0.9857	0.9785	0.9857	0.9642	0.9828	0.9904	0.9904	0.9904
BreastEW	**0.9824**	**0.9824**	0.9766	0.9766	0.9649	0.9649	0.9824	0.9577	0.9707	0.9707	0.9766
CongressEW	**1**	0.9885	0.9923	0.9923	0.9770	0.9770	0.9885	0.9908	0.9923	0.9770	0.9847
DataR2	0.8333	0.8333	0.8857	**0.9142**	0.8333	0.8333	0.8333	0.8965	**0.9142**	**0.9142**	0.8857
HeartEW	**0.9444**	0.8888	0.9135	0.9012	0.8888	0.8703	0.9074	0.8676	0.8888	0.9135	0.9012
Ionosphere	**0.9859**	0.9577	0.9433	0.9339	0.9259	0.9436	0.9295	0.9090	0.9528	0.9433	0.9433
Lung-cancer	**1**	**1**	**1**	**1**	0.8577	0.8577	**1**	**1**	**1**	**1**	**1**
Lymphography	**0.9666**	0.9333	0.9333	0.9333	0.9333	0.9000	0.9000	0.9189	0.9333	0.9555	0.9555
M-of-n	0.8900	0.8450	**1**	**1**	0.9400	0.9050	0.9700	0.9920	**1**	**1**	**1**
sobar-72	**1**	**1**	**1**	**1**	0.9333	**1**	**1**	**1**	**1**	**1**	**1**
Sonar	**0.9523**	0.9285	0.9365	0.9365	0.9285	**0.9523**	0.9285	0.9230	0.9365	0.9365	0.9365
SpectEW	**0.9630**	0.8704	0.9259	0.9135	0.7962	0.8333	0.9074	0.9104	0.9259	0.8888	0.9259
Vote	**1**	0.9666	0.9888	0.9888	0.9833	0.9333	**1**	0.9866	0.9888	0.9888	0.9777
Wholesale customers data	0.8977	0.9431	0.9469	0.9469	**0.9659**	0.9545	0.9318	0.9545	0.9545	0.9621	0.9621
Wine	**1**	0.9722	0.9814	**1**	0.9722	0.9722	0.9444	0.9777	0.9814	0.9814	0.9814
Zoo	**0.9375**	0.8125	0.8695	0.8695	0.7500	0.8125	0.8125	0.8421	0.9130	0.9130	0.9130

Table 12. The number of selected features by FWBEOSA and other methods (KNN Classifier).

Datasets	**FWBEOSA**	**BEO**	**BGWO**	**BSSA**	**BGA**	**BSMO**	**BPSO**	**BASO**	**MFPA**	**BBPSO**	**TMGWO**
Algerian forest fires	**1**	2	6	2	3	2	**1**	4	3	2	2
BreastCancer	3	3	6	5	**2**	3	**2**	4	6	4	3
BreastEW	**2**	3	18	5	9	5	7	7	**2**	6	3
CongressEW	**3**	**3**	7	10	8	4	4	8	5	5	4
DataR2	**1**	2	4	3	4	2	2	4	5	5	5
HeartEW	4	5	**3**	**3**	7	4	**3**	4	5	7	4
Ionosphere	**3**	6	10	10	9	5	7	5	9	11	4
Lung-cancer	**7**	15	24	20	9	14	29	28	18	16	9
Lymphography	6	6	8	6	9	11	8	9	8	8	**4**
M-of-n	**4**	5	6	6	10	9	8	8	6	6	6
sobar-72	**2**	3	5	4	4	13	3	4	3	5	4
Sonar	**5**	8	25	24	32	27	25	19	20	24	16
SpectEW	8	7	9	11	**1**	9	14	13	12	3	13
Vote	**2**	**2**	11	5	4	4	4	7	5	4	3
Wholesale customers data	**1**	**1**	4	2	4	2	2	3	2	3	3
Wine	**3**	4	**3**	5	5	6	**3**	6	**3**	**3**	**3**
Zoo	**2**	**2**	6	8	11	15	**2**	12	8	8	7

and a lower number of features due to the advantages mentioned in section 5.3 and when compared to BEO. A major advantage of the proposed method over BEO is that it has a better convergence rate and, thus, a better final solution and performs better on most data sets. When compared with BEO and 9 other methods of feature selection in the SVM classifier, the proposed method performs significantly better in accuracy criteria as well as the number of features selected. It is also evident that the proposed method is superior to KNN classification since it has strong filter phases and an appropriate balance between exploration and exploitation. As can be seen from the box plot, the proposed method has also produced better data quality.

