

Article

An Efficient Heap Based Optimizer Algorithm for Feature Selection

Mona A. S. Ali ^{1,2,*} , Fathimathul Rajeeana P. P. ¹  and Diao Salama Abd Elminaam ^{3,4,*} 

¹ Computer Science Department, College of Computer Science and Information Technology, King Faisal University, Al Ahsa 400, Saudia Arabia; fatimah.rajeena@kfu.edu.sa

² Computer Science Department, Faculty of Computers and Artificial Intelligence, Benha University, Benha 12311, Egypt; mona.abdelbaset@fci.bu.edu.eg

³ Department of Computer Science, Faculty of Computers and Information, Misr International University, Cairo 12585, Egypt; diaa.salama@miuegypt.edu.eg

⁴ Department of Information System, Faculty of Computers and Artificial Intelligence, Benha University, Benha 12311, Egypt; diaa.salama@fci.bu.edu.eg

* Correspondence: m.ali@kfu.edu.sa or mona.abdelbaset@fci.bu.edu.eg (M.A.S.A.); diaa.salama@fci.bu.edu.eg (D.S.A.E.)

Abstract: The heap-based optimizer (HBO) is an innovative meta-heuristic inspired by human social behavior. In this research, binary adaptations of the heap-based optimizer *B_HBO* are presented and used to determine the optimal features for classifications in wrapping form. In addition, HBO balances exploration and exploitation by employing self-adaptive parameters that can adaptively search the solution domain for the optimal solution. In the feature selection domain, the presented algorithms for the binary Heap-based optimizer *B_HBO* are used to find feature subsets that maximize classification performance while lowering the number of selected features. The textitk-nearest neighbor (textitk-NN) classifier ensures that the selected features are significant. The new binary methods are compared to eight common optimization methods recently employed in this field, including Ant Lion Optimization (ALO), Archimedes Optimization Algorithm (AOA), Backtracking Search Algorithm (BSA), Crow Search Algorithm (CSA), Levy flight distribution (LFD), Particle Swarm Optimization (PSO), Slime Mold Algorithm (SMA), and Tree Seed Algorithm (TSA) in terms of fitness, accuracy, precision, sensitivity, F-score, the number of selected features, and statistical tests. Twenty datasets from the UCI repository are evaluated and compared using a set of evaluation indicators. The non-parametric Wilcoxon rank-sum test was used to determine whether the proposed algorithms' results varied statistically significantly from those of the other compared methods. The comparison analysis demonstrates that *B_HBO* is superior or equivalent to the other algorithms used in the literature.

Keywords: dimensionality reduction; feature selection; meta-heuristics; Heap Based Optimizer (HBO)

MSC: 68T01; 68T20



Citation: Ali, M.A.S.; P.P., F.R.; Abd Elminaam, D.S. An Efficient Heap Based Optimizer Algorithm for Feature Selection. *Mathematics* **2022**, *10*, 2396. <https://doi.org/10.3390/math10142396>

Academic Editors: Adrian Deaconu, Petru Adrian Cotfas and Daniel Tudor Cotfas

Received: 19 May 2022

Accepted: 7 July 2022

Published: 8 July 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

On one hand, the massive amounts of data collected in all industries at present provide more specific and valuable information. On the other side, it is more difficult to analyze this data when not all the information is important. Identifying the appropriate aspects of data is a difficult challenge. Dimension reduction is a strategy used to solve classification and regression problems by identifying a subset of characteristics and eliminating duplicate ones. This method is very useful when there are numerous qualities, and not all of them are needed to interpret the data and conduct additional experiments on the attributes. The essential principle of selecting features is that for many pattern classification tasks, a large number of features does not necessarily equal a high classification accuracy. Idealistically,

the selected attributes subset will improve classifier performance and provide a quicker, more cost-effective classification, resulting in comparable or even higher classification accuracy than using all of the attributes [1].

Selecting feature subsets with a powerful and distinctive impact for high-dimensional data analysis is a critical step. High-dimensional datasets have recently become more prevalent in various real-world applications, including genome ventures, data mining, and computer vision. However, the high dimensionality of the datasets may result from unnecessary or redundant features, which can reduce the effectiveness of the learning algorithm or result in data overfitting [2].

Feature selection (FS) has become a viable data preparation method for addressing the curse of dimensionality. FS strategies focus on selecting feature subsets using various selection criteria while keeping the physical meanings of the original characteristics [3]. It can make learning models easier to comprehend and perceive. FS has proven its efficiency in various real-world machine learning and data mining problems, such as pattern recognition, information retrieval, object-based image classification, intrusion detection, and spam detection, to name only a few [4]. The FS process aims to reduce the search space's dimension to improve the learning algorithm's efficiency [5].

The feature selection methodology derives its strength from two main processes, the search and the evaluation. Choosing the most valuable features from the original set with passing all the incoming subsets may face a combinatorial explosion. Therefore, search methodologies are adopted to select the worthy features efficiently. The traditional greedy search strategies such as forward and backward search have been used. The problem with this type of searching is that it may succumb to locally optimal solutions, resulting in non-optimal features. The evaluation function can handle this issue by assessing each feature subset's overall importance, which may help discover the globally optimal or near-optimal solution. Based on the methods used to evaluate feature subsets, the feature selection algorithms are categorized into three primary approaches: filter, wrapper, and embedding methods [6].

Since FS seeks out the near-optimal feature subset, it is considered an optimization problem. Thus, exhaustive search methodologies will be unreliable in this situation, as they generate all potential solutions to find only the best one [7].

Meta-heuristic algorithms gain their superiority from their ability to find the most appropriate solutions in an acceptable, realistic time [8]. In general, meta-heuristic and evolutionary algorithms can avoid the problem of local-optima better than traditional optimization algorithms [9]. Recently, nature-inspired meta-heuristic algorithms have been used most frequently to tackle optimization problems [10].

Typically, the feature selection problem can be mathematically phrased as a multi-objective problem with two objectives: decreasing the size of the selected feature set and optimizing classification accuracy. Typically, these two goals are incompatible, and the ideal answer is a compromise.

Meta-heuristic algorithms are stochastic algorithms that fall into two categories: single-solution-based and population-based. The solution is randomly generated until it reaches the optimum result [11]. In contrast, the population-based algorithm's strategy is based on evolving a set of solutions (i.e., population) in a given search space during many iterations until it obtains the best solution. According to the theory of evolutionary algorithms, population-based algorithms are categorized into physics laws-based algorithms, swarm intelligence of particles, and bio-inspired algorithms' biological behavior. Evolutionary algorithms (EA) are based on the fitness of survival attempts. GA inspires its strategy from natural evolutionary processes (e.g., reproduction, mutation, recombination, and selection). Swarm intelligence (SI) techniques are based on swarms' mutual intelligence. Finally, the physical processes that motivate the physics law-based algorithms include electrostatic induction, gravitational force, and heating systems of materials [11].

Several algorithms proved their efficiency in both optimization and feature selection fields. The Genetic Algorithms (GA), especially binary GA approaches, are regarded as leading evolution-based algorithms that have been used to handle FS problems [12].

The Particle Swarm Optimization (PSO) algorithm, constructed for continuous optimization issues [13], also has a binary version (BPSO) that was presented for binary optimization problems [14]. The BPSO has been used, as well, in FS by [15–17]. Furthermore, many other optimization algorithms succeeded in solving FS problems, such as the Ant Colony Optimization (ACO) algorithm [18], Artificial Bee Colony (ABC) algorithm [19], Binary Gravitational Search Algorithm (BGSa) [20], Scatter Search Algorithm (SSA) [21], Archimedes Optimization Algorithm (AOA) [22], Backtracking Search Algorithm (BSA) [23], Marine Predators Algorithm (MPA) [24], and Whale Optimization Algorithm (WOA) [25].

The common challenge of the previously suggested metaheuristics for FS is the slow convergence rate, bad scalability [26], and lack of precision and consistency. Moreover, the characteristics of large-scale FS issues with various datasets may differ. As a result, solving diverse large-scale FS issues using an existing approach with only one candidate solution generating process may be inefficient [27]. Furthermore, identifying an appropriate FS approach and parameter values takes time to efficiently address a large-scale FS issue. This limitation motivates the current study, which proposes a novel algorithm for an FS task using the Heap Based optimizer.