There are various domains and applications in which the FWBEOSA algorithm might be useful. In data science and machine learning may have the following applications and implications:

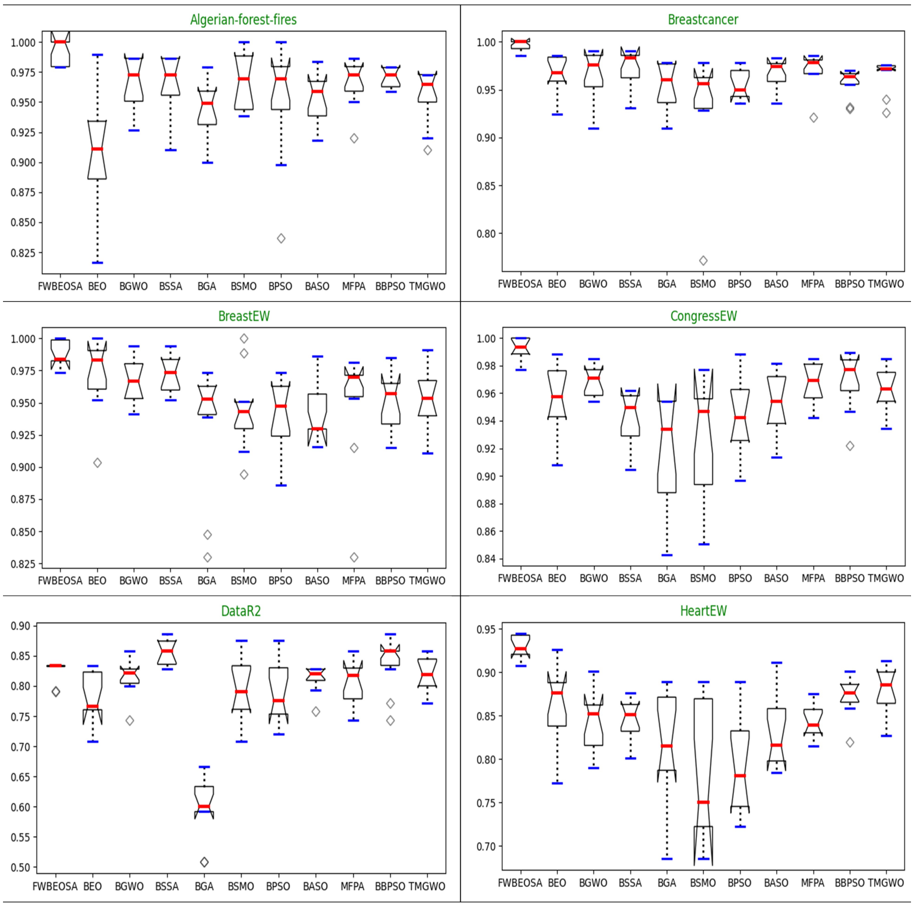

FIGURE 11. The Boxplot of FWBEOSA and other method for the 17 UCI datasets (SVM classifier).

- *High-Dimensional Data Analysis:* High-dimensional data is widely used in fields such as bioinformatics, image processing, text mining, and sensor data analysis. With FWBEOSA and machine learning models, data dimensionality can be reduced and performance improved.
- *Bioinformatics:* Genomic and bioinformatics researchers work with high-dimensional datasets representing gene expression profiles, genomic sequences, and proteomic data. Identifying the most informative genes or markers is essential to predicting gene functions, discovering biomarkers, and categorizing diseases. Selecting the most relevant genes or genetic markers can be made easier via hybrid feature selection.

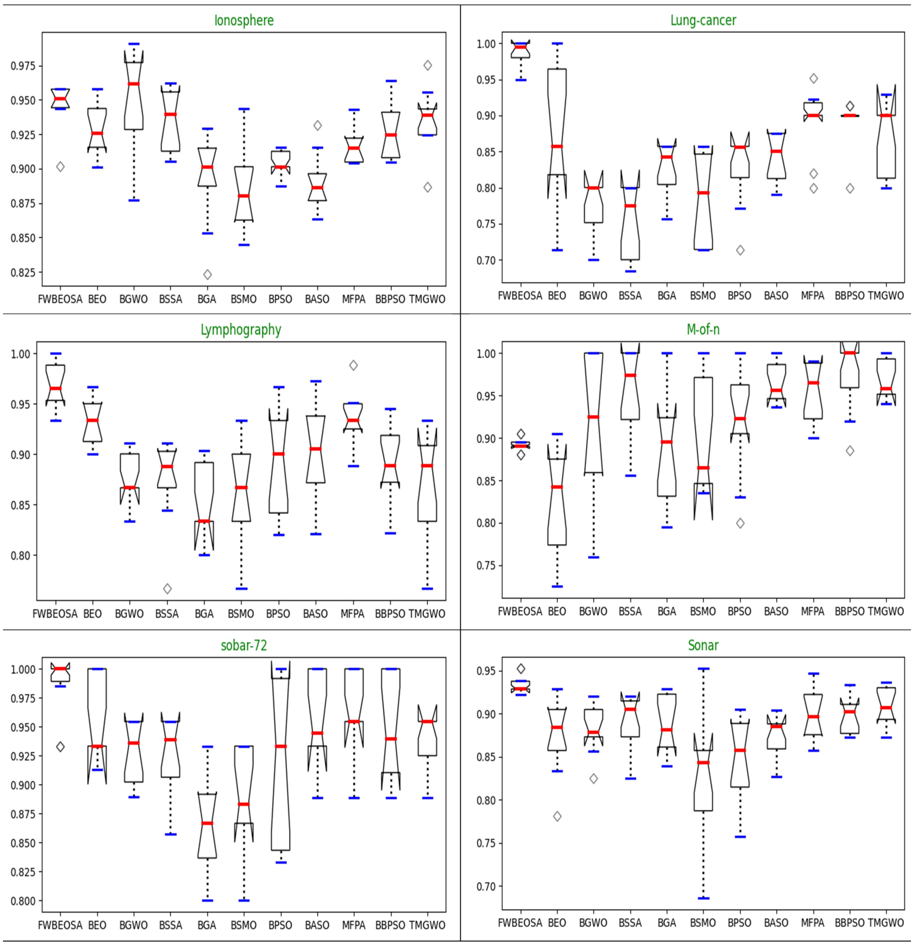

Figure 12. The Boxplot of FWBEOSA and other method for the 17 UCI datasets (SVM classifier).

- *Image and Signal Processing:* Various features are represented by pixels or data points in images and signals. Image classification and signal processing can be improved with feature selection by reducing noise and irrelevant information.
- *Natural Language Processing:* When text mining is performed on documents or text data, multidimensional feature vectors are often employed. It is important to select features correctly when performing sentiment analysis, topic modeling, and document categorization.

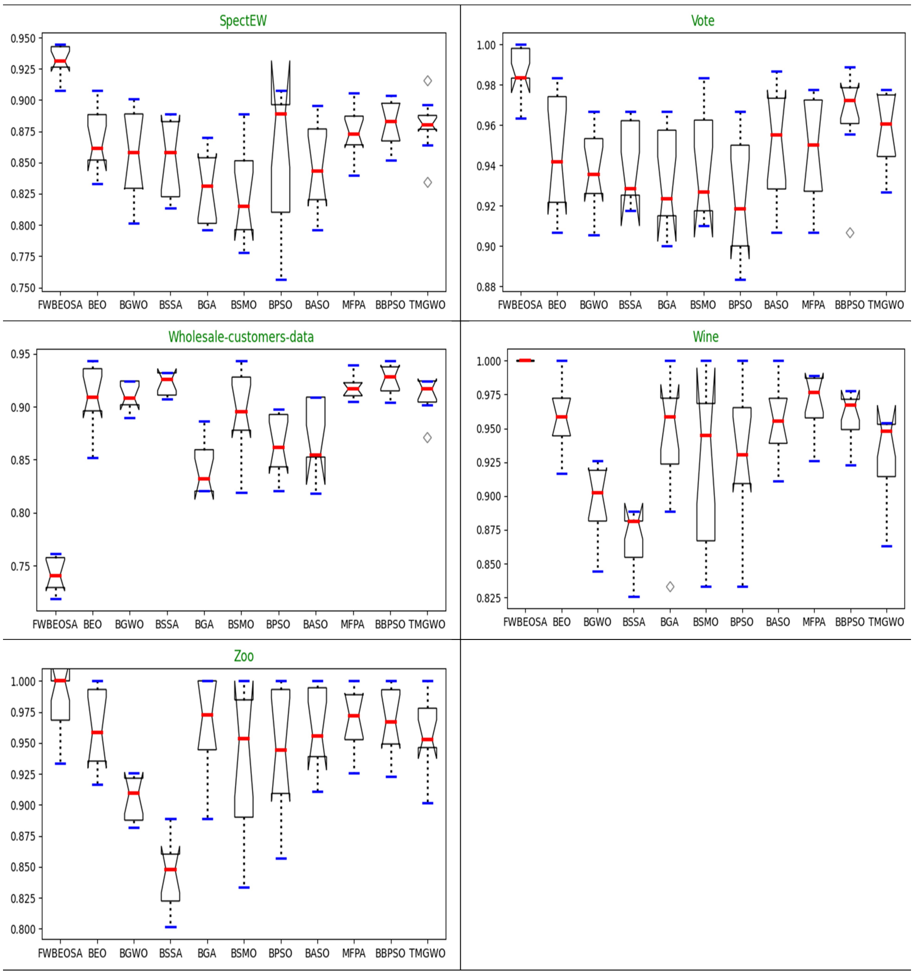

FIGURE 13. The Boxplot of FWBEOSA and other method for the 17 UCI datasets (SVM classifier).