As a result, In this research, we propose an enhancement to a current optimization technique known as the Heap Based Optimizer (HBO), which is a brand-new human behavior-based algorithm [28]. The HBO is a novel meta-heuristic inspired by corporate rank hierarchy (CRH) and some human behavior. An adaptive opposition strategy is proposed to enable the original algorithm to achieve more precise outcomes with increasingly complex challenges.

HBO displayed incredibly competitive performance. It demonstrates effectiveness in optimization issues. HBO provides numerous benefits, including fewer parameters, a straightforward configuration, simple implementation, and precise calculation. In addition, this method is superior to other algorithms. The HBO algorithm takes fewer iterations. All of these features are really beneficial for resolving the FS issue. It has a straightforward approach, low computational burden, rapid convergence, near-global solution, issue independence, and gradient-free nature [29,30].

This paper reports the following main contributions:

- An improved Heap Based Optimizer (HBO) algorithm, termed *B_HBO*, is proposed for the feature selection problem
- The proposed improved version was tested on 20 datasets, in which 8 belong to a considerably high-dimensional class. The performance of the meta-heuristic algorithms on FS problems for such high-dimensional datasets is rarely investigated.
- The performance of proposed *B_HBO* in terms of fitness, accuracy, precision, sensitivity, F-score, and the number of selected features is compared with some recent optimization methods.

The remainder of the article is as follows: Section 2 demonstrates the process of reviewing the literature on FS metaheuristic algorithms. Section 3 introduces Continuous Heap Based Optimizer steps. The binary HBO strategy is detailed in Section 4. The experiment's results are discussed in Section 5. In Section 6, conclusions and future work are discussed.

2. Related Works

Real-world numerical optimization problems have become increasingly complicated, necessitating efficient optimization approaches. The research direction of evolving and implementing metaheuristics to solve FS problems is still in progress. Generally, identifying prominent feature subsets requires an evaluation that compares subsets and chooses the best. When using a wrapper feature selection model, three performance indicators should

be used: a classifier (e.g., Support Vector Machine, k-Nearest Neighbors, Decision Tree), feature selection criteria (e.g., accuracy, area under the ROC curve, and false-positive (FP) elimination rate), and an optimization algorithm to search for the optimal combination of features [31]. Numerous novel SI optimization algorithms were used as search strategies for several wrapper feature selection techniques. Several studies are interested in converting the continuous optimization methods into binary versions to deal with discrete problems and find the optimal feature subset in classification processes. For instance, different binary versions of PSO have been proposed in [32,33], and [6]. The authors of [32] presented an optimization algorithm based on PSO. This study uses the catfish influence to enhance binary particle swarm optimization (BPSO). The proposed algorithm concept is to introduce new particles into the search area that are initialized at the search space's extreme points, and then substitute particles with the worst fitness if the global best particle's fitness does not improve after several iterations. The authors used the K-NN technique with leave-one-out cross-validation (LOOCV) to determine the efficacy of the proposed algorithm's output.

The authors of [33] presented a binary PSO algorithm with a mutation operator (MBPSO) to address the spam detection problem and minimize false positives caused by mislabeling non-spam as spam.

Another binary optimization algorithm based on Grasshopper Optimization was developed in [34]. The authors presented two main methods to produce BGOA. The first method employs Sigmoid and V-shaped transfer functions to convert continuous solutions to binary equivalents. In the second method, a novel strategy, BGOA-M repositions the current solution based on the best solution's location up to this point. In the same context, the authors of [35] transformed the continuous version of gray wolf optimization to a binary form (bGWO) to be used in the classification process. Based on the wrapper model, they created two approaches for obtaining the optimal features. The first approach tended to pick the best three solutions, binarize them, and then apply a random crossover. The second approach uses a sigmoid function to smash the continuous position, which is then stochastically thresholded to obtain the modified binary gray wolf position. The results of the proposed approaches outperformed GA and PSO. Another discrete version of GWO, the multi-objective gray wolf optimization algorithm, was presented in [36]. This proposed algorithm's goal was to reduce the dimensions of the features for multiclass sentiment classification.

One more neoteric nature-inspired optimization strategy is the Whale optimization algorithm (WOA). WOA is a strategy that emulates the behavioral patterns of a bubble-net hunter. As a binary model based on a sigmoid transfer function, it was adopted in [10] for feature selection and classification problems (S-shape). BWOA works on the principle of converting any free position of the whale to its binary solutions. This proposed approach forces the search agents to travel in a binary space by implementing an S-shaped transfer function for each dimension.

Likewise, the Ant Lion Optimizer (ALO) algorithm was converted to a binary version by [35]. This binary version was developed using two approaches. The first just converted the ALO operators to their corresponding binary. The authors used the original ALO and the proper threshold function to threshold its continuous steps after squelching them in the second approach. According to this study, the BALO outperformed the original ALO and other popular optimization algorithms such as GA, PSO, and binary bat algorithm. The authors of [37] introduced a new version of the Salp Swarm Optimizer (SSA) for feature selection. The SSA's position was updated in the proposed methodology by a type of Sine Cosine Algorithm called sinusoidal mathematical function. This study aimed to improve the exploration process and avoid slumps in a local area.

Furthermore, it tried to enhance the population variety and keep a balance between exploration and exploitation processes using the Disruption Operator. SSA was also used to develop two wrapper-based feature selection models in [2]. The first model used eight transfer functions to transform the continuous version of SSA. The second model added the crossover operator to improve the swarm behavior with these transfer functions.

Other optimization algorithms, such as the Binary Spotted Hyena Optimizer (BSHO) [38], Binary bat algorithm (BBA) [39], Binary dragonfly algorithm [40], Henry gas solubility optimization algorithm (HGSO) [41], Gradient-based optimizer (GBO) [42], Crow search algorithm (CSA) [43,44], Equilibrium Optimizer (EO) [45], and chaotic cuckoo algorithm [46] are algorithms that have been used to solve feature selection problems and improve classification performance.

One of the recent directions in the literature is using hybrid algorithms to improve performance and efficiency. The authors of [47] introduced a hybrid algorithm that merges the Flower Pollination Algorithm (FPA) and the Clonal Selection Algorithm to produce a Binary Clonal Flower Pollination Algorithm (BCFA). Their proposed algorithm used the Optimum-Path Forest classifier as an objective function. BCFA was tested on three benchmark datasets and achieved stunning results in selecting the optimal features in less time. A hybrid feature selection algorithm based on the forest optimization algorithm (FOA) and minimization of redundancy and maximum relevance (mRMR) was proposed in [48]. The results showed that applying K-NN and NB classifiers with the proposed FOA algorithm outperformed standard classifier algorithms. The authors of [49] presented a binary swallow swarm optimization (BSSO) algorithm with a fuzzy rule-based classifier to solve the feature selection problem for fuzzy classification. This proposed approach focused on eliminating more potentially noise-inducing features. However, as mentioned in this study, the main drawback of BSSO is the significant number of tunable parameters that must be empirically chosen.

The authors of [50] combined the GA and PSO to benefit from the exploitation capabilities of GA and the exploration capabilities of PSO. The proposed binary genetic swarm optimization (BGSO) allows GA and PSO to operate independently. Furthermore, BGSO integrated its outcome with the average weighted combination method to generate an intermediate solution to elicit sufficient information from the obtained features. The classifiers used to obtain the final results were K-NN and MLP.

According to the various mentioned studies, binary versions of optimization algorithms have strengths over traditional feature selection algorithms.

Some other recent notable works of interest which proposed techniques for feature selection based on fuzzy entropy such as in [51,52,52,53]. Other notable works of interest that proposed metaheuristics for the FS problem are discussed below.