- *Network Analysis:* The hybrid filter-wrapper feature selection method can be used to identify the most influential nodes or features in complex networks. The approach can be helpful in identifying central nodes in social networks, analyzing disease spread in epidemiological networks, and detecting anomalies in network traffic.
- *Interpretability and Resource Efficiency:* The purpose of feature selection techniques is to select a subset of relevant features to improve

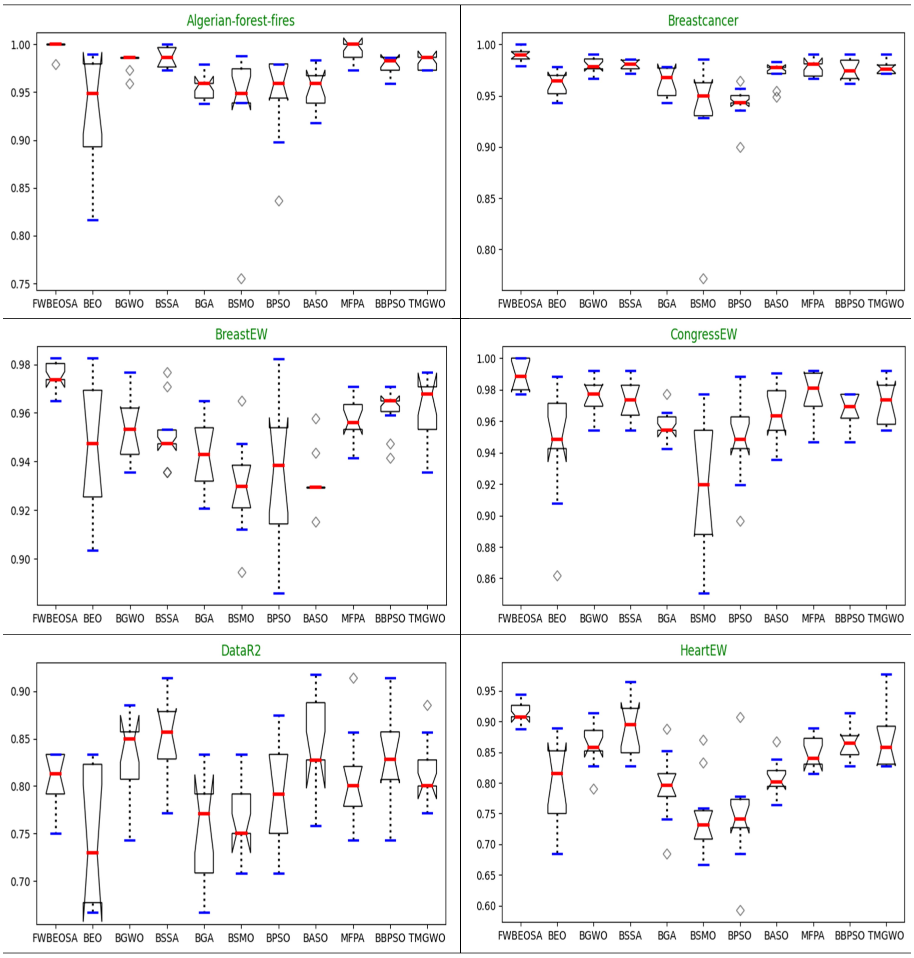

Figure 14. The Boxplot of FWBEOSA and other method for the 17 UCI datasets (KNN classifier).

model interpretability. Machine learning models can be more resource-efficient when they have fewer features.

- *Ensemble Learning and Model Aggregation:* Ensemble learning methods can be improved by selecting different and informative features for each base model. As a result, the aggregation process will become more accurate and robust.

Data analysis that involves high-dimensional data can benefit from FWBEOSA and hybrid filter-wrapper feature selection. Machine learning and data science in practice benefit from their ability to select relevant features efficiently, as

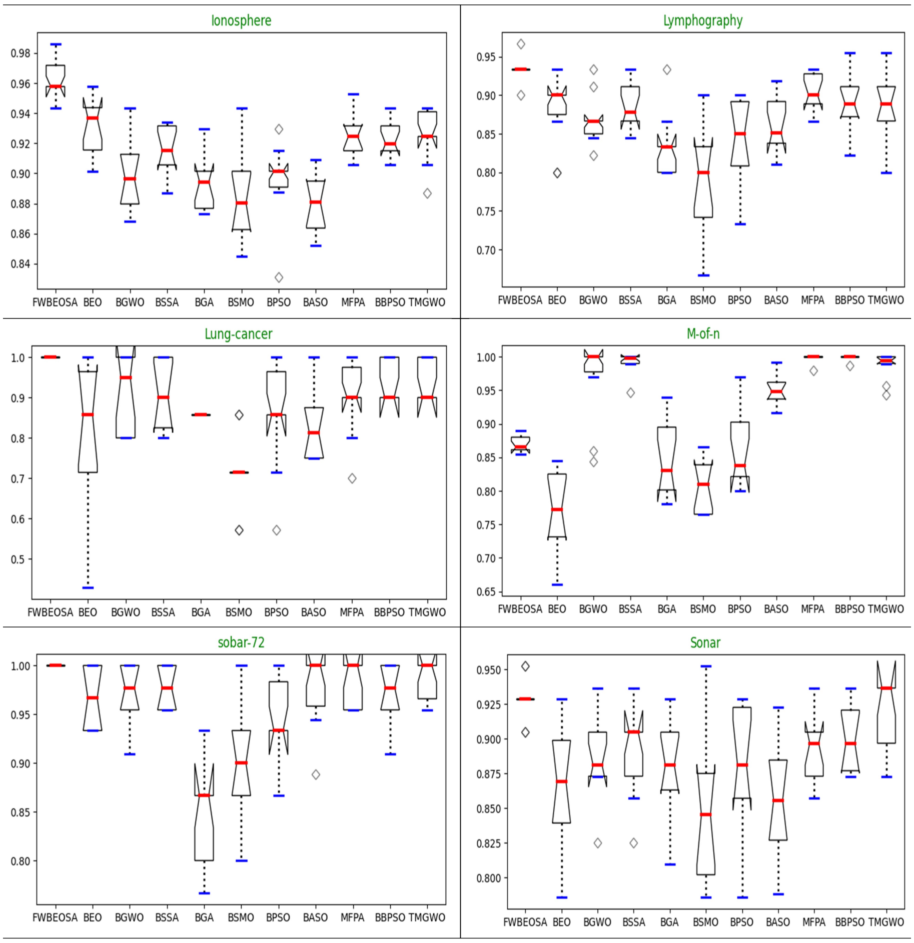

FIGURE 15. The Boxplot of FWBEOSA and other method for the 17 UCI datasets (KNN classifier).

their ability to improve performance, resource efficiency, and interpretability can lead to improved model performance, efficiency, and interpretability. To achieve the best results in real-world applications, it is crucial to tune parameters appropriately and validate on specific datasets.

6. Conclusion and Future Work

To address feature selection effectively, this paper uses a binary variant of Equilibrium Optimizer (EO). A sigmoid function transforms continuous values

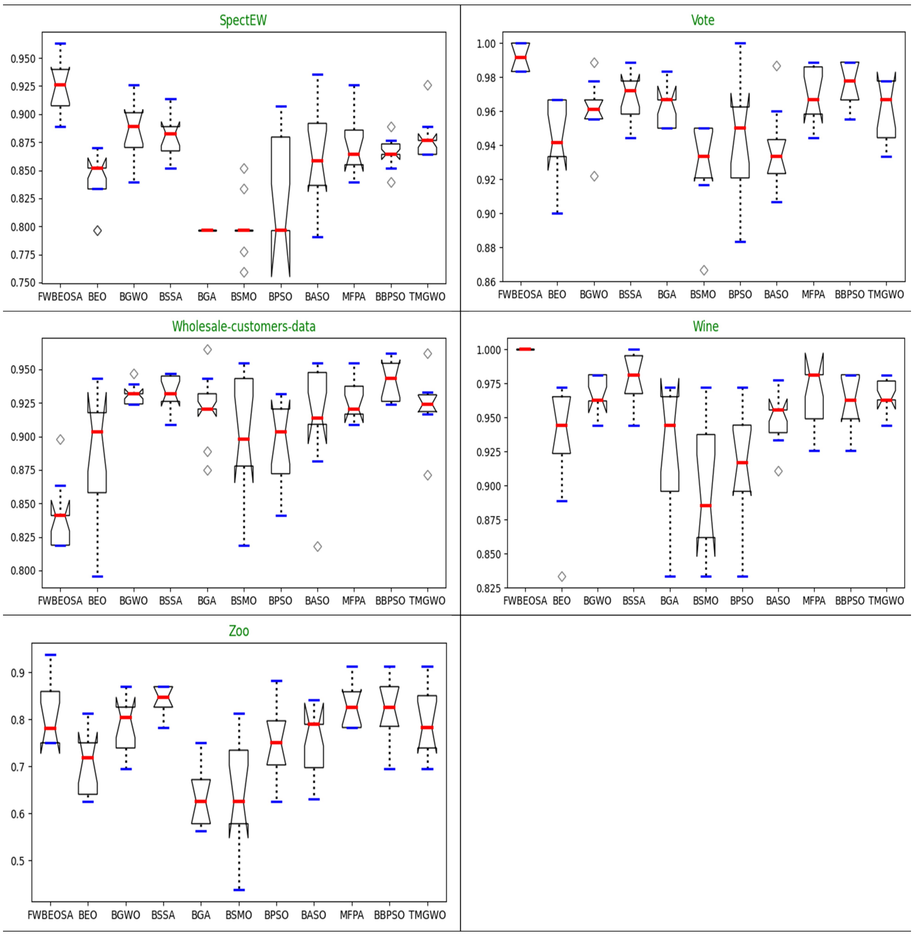

Figure 16. The Boxplot of FWBEOSA and other method for the 17 UCI datasets (KNN classifier).

of EO into binary search spaces. Most feature selection methods are prematurely convergent. In the proposed method FWBEOSA, BEO is combined with Simulation Annealing (SA) to overcome the problems mentioned above. The proposed method also provided useful and relevant features for the wrapper phase through the combination of filter methods. The SVM and KNN classifiers are evaluated on 17 popular UCI datasets, and they are evaluated using a variety of metrics, such as classification accuracy, the number of features selected, and convergence charts. Compared to other methods, the proposed method achieves maximum classification accuracy on 76% of datasets with

SVM and 82% with KNN. There is a clear superiority of the proposed method based on selected features in both classifiers. Additionally, filtering methods have helped the method to become more efficient, save money, and save time. Nevertheless, some problems remain.