The authors of [54] proposed an improved multi-objective Salp Swarm Algorithm (SSA) that incorporates a dynamic time-varying strategy and local fittest solutions. The SSA algorithm relies on these components to balance exploration and exploitation. As a result, it achieves faster convergence while avoiding locally optimum solutions. The method is tested on 13 datasets (Nci9, Glioma, Lymphography, PenglungEW, WaveformEW, Zoo, Exactly, Exactly2, HeartEW, SonarEW, SpectEW, Colon, and Leukemia). The authors of [55] combined monarch butterfly optimizer (MBO) and a KNN classifier wrapper-based FS method. The simulations were made on 18 standard datasets. The suggested method resulted from high classification accuracy of 93% as compared to the related other methods. The authors of [56] gave two binary versions of the Butterfly Optimization Algorithm (BOA), which were applied to pick the optimal subset of features for classification with the help of a wrapper mode. The methods were assessed on 21 datasets of the UCI repository. The authors of [57] adopted a binary equilibrium optimizer (EO) motivated by controlled volume mass balance models for determining the dynamic and equilibrium states with a V-shaped transfer function (BEO-V) to choose the optimal subset of features for classification. The anticipated method is tested on 19 UCI datasets to gauge the performance. The authors of [58] proposed a Social Mimic Optimization (SMO) algorithm based on people's behavior in society, with the X-shaped transfer function based on crossover operation, to augment the exploration and exploitation credibility of binary SMO. The anticipated method is analyzed using 18 benchmark UCI datasets for performance evaluation. Sigmoid transfer function maps continuous search space into binary space. The authors of [59] suggested a binary version of Sailfish Optimizer (SFO), called Binary Sailfish (BSF), to deal with FS problems.

To promote the exploitation credibility of the BSF algorithm, they combined it with adaptive β -hill climbing. The algorithms are evaluated on 18 UCI benchmark datasets, which were then compared with ten other meta-heuristics applied for FS methods. The authors of [60] blended GA with the Great Deluge Algorithm (GDA) used in place of mutation operation, thus achieving a high degree of exploitation through perturbation of candidate solutions. The proposed method was evaluated on 15 openly available UCI datasets. The authors of [61] proffered using a binary Dragonfly Algorithm (BDA) with an improved updating mechanism to capitalize on the survival-of-the-fittest basis by investigating some functions and strategies to revise its five main coefficients for the FS problems. The methods were tested and analyzed with 18 standard datasets, with Sinusoidal-BDA obtaining the best results. The authors of [62] forwarded a binary symbiotic organism search (SOS) that was used to map continuous-space to a discrete space with the help of an adaptive S-shaped transfer function to obtain the optimal subset of features. The method is assessed on 19 datasets of UCI. The authors of [63] suggested a multi-objective FS method based on forest optimization algorithm (FOA) with archive, grid, and region-based selection. The performance is evaluated on nine UCI datasets and two microarray datasets. The authors of [64] proposed an improved Coral Reefs Optimization (CRO) for finding the best subsets of features. The method applied tournament selection to improvize the initial population's diversity. The method outperforms other algorithms.

To summarize, FS is a binary optimization problem that aims to improve the classification accuracy of machine learning algorithms using a smaller subset of features. To solve the FS problem, many meta-heuristic algorithms were proposed to explore the solution space to determine the optimal or near-optimal solution by belittling the fitness function. However, deciding a specific subset of features through a meta-heuristic algorithm involves a transfer function that can convert the continuous search space to a binary one. Most of the standard datasets from the UCI repository were for evaluation. The literature review results are summarized in Table 1.

Table 1. Summary of related works for feature Selection.

Reference	Optimization Method	Dataset
[6]	V-shaped binary PSO	10 dataset
[40]	Binary dragonfly algorithm	18 dataset
[2]	SSA with crossover operator	22 dataset
[34]	BGOA and BGOAM	25 dataset
[31]	Binary coordinate ascent based FSS algorithm	12 dataset
[36]	multi-objective GWO	3 dataset
[44]	Binary Crow Search Algorithm with Time Varying Flight Length	20 dataset
[55]	MBO with KNN wrapper	18 UCI datasets
[65]	MBA-SA	18 UCI datasets
[66]	SHO with chaotic maps	21 UCI dataset
[67]	CSO with chaos maps and fuzzy c-means function	11 medical datasets
[56]	BOA	21 UCI datasets
[59]	Binary SFO with adaptive β -hill climbing	18 UCI datasets
[61]	BDA with sinusoidal updating function	
[62]	binary SOS with adaptive S-shaped transfer function	19 UCI datasets
[68]	BBHO with chaotic maps	Chalcone, Hepatitis, H1N1

The common challenge of the previously suggested metaheuristics for FS is the slow convergence rate, bad scalability [26], and lack of precision and consistency. It takes time to identify an appropriate FS approach and parameter values efficiently.

This limitation motivates the current study, which proposes a novel algorithm for an FS task using the Heap Based optimizer. The HBO was selected as an effective optimization engine for a wrapper FS approach because, relative to Si-based optimizers, it showed sufficient efficacy in tackling several optimization difficulties. The HBO is a new optimizer that has not yet been used to solve FS issues. Its distinctive properties make it a suitable search engine for global optimization and FS issues. The HBO is initially efficient, adaptable, simple, and straightforward to deploy. To balance exploration and exploitation, HBO has

only one parameter. This parameter is decreased adaptively across successive iterations, enabling the HBO to explore the majority of the search space at the beginning of the searching process and then exploit the promising regions in the final phases, preventing the HBO from becoming trapped at local optima. Therefore, we used an improved and Enhanced algorithm of HBO in this study to select the most optimal features. We proposed a new dimension reduction approach, the bHBO-based one, based on the k-NN classifier, for selecting the optimal set of features.

3. Procedure and Methodology

The proposed framework for the Binary Heap-Based optimizer B_HBO for Feature Selection contains three significant steps illustrated in Figure 1. The Heap Based Optimizer (HBO) algorithm is the most recent and advanced SI algorithm. HBO was proposed in 2020 by Qamar Askari, Mehreen Saeed, and Irfan Younas [28], and the competition was fierce. It exhibits effectiveness when it comes to tackling optimization difficulties. HBO provides a variety of advantages, including fewer parameters, simple configuration, ease of implementation, and high calculation accuracy. Furthermore, this method surpasses competing algorithms in terms of performance. It retains optimization findings, showing that marine predators benefit from a strong memory for recalling both their associates and the location of successful foraging. Furthermore, the HBO algorithm requires fewer iterations. It has a simple technique, a low computational cost, rapid convergence, a near-global solution, problem independence, and a gradient-free nature [28]. All of these benefits are critical in resolving the FS issue. HBO was ranked second and displayed exceptionally competitive performance compared to LSHADE-cnEpSin, the highest performing technique and a CEC 2017 winner. HBO can be called a high-performance optimizer because it statistically outperforms GA, PSO, GSA, CS, and SSA.

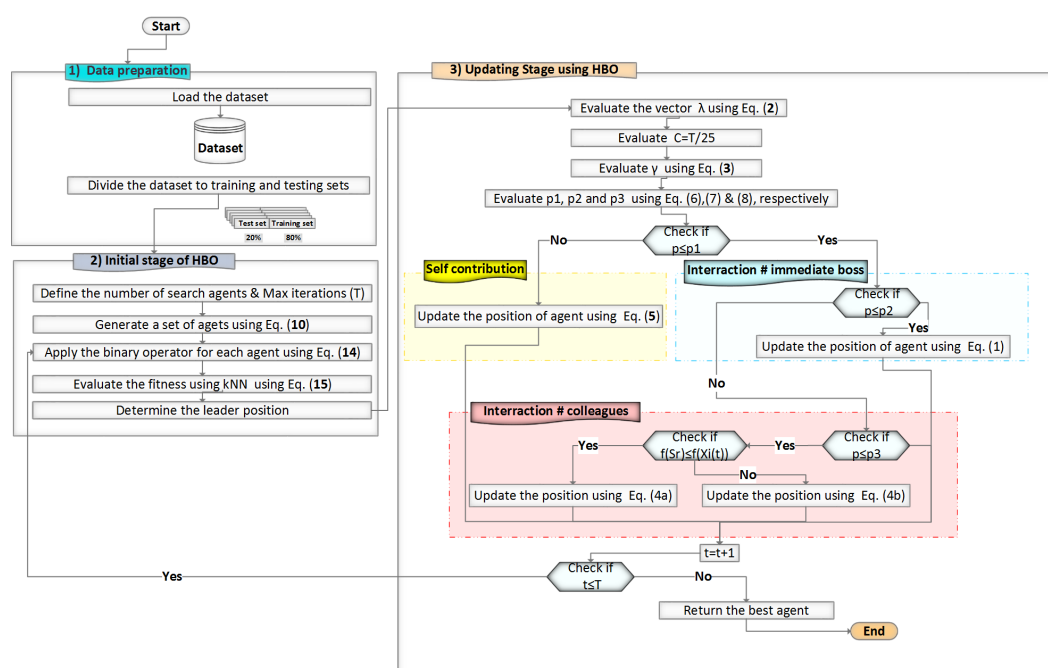


Figure 1. The framework of the proposed B_HBO for feature selection based on KNN classifier.