- *Hyperparameter Tuning:* Finding the optimal values for the hyperparameters for the EO-SA algorithm can be a time-consuming task. A good hyperparameter selection is crucial for avoiding overfitting and achieving good performance.
- *Limited Scalability:* Datasets with high dimensionality and many features may not be well suited to EO-SA. It may become computationally prohibitive when the population size and search space become large enough.
- *Computational Complexity:* The algorithm's execution time may increase with particles, iterations, and pool size as the dataset grows.

These limitations can be overcome in several ways:

- *Parallelization:* EO-SA iterations can be parallelized for faster feature selection on multi-core processors by parallelizing computationally intensive parts of the code.
- *Hyperparameter Optimization:* Grid and random search techniques can be used to find optimal hyperparameter values for EO-SA, SVM, and KNN classifiers.
- *Algorithm Selection:* The best feature selection algorithm will be determined by comparing it with other metaheuristic approaches and compare it with EO-SA.
- *Evaluation with Different Classifiers:* Using various classifiers (including SVM and KNN) evaluate the selected features to see if they generalize well.

The application of evolutionary algorithms to deep learning has been successfully done in recent years, for example, when genetic programming is involved. During the future study, deep learning will also be examined as a possibility for combining the algorithm. The feature selection process can be improved to become more efficient, robust, and applicable to a broader range of datasets and tasks. The proposed FWBEOSA algorithm will be tested on NP-hard problems in the future, such as cloud-based task ordering and neural network hyperparameter optimization.

References

[1] Adamu, A., Abdullahi, M., Junaidu, S. B., & Hassan, I. H. (2021). An hybrid particle swarm optimization with crow search algorithm for feature selection. Machine Learning with Applications, 6, 100108. https://doi.org/10.1016/j.mlwa.2021.100108

[2] Abualigah, L., & Dulaimi, A. J. (2021). A novel feature selection method for data mining tasks using hybrid sine cosine algorithm and genetic algorithm. Cluster Computing, 24, 2161-2176. https://doi.org/10.1007/s10586-021-03254-y

[3] Abdel-Basset, M., Ding, W., & El-Shahat, D. (2021). A hybrid Harris Hawks optimization algorithm with simulated annealing for feature selection. Artificial Intelligence Review, 54, 593-637. https://doi.org/10.1007/s10462-020-09860-3

[4] Abdel-Basset, M., El-Shahat, D., El-Henawy, I., De Albuquerque, V. H. C., & Mirjalili, S. (2020). A new fusion of grey wolf optimizer algorithm with a two-phase mutation for feature selection. Expert Systems with Applications, 139, 112824. https://doi.org/10.1016/j.eswa.2019.112824

[5] Agrawal, P., Abutarboush, H. F., Ganesh, T., & Mohamed, A. W. (2021). Metaheuristic algorithms on feature selection: A survey of one decade of research (2009-2019). Ieee Access, 9, 26766-26791. https://doi.org/10.1109/ACCESS.2021.3056407

[6] Agrawal, P., Ganesh, T., & Mohamed, A. W. (2021). Chaotic gaining sharing knowledge-based optimization algorithm: an improved metaheuristic algorithm for feature selection. Soft Computing, 25(14), 9505-9528. https://doi.org/10.1007/s00500-021-05874-3

[7] Al-Wajih, R., Abdulkadir, S. J., Aziz, N., Al-Tashi, Q., & Talpur, N. (2021). Hybrid binary grey wolf with Harris hawks optimizer for feature selection. IEEE Access, 9, 31662-31677. https://doi.org/10.1109/ACCESS.2021.3060096

[8] Ahmed, S., Ghosh, K. K., Mirjalili, S., & Sarkar, R. (2021). AIEOU: Automata-based improved equilibrium optimizer with U-shaped transfer function for feature selection. Knowledge-Based Systems, 228, 107283. https://doi.org/10.1016/j.knosys.2021.107283

[9] Bacanin, N., Venkatachalam, K., Bezdan, T., Zivkovic, M., & Abouhawwash, M. (2023). A novel firefly algorithm approach for efficient feature selection with COVID-19 dataset. Microprocessors and Microsystems, 98, 104778. https://doi.org/10.1016/j.micpro.2023.104778

[10] Balochian, S., & Baloochian, H. (2019). Social mimic optimization algorithm and engineering applications. Expert Systems with Applications, 134, 178-191. https://doi.org/10.1016/j.eswa.2019.05.035

[11] Chandrashekar, G., & Sahin, F. (2014). A survey on feature selection methods. Computers & Electrical Engineering, 40(1), 16-28. https://doi.org/10.1016/j.compeleceng.2013.11.024

[12] Chantar, H., Tubishat, M., Essgaer, M., & Mirjalili, S. (2021). Hybrid binary dragonfly algorithm with simulated annealing for feature selection. SN computer science, 2(4), 295. https://doi.org/10.1007/s42979-021-00687-5

[13] Ding, Y., Zhou, K., & Bi, W. (2020). Feature selection based on hybridization of genetic algorithm and competitive swarm optimizer. Soft Computing, 24, 11663-11672. https://doi.org/10.1007/s00500-019-04628-6

[14] Elgamal, Z. M., Yasin, N. B. M., Tubishat, M., Alswaitti, M., & Mirjalili, S. (2020). An improved harris hawks optimization algorithm with simulated annealing for feature selection in the medical field. IEEE access, 8, 186638-186652. https://doi.org/10.1109/ACCESS.2020.3029728

[15] Faramarzi, A., Heidarinejad, M., Stephens, B., & Mirjalili, S. (2020). Equilibrium optimizer: A novel optimization algorithm. Knowledge-Based Systems, 191, 105190. https://doi.org/10.1016/j.knosys.2019.105190

[16] Ghosh, K. K., Guha, R., Bera, S. K., Sarkar, R., & Mirjalili, S. (2020). BEO: Binary equilibrium optimizer combined with simulated annealing for feature selection. Research Square. https://doi.org/10.21203/rs.3.rs-28683/v1

[17] Gao, Y., Zhou, Y., & Luo, Q. (2020). An efficient binary equilibrium optimizer algorithm for feature selection. IEEE Access, 8, 140936-140963. https://doi.org/10.1109/ACCESS.2020.3013617

[18] Hussain, K., Neggaz, N., Zhu, W., & Houssein, E. H. (2021). An efficient hybrid sine-cosine Harris hawks optimization for low and high-dimensional feature selection. Expert Systems with Applications, 176, 114778. https://doi.org/10.1016/j.eswa.2021.114778

[19] Igiri, C. P., Singh, Y., & Poonia, R. C. (2020). A review study of modified swarm intelligence: particle swarm optimization, firefly, bat and gray wolf optimizer algorithms. Recent Advances in Computer Science and Communications (Formerly: Recent Patents on Computer Science), 13(1), 5-12. https://doi.org/10.2174/2213275912666190101120202

[20] Ibrahim, R. A., Ewees, A. A., Oliva, D., Abd Elaziz, M., & Lu, S. (2019). Improved salp swarm algorithm based on particle swarm optimization for feature selection. Journal of Ambient Intelligence and Humanized Computing, 10, 3155-3169. https://doi.org/10.1007/s12652-018-1031-9

[21] Ibrahim, R. A., Abd Elaziz, M., Ewees, A. A., El-Abd, M., & Lu, S. (2021). New feature selection paradigm based on hyper-heuristic technique. Applied Mathematical Modelling, 98, 14-37. https://doi.org/10.1016/j.apm.2021.04.018

[22] Johnson, J. M., & Rahmat-Samii, Y. (1994, June). Genetic algorithm optimization and its application to antenna design. In Proceedings of IEEE Antennas and Propagation Society International Symposium and URSI National Radio Science Meeting (Vol. 1, pp. 326-329). IEEE. https://doi.org/10.1109/APS.1994.407746

[23] Kennedy, J. (2003, April). Bare bones particle swarms. In Proceedings of the 2003 IEEE Swarm Intelligence Symposium. SIS'03 (Cat. No. 03EX706) (pp. 80-87). IEEE. https://doi.org/10.1109/SIS.2003.1202251

[24] Kirkpatrick, S., Gelatt Jr, C. D., & Vecchi, M. P. (1983). Optimization by simulated annealing. science, 220(4598), 671-680. https://doi.org/10.1126/science.220.4598.671

[25] Liu, Y., Zou, X., Ma, S., Avdeev, M., & Shi, S. (2022). Feature selection method reducing correlations among features by embedding domain knowledge. Acta Materialia, 238, 118195. https://doi.org/10.1016/j.actamat.2022.118195

[26] Moslehi, F., & Haeri, A. (2020). A novel hybrid wrapper–filter approach based on genetic algorithm, particle swarm optimization for feature subset selection. Journal of Ambient Intelligence and Humanized Computing, 11, 1105-1127. https://doi.org/10.1007/s12652-019-01364-5

[27] Mafarja, M., Qasem, A., Heidari, A. A., Aljarah, I., Faris, H., & Mirjalili, S. (2020). Efficient hybrid nature-inspired binary optimizers for feature selection. Cognitive Computation, 12, 150-175. https://doi.org/10.1007/s12559-019-09668-6