3.1. Continuous Heap Based Optimizer

This section describes the steps of the heap-based optimizer algorithm (HBO). The HBO imitates the job titles, duties, and job descriptions of employees [28]. Although the designations differ from company to company and business to business, they are all structured hierarchically. Many know them by titles, including corporate rank hierarchy (CRH), organizational chart tree, and corporate hierarchy structure [69]. The organizational structure is a set of strategies for dividing tasks into specific responsibilities and coordinating

them. The main body of HBO is presented in Algorithm 1. The mathematical model of the heap-based optimizer is discussed in this section.

Algorithm 1 HBO Pseudo-code

```

for ( $i \leftarrow 1$  to  $T$ ) do
   $\gamma$  is calculated using Equation (3).
  ( $p_1$ ) is calculated using Equation (6)
  ( $p_2$ ) is calculated using Equation (7)
  for ( $i \leftarrow N$  down to 2) do
     $i \leftarrow \text{heap}[I].\text{value}$ 
     $b_i \leftarrow \text{heap}[\text{parent}(I)].\text{value}$ 
     $c_i \leftarrow \text{heap}[\text{colleague}(I)].\text{value}$ 
     $\vec{B} \leftarrow \vec{x}_{bi}$ 
     $\vec{S} \leftarrow \vec{x}_{ci}$ 
    for ( $k \leftarrow 1$  to  $D$ ) do
       $p \leftarrow \text{rand}()$ 
       $x_{temp}^k \leftarrow \text{update}x_i^k(t)$  with Equation (9)
    end for
    if  $f(\vec{x}_{temp}) < f(\vec{x}_i(t))$  then
       $\vec{x}_i(t+1) \leftarrow \vec{x}_i(t)$ 
    end if
    Heapify_Up( $I$ )
  end for
end for
return  $x_{\text{heap}}[1].\text{value}$ 

```

3.1.1. Mathematical Formulating of the Collaboration with the Direct Boss

In a centralized organizational structure, upper-level policies and norms are imposed, and subordinates report to their immediate superior. This behavior may be simulated by changing each search agent's position \vec{x}_i regarding its parent node B as shown in Equation (1)

$$X_i^k(t+1) = B^k + \gamma \lambda^k |B^k - X_i^k(t)| \quad (1)$$

where t represents the current iteration and k represents the k th component of the vector, and $||$ calculates the absolute value. λ^k represents the k th component of the vector $\vec{\lambda}$, which is generated randomly as demonstrated by Equation (2)

$$\vec{\lambda} = 2r - 1 \quad (2)$$

where r is an integer generated randomly in between range $[0, 1]$ using the uniform distribution. γ is a well chosen parameter in Equation (1), and it is computed as as shown in Equation (3)

$$\gamma = \left| 2 - \frac{(t \bmod \frac{T}{C})}{\frac{T}{4C}} \right| \quad (3)$$

where T is the total number of iterations, and C is a user-defined parameter, as described below. γ decreases linearly from 2 to 0 throughout iterations, and after reaching 0, it begins to rise again to 2 with more iterations. However, C specifies the number of cycles γ will complete in T iterations.

To determine the effect of the parameter C on the performance of HBO, we solved a variety of unimodal and multimodal benchmark functions while varying C from its minimum to its maximum value. After doing this experiment for many other functions, we chose to determine the balanced value of C by dividing the maximum number of Iterations by 25.

3.1.2. Mathematical Formulating of the Collaboration between Colleagues

Similar-ranking officials are referred to as “colleagues”. They collaborate to execute official responsibilities. We assume that nodes on the same level are colleagues in a heap, and that each search agent \vec{x}_i modifies its position relative to a randomly chosen colleague \vec{S}_r using (4)

The main objective function

$$X_i^k(t+1) = \begin{cases} S_r^k + \gamma \lambda^k |S_r^k - x_i^k(t)|, & f(\vec{S}_r) < f(\vec{x}_i(t)) \\ x_i^k + \gamma \lambda^k |S_r^k - x_i^k(t)|, & f(\vec{S}_r) \geq f(\vec{x}_i(t)) \end{cases} \quad (4)$$

where f is the fitness-calculating objective function for the search agent. Equation (4) position updating process is fairly similar to Equation (1). In contrast, Equation (4) permits the search agent to explore the region surrounding S_r^k if $(\vec{S}_r) < f(\vec{x}_i(t))$ and the region surrounding x_i^k otherwise.

3.1.3. Self Contribution of an Employee

This phase’s mapping procedure is relatively straightforward. This phase depicts the concept of an employee’s self contribution. The behavior is modeled by keeping the employee’s previous position in the next iteration, as shown in Equation (5)

$$x_i^k(t+1) = x_i^k(t) \quad (5)$$

The search agent \vec{x}_i in Equation (5) does not modify its position for its k th design variable in the next iteration. This behavior is used to control a search agent’s rate of change.

3.1.4. Putting It All Together

This part explains how the previous subsections’ position updating equations are combined into a single equation. Calculating the probabilities of selection for all three equations is a big task, as probabilities play an important part in balancing exploration and exploitation. A roulette wheel, which is divided into three proportions of p_1 , p_2 , and p_3 , is designed to balance them. Using Equation (6), a search agent can update its location by selecting the proportion p_1 . The p_1 limit is calculated as follows:

$$p_1 = 1 - \frac{t}{T} \quad (6)$$

where t stands for the current and T stands for the total number of iterations. Using Equation (7), a search agent can update its position by selecting proportion p_2 . The p_2 limit is calculated as follows:

$$p_2 = p_1 + \frac{1 - p_1}{2} \quad (7)$$

Finally, p_3 is chosen to represent updated position using Equation (8), and the limit of p_3 is calculated as follows:

$$p_3 = p_2 + \frac{1 - p_1}{2} = 1 \quad (8)$$

The following equation depicts HBO’s general position update mechanism:

$$x_i^k(t+1) = \begin{cases} x_i^k(t), & p \leq p_1 \\ B^k + \gamma \lambda^k |B^k - x_i^k(t)|, & p > p_1 \text{ and } p \leq p_2 \\ S_r^k + \gamma \lambda^k |S_r^k - x_i^k(t)|, & p > p_2 \text{ and } p \leq p_3 \text{ and } f(\vec{S}_r) < f(\vec{x}_i(t)) \\ x_i^k + \gamma \lambda^k |S_r^k - x_i^k(t)|, & p > p_2 \text{ and } p \leq p_3 \text{ and } f(\vec{S}_r) \geq f(\vec{x}_i(t)) \end{cases} \quad (9)$$

where p is a number within the range $[0, 1]$ chosen at random. It is worthwhile to note that Equation (5) supports exploration, while Equation (1) supports exploitation and convergence, while Equation (4) supports exploration as well as exploitation. Based on these findings, p_1 is initially increased and subsequently linearly decreased over repetitions,

decreasing exploration and increasing exploitation. After the computation of p_1 , the remaining span is split into two equal parts, increasing the chances of attraction to the boss and coworkers equally likely.

3.1.5. The HBO Step by Step

This section describes the HBO phases and algorithm in detail.