[28] Nssibi, M., Manita, G., & Korbaa, O. (2023). Advances in nature-inspired metaheuristic optimization for feature selection problem: A comprehensive survey. Computer Science Review, 49, 100559. https://doi.org/10.1016/j.cosrev.2023.100559

[29] Piri, J., Mohapatra, P., Singh, H. K. R., Acharya, B., & Patra, T. K. (2022). An Enhanced Binary Multiobjective Hybrid Filter-Wrapper Chimp Optimization Based Feature Selection Method for COVID-19 Patient Health Prediction. IEEE Access, 10, 100376-100396. https://doi.org/10.1109/ACCESS.2022.3203400

[30] Sharifai, A. G., & Zainol, Z. B. (2021). Multiple filter-based rankers to guide hybrid grasshopper optimization algorithm and simulated annealing for feature selection with high dimensional multi-class imbalanced datasets. IEEE Access, 9, 74127-74142. https://doi.org/10.1109/ACCESS.2021.3081366

[31] Sayed, G. I., Khoriba, G., & Haggag, M. H. (2022). A novel chaotic equilibrium optimizer algorithm with S-shaped and V-shaped transfer functions for feature selection. Journal of Ambient Intelligence and Humanized Computing, 1-26. https://doi.org/10.1007/s12652-021-03151-7

[32] Shambour, M. D. K. Y., Abusnaina, A. A., & Alsalibi, A. I. (2019). Modified global flower pollination algorithm and its application for optimization problems. Interdisciplinary Sciences: Computational Life Sciences, 11, 496-507. https://doi.org/10.1007/s12539-018-0295-2

[33] Tiwari, A., & Chaturvedi, A. (2022). A hybrid feature selection approach based on information theory and dynamic butterfly optimization algorithm

for data classification. Expert Systems with Applications, 196, 116621. https://doi.org/10.1016/j.eswa.2022.116621

[34] Thakkar, A., & Lohiya, R. (2022). A survey on intrusion detection system: feature selection, model, performance measures, application perspective, challenges, and future research directions. Artificial Intelligence Review, 55(1), 453-563. https://doi.org/10.1007/s10462-021-10037-9

[35] Too, J., & Mirjalili, S. (2021). General learning equilibrium optimizer: a new feature selection method for biological data classification. Applied Artificial Intelligence, 35(3), 247-263. https://doi.org/10.1080/08839514.2020.1861407

[36] Too, J., & Abdullah, A. R. (2020). Chaotic atom search optimization for feature selection. Arabian Journal for Science and Engineering, 45(8), 6063-6079. https://doi.org/10.1007/s13369-020-04486-7

[37] Tanveer, M., Rajani, T., Rastogi, R., Shao, Y. H., & Ganaie, M. A. (2022). Comprehensive review on twin support vector machines. Annals of Operations Research, 1-46. https://doi.org/10.1007/s10479-022-04575-w

[38] Thaher, T., Chantar, H., Too, J., Mafarja, M., Turabieh, H., & Houssein, E. H. (2022). Boolean Particle Swarm Optimization with various Evolutionary Population Dynamics approaches for feature selection problems. Expert Systems with Applications, 195, 116550. https://doi.org/10.1016/j.eswa.2022.116550

[39] Unler, A., Murat, A., & Chinnam, R. B. (2011). mr2PSO: A maximum relevance minimum redundancy feature selection method based on swarm intelligence for support vector machine classification. Information Sciences, 181(20), 4625-4641. https://doi.org/10.1016/j.ins.2010.05.037

[40] Vergara, J. R., & Estévez, P. A. (2014). A review of feature selection methods based on mutual information. Neural computing and applications, 24, 175-186. https://doi.org/10.1007/s00521-013-1368-0

[41] Wang, L., Jiang, S., & Jiang, S. (2021). A feature selection method via analysis of relevance, redundancy, and interaction. Expert Systems with Applications, 183, 115365. https://doi.org/10.1016/j.eswa.2021.115365

[42] Zivkovic, M., Stoean, C., Chhabra, A., Budimirovic, N., Petrovic, A., & Bacanin, N. (2022). Novel improved salp swarm algorithm: An application for feature selection. Sensors, 22(5), 1711. https://doi.org/10.3390/s22051711

Mohammad Ansari Shiri
Orcid number: 0009-0002-2273-2424
Department of Computer Science
Shahid Bahonar University of Kerman
Kerman, Iran
Email address: mohammadansari7685@gmail.com

Mohammad Reza Omidi
Orcid number: 0009-0007-1783-5070
Department of Computer Science
Shahid Bahonar University of Kerman
Kerman, Iran
Email address: r_omidi@uk.ac.ir

Najme Mansouri
Orcid number: 0000-0002-1928-5566
Department of Computer Science
Shahid Bahonar University of Kerman
Kerman, Iran
Email address: Najme.mansouri@gmail.com