- Initialize generic parameters such as the size of the population (N), the number of design variables/dimensions (D), the maximum iteration (T), and the ranges of the model parameters $C = \lfloor T/25 \rfloor$ are used to compute the algorithm-specific parameter C .
- Create the first population: Create a P by chance population of N -dimensional search agents with D . Following is a representation of population P :

$$p = \begin{bmatrix} \vec{x}_1^T \\ \vec{x}_2^T \\ \vdots \\ \vec{x}_N^T \end{bmatrix} = \begin{bmatrix} x_1^1 & x_1^2 & x_1^3 & x_1^D \\ x_2^1 & x_2^2 & x_2^3 & x_2^D \\ \vdots & \vdots & \vdots & \vdots \\ x_N^1 & x_N^2 & x_N^3 & x_N^D \end{bmatrix} \quad (10)$$

A heap is typically represented by a d_{ary} tree. To implement CRH, however, the 3_{ary} heap is used. Due to its completeness, a heap, which is a tree-shaped data structure, can be built quickly using an array. The following are essential d_{ary} heap-based operations that HBO requires:

1. The index of a node is received, and the index of the node's parent is returned by this function, assuming the heap is implemented as an array. For example, the following is the formula for calculating the index of the parent of node (i):

$$parent(i) = \left\lfloor \frac{i+1}{d} \right\rfloor \quad (11)$$

$\lfloor \cdot \rfloor$ is the floor operator, which produces the largest integer less than or equal to the input.

2. ($i; j$): This method returns the index of the j th child of the given node. A node in a 3_{ary} heap can have no more than three offspring. According to our concept, a leader may have no more than three direct reports. The following is the mathematical formula for this function in constant time:

$$child(i, j) = d \times i - d + j + 1 \quad (12)$$

3. depth (i): Using the following formula, the depth of any node i may be determined in constant time if the depth of the previous level equals 0.

$$depth(i) = \lceil \log(d \times i - i + 1) \rceil - 1 \quad (13)$$

$\lceil \cdot \rceil$ is the notation for the ceil function, which returns the smallest integer greater than or equal to the input.

4. colleague (i): At the node level i , all nodes are considered their colleagues. This function returns the index of colleague of node i chosen randomly, which may be determined by producing random number in the range $\frac{dd^{depth(i)-1}-1}{d-1} + 1, f \frac{dd^{depth(i)-1}-1}{d-1}$.
5. _Up (i) Heapify Up: To maintain the heap property, it searches upward in a heap and enters the node i in its proper spot. Algorithm 2 contains the pseudo code for this operation.

Algorithm 2 Heapify_Up (i) Pseudo-code

```

Inputs:  $i$  (The index of the node that is being heaped.)
        ▷ Considering the remaining nodes satisfy the heap property
while  $i \neq \text{root}$  and  $\text{heap}[i].\text{key}[]\text{parent}(i).\text{key}$  do
    Swap ( $\text{heap}[i]$ ,  $\text{heap}[\text{parent}(i)]$ )
     $i \leftarrow \text{parent}(i)$ 
end while

```

Finally, Algorithm 3 describes the heap-building algorithm.

Algorithm 3 Build_Heap (P, N) Pseudo-code

Inputs: N is the population size , P is the search agents population
for ($i \leftarrow 1$ to N) **do**
 $\text{heap}[i].\text{value} \leftarrow i$
 $\text{heap}[i].\text{key} \leftarrow f(x_i)$
 Heapify_Up (i)
end for

6. Position updating mechanism: Search agents update their positions regularly according to the previously described equations to converge on the best global solution.

4. The Proposed Binary HBO (B_HBO) for Feature Selection

This section includes the proposed Heap Based Optimizer’s (HBO) steps for solving feature selection using KNN as the classifier. The proposed technique is a mix of the *B_HBO* and KNN algorithms for classification, feature selection, and parameter optimization. In *B_HBO*, KNN parameters are used to identify the best selection accuracy, and the selected features are used for all cross-validation folds. Figure 1 depicts the suggested *B_HBO*-KNN approach’s flowchart, which depicts the three steps of the proposed method. Algorithm 4 shows the pseudocode for the proposed *B_HBO* with KNN classification algorithm.

Algorithm 4 The Pseudo code of the proposed *B_HBO* based on KNN classifier.

Inputs: The size of the population, N , and the maximum number of generations T , group classifier G , characteristic X , dataset set D , and a novel fitness function (f_{obj}).

Outputs: The prediction accuracy for each iteration (optimal location) and the highest accuracy value

Randomly Initiate the population X_i ($i = 1, 2, \dots, N$)

while *thestopconditionisnotmet* **do**

 New fitness function computed Using the strategy for selecting call features, Call K-NN classifier

for ($i \leftarrow 1$ to T) **do**

γ is calculated using Equation(3).

 (p_1) is calculated using Equation (6)

 (p_2) is calculated using Equation (7)

for ($i \leftarrow N$ down to 2) **do**

$i \leftarrow \text{heap}[I].\text{value}$

$b_i \leftarrow \text{heap}[\text{parent}(I)].\text{value}$

$c_i \leftarrow \text{heap}[\text{colleague}(I)].\text{value}$

$\vec{B} \leftarrow \vec{x}_{bi}$

$\vec{S} \leftarrow \vec{x}_{ci}$

for ($k \leftarrow 1$ to D) **do**

 Calculation of a new fitness function using the call feature selection technique.

 Call K-NN classifier

$p \leftarrow \text{rand}()$

$x_{temp}^k \leftarrow \text{update}x_i^k(t)$ using Equation (9)

end for

if $f(\vec{x}_{temp}) < f(\vec{x}_i(t))$ **then**

$\vec{x}_i(t+1) \leftarrow \vec{x}_i(t)$

end if

 Heapify_Up(I)

end for

end for

 return $x_{heap}[1].\text{value}$

end while

4.1. FS for Classification

Classification is the most important problem in data mining, and its fundamental function is to estimate the class of an unknown object. A dataset (also known as a training set) typically consists of rows (referred to as objects) and columns (referred to as features) that correspond to predetermined classifications (decision features). A significant number of redundant or irrelevant characteristics in the dataset may be the primary factor affecting a classifier's accuracy and performance. Redundant characteristics may negatively impact the classifier's performance in various ways; adding more features to a dataset involves adding more examples, which increases the learning time of the classifier. Moreover, a classification that learns from features in the dataset is more accurate than one that learns from irrelevant data. This is because irrelevant features can confuse the classifier, overfitting the data. In addition, the duplicated and irrelevant input will increase the complexity of the classifier, making it more challenging to comprehend the learned results. As was demonstrated previously, the selection of a suitable searching strategy in FS techniques is crucial for optimizing the efficiency of the learning algorithm. FS often aids in detecting redundant and unneeded features and eliminating them to enhance the classifier's results in terms of learning time and accuracy, as well as simplifying the findings to make them more understandable. By selecting the most informative feature and removing unneeded and redundant features, the dimension of the feature space is decreased, and the convergence rate of the learning algorithm is accelerated.

Because of the above, the HBO was selected as an efficient optimization engine in a wrapper FS method since it has shown sufficient efficacy in solving several optimization issues compared to Si-based optimization techniques. The HBO is a new optimizer that has not yet been used to solve FS issues. Its distinctive properties make it a suitable search engine for global optimization and FS issues. The HBO is initially efficient, adaptable, simple, and straightforward to deploy. To balance exploration and exploitation, HBO has only one parameter.

4.2. The Proposed Binary HBO (B_HBO)

Searching for the optimal feature subset in FS is a difficult problem, particularly for wrapper-based approaches. This is because the supervised learning (e.g., classifier) must evaluate the selected subset at each optimization step. Consequently, a suitable optimization approach is crucial to minimize the number of evaluations.

The based on comparative of HBO prompted us to suggest using this method as a search strategy in a wrapper-based FS procedure. We proposed a binary version of the HBO to solve the FS problem because the search space may be represented by binary values [0, 1]. Binary operators are believed to be considerably easier than their continuous counterparts. In the continuous form of HBO, each agent's location is updated depending on its current position, the position of the best solution so far (target), and the positions of all other solutions, as shown in Equations (1) and (5). The new solution derived from Equation (4) is obtained by adding the step vector to the target vector (position vector). However, the addition operator cannot be used in a binary space because the position vector only includes 0 s and 1 s. The next three subsections elaborate on these approaches.

4.3. B_HBO Proposed for FS Based on KNN

Previous sections demonstrated the significance of an effective search strategy for FS approaches. Another feature of FS techniques is evaluating the selected subset's quality. Since the suggested method is wrapper-based, an algorithm for learning (such as a classifier) must be incorporated into the evaluation process. This study employs the k-Nearest Neighbor. The classifying quality of the chosen features is integrated into the proposed fitness values because the primary problem of this study is the feature selection problem—not the classification problem—which is tackled by the HBO technique. Each algorithm is executed 51 times with 1000 iterations; the classification is used to select the iteration with the highest accuracy for each run. Therefore, we require the simplicity classifier to reduce each method's complexity and execution time.

In this proposed approach, the KNN is employed as a classification to guarantee the selected features' quality. When relevant features are selected from a subset, the classification accuracy will be enhanced. One of the primary goals of FS approaches is to improve classification accuracy; another is to reduce the number of selected features. The superiority of a solution increases as its number of components decreases. In the proposed fitness function, these two contradicting goals are considered. Equation (16) depicts the fitness function that considers classification accuracy and the number of selected features when evaluating a subset of characteristics across all techniques. In HBO, KNN parameters are used to identify the best selection accuracy, and the selected features are used for all cross-validation folds. Figure 1 depicts the suggested HBO-KNN approach's flowchart, which depicts the three steps of the proposed method. The first step is preprocessing, followed by FS and optimization phase, and then the classification and cross-validation phase. Algorithm 4 shows the pseudocode for the proposed B_HBO based on the KNN classifier.

4.4. Fitness Function for FS

To define FS as an optimization problem, it is necessary to examine two crucial factors: how to represent a solution and how to evaluate it. A wrapper FS strategy employing HBO as a search algorithm and a k-NN classifier as an evaluator has been developed. A feature subset is converted into a binary vector with the same length as the

number of selected features used for this investigation. If the value of a feature is 1, it has been selected; otherwise, it has not. The quality of a feature subset is determined by the classification accuracy (error rate) and the number of features selected simultaneously. These two contradicting objectives are represented by a single fitness function denoted by Equation (14).

$$\text{Fitness} = \alpha \gamma_R(D) + \beta \frac{|R|}{|C|} \quad (14)$$

where $|R|$ is the number of selected features in a reduct, $|C|$ is the number of conditional features in the original dataset, and $\alpha \in [0, 1], \beta = (1 - \alpha)$ are two main parameters related to the significance of classifying performance and subset length.

The proposed fitness function governs the Accuracy of the selected features. During the iterative process, the solutions HBO finds must be reviewed to verify the performance of each iteration. Before the evaluation of fitness, a binary conversion is realized using Equation (15) and the HBO fitness function is defined by Equation (16).

$$x_i^{\text{bin}} = \begin{cases} 1 & \text{if } x_i^t > 0.5 \\ 0 & \text{otherwise.} \end{cases} \quad (15)$$

$$\text{Fit} = 0.99 \times R + 0.01 \times \frac{|c|}{C} \quad (16)$$

R is the classification error rate computed by k -NN (80% for training and 20% for testing), where C denotes the total number of features and c indicates the relevant selected features.

As illustrated in Figure 1, HBO is customized to choose the most important and best features.

5. Results and Discussion

In this section, a comparison between the results of the developed FS approach and other methods is performed. The proposed B_HBO algorithm is compared with eight recent evolutionary feature selection algorithms, such as ALO, AOA, BSA, CSA, LFD, PSO, SMA, and TSA. Each compared algorithm was run 51 times on a population size set to 30 with 1000 iterations. The suggested B_HBO algorithm was constructed in Matlab using the same interactive environment, which was executed on a computer with an Intel(R) Core i7 2.80 GHz processor and 32 GB RAM.

The experiment is achieved using different datasets with different characteristics. The details of the behavior of datasets are given in the following section.

5.1. Datasets and Parameter Setup

Twenty datasets in Table 2 from the UCI machine learning repository [70] are used in the experiments to evaluate the effectiveness of the suggested method. Each dataset's instances are randomly partitioned into 80% for training and 20% for testing. The provided datasets are arranged from the category of low dimensionality to high dimensionality data. Low-dimensional datasets have less than ten feature sizes, whereas high dimension is greater than ten features. The challenge is finding an optimal subset of features with high accuracy to justify the quality performance. This study employs a wrapper-based method for feature selection based on the KNN classifier, where $K = 5$ was determined to be the optimal choice for all datasets. Table 3 presents the settings of parameters of algorithms considered in this work to analyze and assess the performance of the proposed method.

Table 2. Details of Used Datasets.

No.	Dataset	Nunmer of Features	Number of Instances	Number of Classes
1	Arrhythmia	279	452	5
2	Breastcancer	9	699	2
3	BreastEW	30	569	2
4	Congress	16	435	2
5	Diabetes	8	768	2
6	German	24	1000	2
7	Glass	10	214	7
8	Heart-C	14	303	5
9	Heart-StatLog	13	270	2
10	Hepatitis	19	155	2
11	Hillvalley	101	606	2
12	Ionosphere	34	351	2
13	Iris	4	150	3
14	Lung-Cancer	56	32	3
15	Lymphography	18	148	2
16	Vowel	10	990	11
17	WaveformEW	40	5000	3
18	WDBC	32	596	2
19	Wine	13	178	3
20	zoo	16	101	6

Table 3. Parameters settings of B_HBO and other computational algorithms.

Algorithms	Parameters Setting
Common settings	Maximum number of iterations : ($It_{Max} = 1000$) Number of independent runs 51 Population size: $N = 30$
ALO	$r = rand$, r is ranged in $[0, 1]$, Migration coefficient $= 2n$
AOA	$c_1 = 2$ and $c_2 = 6$ $\alpha = 0.9$, $\beta = 0.1$
BSA	$\beta = 1.5$ (Default) Mix Rate =1 $rnd = [0 - 1]$
AO	$U = 0.00565$; $r1 = 10$; $\omega = 0.005$; $\alpha = 0.1$; $\delta = 0.1$; $G1 \in [-1, 1]$; $G2 = [2, 0]$
PSO	$W_{max} = 0.9$ and $W_{min} = 0.2$ $c_1 = 2$ and $c_2 = 2$ Population size = 50
CSA	$A = 2$
SMA	$z = 0.3$, $r = [0, 1]$, $b = [0, 1]$
TSA	$P_{min} = 1$, $p_{max} = 4$
HBO	$sv = 100$, $degree = 3$

5.2. Performance Metrics

It is imperative to quantify the relevant performance metrics which can guide to analysis of the performance behavior of an anticipated algorithm. As a result, the following evaluation metrics and measures were computed for the proposed method (*B_HBO*), developed to solve the feature selection problem [71].

- **Average fitness value:** is the best fitness value Fit_{val} obtained when running several algorithms for N times. It represents decreasing the selection ratio and minimizing the classification error rate. It is calculated by Equation (17):

$$Mean = \frac{\sum_{i=1}^N Fit_{val}}{N} \quad (17)$$

- **Standard Deviation (*StdDev*):** It is an indicator of the stability of the used algorithm. It is calculated by (18):

$$StdDev = \sqrt{\frac{1}{N-1} \sum_{i=1}^N (fit_{val} - mean)^2} \quad (18)$$

- **Average accuracy AVG_{ACC} :** The accuracy metric (ACC) identifies the correct data classification rate. It is calculated by (19):

$$ACC = \frac{TP + TN}{TP + FN + FP + TN} \quad (19)$$

In our study, nine different algorithms are running N times, so it is more suitable to use the AVG_{ACC} metric, which is calculated by (20):

$$AVG_{ACC} = \frac{1}{N} \sum_{i=1}^N ACC \quad (20)$$

- **Sensitivity or True Positive Rate (TPR):** it presents the rate of predicting positive patterns. It is calculated by (21):

$$TPR = \frac{TP}{TP + FN} \quad (21)$$

- **Specificity or True Negative Rate (TNR):** it indicates the percentage of actual negatives which are correctly detected. Equation (22) is used to calculate it:

$$TNR = \frac{TN}{FP + TN} \quad (22)$$

5.3. Comparison of B_HBO with Other Metaheuristics

In this section, the comparison of performance of B_HBO with other well-known meta-heuristic algorithms is performed. The results are discussed in terms of different performance analyses such as:

- **In terms of fitness:** The comparison results between the suggested B_HBO and other competing algorithms are shown in Table 4. It is evident from the obtained results that our B_HBO provides results better than the others. For example, it has the smallest results compared with the competitive methods at 17 datasets which represents 85% of the total number of tested datasets. The ALO follows this, which achieved the smallest fitness value in the two datasets, while AOA, BSA, and CSA are the worst algorithms.
- **In terms of accuracy:** The following points can be observed from the results given in Table 5. First, the B_HBO has higher accuracy at nearly 80% of the total number of datasets. In addition, it is more stable than all other tested algorithms. However, it has the worst accuracy of the three datasets. This indicates the high efficiency of the proposed B_HBO . This is followed by the ALO, which achieved the second accuracy rank at seven datasets. At the same time, SMA is the worst algorithm. The standard deviation is computed to evaluate the stability of fitness value for each FS method. From the results of Std, it can be seen that the B_HBO is more stable than other algorithms in 14 datasets.
- **In terms of precision:** It can be seen from the results presented in Table 6 that list the precision of the proposed method B_HBO with eight wrapper FS algorithms. By examining the average precision values for all 20 datasets, it is evident that B_HBO outperforms all advanced competitor algorithms. For example, the average precision has the highest results compared with the competitive methods at eight datasets which represent 40% of the total number of tested datasets. The CSA followed this, achieving

the highest precision value in the five datasets, while LFD, BSA, AOA, and TSA are the worst algorithms.

- **In terms of sensitivity:** The results presented in Table 7 demonstrate the sensitivity of the proposed method *B_HBO* with eight wrapper FS algorithms. Examining the average sensitivity values for each of the 20 Datasets reveals that *B_HBO* outperforms all advanced competitor algorithms. Eight datasets, or 40% of the total number of datasets tested, produce the best results based on the average sensitivity. This was followed by the CSA with the highest value of precision across five datasets. LFD, BSA, AOA, and TSA are the four worst algorithms.
- **In terms of F-score and number of selected features:** In terms of F-score, Table 8 reveals that the proposed method *B_HBO* outperforms all other competitors. It has the highest results compared with the competitive methods of eight datasets which represent 40% of the total number of tested datasets. The PSO followed this, which achieved the highest F-Score value in five datasets, while LFD, BSA, SMA, and TSA are the worst algorithms.

Based on the results of Table 9, which depicts the number of selected features, the proposed method *B_HBO* exhibited excellent performance in selecting relevant features from other competitors. It has the smallest results of the competitive methods at 15 datasets, which represents 75% of the total number of tested datasets. The SMA follows this, achieving the smallest number of selected features at ten datasets.

Table 4. The average fitness values of *B_HBO* against other recent optimizers.

Data Set	Fitness								
	ALO	AOA	BSA	CSA	LFD	PSO	SMA	TSA	HBO
Arrhythmia	0.2262	0.2510	0.2490	0.2432	0.2675	0.2339	0.2372	0.2330	0.2338
Breastcancer	0.0165	0.0181	0.0189	0.0161	0.0228	0.0164	0.0209	0.0185	0.0149
BreastEW	0.0103	0.0136	0.0137	0.0096	0.0180	0.0095	0.0143	0.0121	0.0088
Congress	0.0244	0.0286	0.0273	0.0220	0.0336	0.0244	0.0292	0.0284	0.0202
Diabetes	0.1369	0.1363	0.1438	0.1357	0.1620	0.1359	0.1436	0.1418	0.1341
German	0.1351	0.1503	0.1454	0.1329	0.1640	0.1327	0.1668	0.1413	0.1319
Glass	0.1227	0.1295	0.1383	0.1211	0.1502	0.1192	0.1337	0.1338	0.1138
Heart-C	0.2413	0.2475	0.2525	0.2407	0.2649	0.2391	0.2622	0.2503	0.2348
Heart-StatLog	0.0849	0.0950	0.0979	0.0814	0.1102	0.0874	0.1062	0.0982	0.0802
Hepatitis	0.1409	0.1501	0.1614	0.1407	0.1689	0.1396	0.1562	0.1439	0.1370
Hillvalley	0.1807	0.1859	0.1932	0.1910	0.2082	0.1856	0.1801	0.1861	0.1856
Ionosphere	0.0406	0.0435	0.0569	0.0512	0.0670	0.0449	0.0409	0.0458	0.0451
Iris	0.0204	0.0205	0.0238	0.0203	0.0230	0.0229	0.0210	0.0256	0.0200
Lymphography	0.3258	0.3545	0.3496	0.3200	0.3647	0.3178	0.3794	0.3401	0.3153
Vehicule	0.1350	0.1432	0.1450	0.1341	0.1630	0.1341	0.1456	0.1397	0.1338
Vowel	0.0049	0.0116	0.0079	0.0042	0.0231	0.0045	0.0116	0.0149	0.0035
WaveformEW	0.1010	0.1159	0.1080	0.0985	0.1473	0.0973	0.1349	0.1066	0.0965
WDBC	0.0097	0.0134	0.0143	0.0094	0.0173	0.0092	0.0142	0.0113	0.0081
Wine	0.0023	0.0069	0.0071	0.0025	0.0097	0.0028	0.0072	0.0074	0.0002
Zoo	0.0124	0.0243	0.0215	0.0111	0.0262	0.0113	0.0262	0.0196	0.0089

Table 5. The average Accuracy of B_HBO against other recent optimizers.

Data Set		Accuracy								
		ALO	AOA	BSA	CSA	LFD	PSO	SMA	TSA	HBO
Arrhythmia	Avg	0.9985	0.8783	0.7035	0.8468	0.9808	0.6165	0.8352	0.9540	1.0000
Breastcancer	Avg	0.8116	0.7451	0.7416	0.7347	0.7621	0.7274	0.7329	0.7541	0.8975
	Std	0.0072	0.0163	0.0122	0.0112	0.0105	0.0137	0.0117	0.0136	0.0060
BreastEW	Avg	0.8527	0.9120	0.8679	0.8929	0.9266	0.8598	0.8639	0.9159	0.9380
	Std	0.0093	0.0122	0.0176	0.0095	0.0121	0.0310	0.0323	0.0231	0.0091
Congress	Avg	0.9253	0.8788	0.8747	0.8911	0.9129	0.8413	0.8532	0.9013	0.9445
	Std	0.0124	0.0276	0.0260	0.0164	0.0158	0.0342	0.0227	0.02123	0.0201
Diabets	Avg	1.0000	0.9547	0.9408	0.9715	1.0000	0.8583	0.9185	0.9968	1.0000
	Std	0.0000	0.0573	0.0535	0.0289	0.0000	0.0435	0.0422	0.0104	0.0000
German	Avg	0.9873	0.9278	0.9063	0.9063	0.9643	0.8429	0.8810	0.9357	0.9905
	Std	0.0136	0.0246	0.0265	0.0257	0.0223	0.0304	0.0258	0.0243	0.0134
Glass	Avg	0.9711	0.9729	0.9668	0.9747	0.9739	0.9530	0.9634	0.9733	1.0000
	Std	0.0054	0.0064	0.0072	0.0066	0.0044	0.0075	0.0077	0.0069	0.0000
Heart-C	Avg	0.9673	0.9463	0.9429	0.9523	0.9646	0.9162	0.9423	0.9607	0.9816
	Std	0.0112	0.0161	0.0143	0.0135	0.0052	0.0213	0.0151	0.0078	0.0062
Heart-StatLog	Avg	1.0000	0.9925	0.9870	0.9965	1.0000	0.9722	0.9833	1.0000	1.0000
	Std	0.0000	0.0130	0.0157	0.0060	0.0000	0.0100	0.0164	0.0000	0.0000
Hepatitis	Avg	0.9866	1.0000	1.0000	0.9977	1.0000	0.9904	0.9942	1.0000	1.0000
	Std	0.0212	0.0000	0.0000	0.0086	0.0000	0.0253	0.0182	0.0000	0.0000
Hillvalley	Avg	0.9922	0.9680	0.9612	0.9672	0.9773	0.9471	0.9626	0.9673	0.9926
	Std	0.0005	0.0062	0.0074	0.0062	0.0063	0.0062	0.0071	0.0062	0.0048
Ionosphere	Avg	0.9911	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
	Std	0.0300	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
Iris	Avg	0.9684	0.9497	0.8543	0.9133	0.9368	0.8101	0.8510	0.8675	1.0000
	Std	0.0270	0.0328	0.0330	0.0362	0.0270	0.0352	0.0319	0.0172	0.0000
Lung-Cancer	Avg	0.9463	0.9303	0.7284	0.8163	0.8972	0.5942	0.6761	0.7865	0.9225
	Std	0.0517	0.0749	0.0448	0.1128	0.0814	0.0442	0.0658	0.0644	0.0771
Lymphography	Avg	1.0000	0.9633	0.7178	0.8621	0.9132	0.6621	0.7354	0.7377	0.9922
	Std	0.0000	0.0524	0.0379	0.0571	0.0525	0.0159	0.0220	0.0310	0.0220
Vowel	Avg	1.0000	1.0000	0.9188	0.9812	0.9968	0.8479	0.8923	0.9434	1.0000
	Std	0.0000	0.0000	0.0369	0.0200	0.0112	0.0417	0.0473	0.0385	0.0000
WaveformEW	Avg	1.0000	0.9980	0.8627	0.9627	0.9843	0.7961	0.8314	0.8765	1.0000
	Std	0.0000	0.0107	0.0446	0.0423	0.0265	0.0336	0.0336	0.0237	0.0000
WDBC	Avg	1.0000	1.0000	0.9708	0.9828	1.0000	0.9527	0.9608	0.9635	1.0000
	Std	0.0000	0.0000	0.0307	0.0149	0.0000	0.0149	0.0307	0.0244	0.0000
Wine	Avg	0.9684	0.9366	0.9069	0.9217	0.9414	0.8573	0.8978	0.9078	1.0000
	Std	0.0230	0.0304	0.0163	0.0262	0.0260	0.0261	0.0215	0.0116	0.0000
Zoo	Avg	0.5750	0.5569	0.4831	0.5294	0.5521	0.4424	0.4662	0.4931	0.6783
	Std	0.0162	0.0182	0.0164	0.0162	0.0203	0.0115	0.0136	0.0137	0.0236

Table 6. The average Precision of B_HBO against other recent optimizers.

Data Set		Precision								
		ALO	AOA	BSA	CSA	LFD	PSO	SMA	TSA	HBO
Arrhythmia		0.69602	0.65225	0.66923	0.7196	0.57032	0.69944	0.61707	0.69939	0.73501
Breastcancer		0.97899	0.97807	0.97641	0.98017	0.97357	0.98032	0.9735	0.97834	0.9812
BreastEW		0.99068	0.99023	0.98643	0.99079	0.9824	0.99365	0.98849	0.99115	0.99446
Congress		0.97224	0.97063	0.97108	0.97713	0.96597	0.96983	0.96336	0.9668	0.97678
Diabets		0.85369	0.84612	0.84387	0.85675	0.83947	0.85314	0.85213	0.84902	0.85582

Table 6. Cont.

Data Set	Precision								
	ALO	AOA	BSA	CSA	LFD	PSO	SMA	TSA	HBO
German	0.85433	0.84345	0.84188	0.85536	0.83362	0.85487	0.82364	0.83641	0.85667
Glass	0.90223	0.84802	0.87037	0.8854	0.86668	0.86558	0.8351	0.82104	0.88204
Heart-C	0.71403	0.7496	0.70625	0.67328	0.65387	0.75457	0.75621	0.68932	0.73267
Heart-StatLog	0.91144	0.90911	0.91112	0.91498	0.89747	0.91717	0.89127	0.8920 6	0.9174
Hepatitis	0.86912	0.85952	0.84633	0.8603	0.83735	0.87431	0.84364	0.85234	0.86822
Hillvalley	0.81776	0.81996	0.80125	0.81978	0.79777	0.81183	0.81582	0.80539	0.81496
Ionosphere	0.96595	0.96378	0.95829	0.96182	0.94631	0.96591	0.95788	0.96454	0.96836
Iris	0.98012	0.98012	0.97358	0.976	0.97576	0.97576	0.98012	0.97123	0.98012
Lung-Cancer	0.9861	0.97916	0.97277	0.96638	0.95999	0.97971	0.94152	0.9861	0.97916
Lymphography	0.54359	0.48614	0.48422	0.54567	0.49987	0.59507	0.45346	0.49971	0.54191
Vowel	0.90546	0.89417	0.9021	0.90576	0.88834	0.90546	0.90042	0.89546	0.90546
WaveformEW	0.88087	0.88202	0.88732	0.90344	0.84512	0.90145	0.84952	0.89964	0.90171
WDBC	0.9923	0.98483	0.98724	0.99068	0.98073	0.9915	0.98356	0.99186	0.99207
Wine	0.99773	0.99361	0.99329	1	0.98732	1	0.99815	0.98628	0.99773
Zoo	0.97618	0.97648	0.9538	0.98941	0.95864	0.95713	0.94742	0.97548	0.98147

Table 7. The average Sensitivity of B_HBO against other recent optimizers.

Data Set	Sensitivity								
	ALO	AOA	BSA	CSA	LFD	PSO	SMA	TSA	HBO
Arrhythmia	0.4925	0.46676	0.44693	0.45794	0.4058	0.45507	0.48336	0.50149	0.48872
Breastcancer	0.98566	0.98326	0.97947	0.98537	0.97598	0.98402	0.97835	0.98293	0.98643
BreastEW	0.98681	0.98602	0.98226	0.98791	0.97629	0.9901	0.98398	0.98759	0.99057
Congress	0.97462	0.97127	0.97436	0.97947	0.96988	0.97411	0.96591	0.97087	0.97983
Diabets	0.8443	0.83728	0.83921	0.84448	0.83357	0.84397	0.84421	0.8449	0.84593
German	0.81175	0.80722	0.80032	0.82103	0.78762	0.81286	0.79333	0.80175	0.81944
Glass	0.8526	0.86605	0.84753	0.84734	0.80424	0.86312	0.77511	0.79492	0.8531
Heart-C	0.55174	0.56548	0.5372	0.54678	0.50586	0.57511	0.53851	0.54874	0.56413
Heart-StatLog	0.91167	0.90583	0.90889	0.9125	0.895	0.915	0.88611	0.88833	0.915
Hepatitis	0.86417	0.85466	0.83507	0.84987	0.82717	0.86149	0.84201	0.8478	0.85674
Hillvalley	0.81649	0.81867	0.79866	0.81804	0.79476	0.80974	0.8143	0.80311	0.81244
Ionosphere	0.95175	0.94397	0.92635	0.93809	0.91661	0.94603	0.9473	0.94339	0.94429
Iris	0.98	0.98	0.97333	0.97555	0.97555	0.97555	0.98	0.97111	0.98
Lung-Cancer	0.96296	0.94444	0.92592	0.9074	0.88888	0.94444	0.92994	0.96296	0.94444
Lymphography	0.45097	0.41832	0.4173	0.47831	0.42561	0.48819	0.39708	0.4494	0.48434
Vowel	0.9064	0.89495	0.90303	0.90673	0.88855	0.9064	0.90135	0.8963	0.9064
WaveformEW	0.87898	0.8821	0.88743	0.90354	0.84373	0.90152	0.84961	0.89972	0.90182
WDBC	0.98774	0.98133	0.98273	0.98681	0.97789	0.98727	0.98007	0.98695	0.98917
Wine	0.99843	0.9953	0.9953	1	0.99061	1	0.99843	0.98754	0.99843
Zoo	0.98204	0.97142	0.92728	0.99046	0.95475	0.94285	0.92142	0.95237	0.9857

Table 8. The average test F – score of B_HBO against other recent optimizers.

Data Set	F-score								
	ALO	AOA	BSA	CSA	LFD	PSO	SMA	TSA	HBO
Arrhythmia	0.5762	0.5441	0.5359	0.5592	0.4731	0.5497	0.5412	0.5838	0.5864
Breastcancer	0.9823	0.9807	0.9779	0.9828	0.9748	0.9822	0.9759	0.98063	0.9838
BreastEW	0.9887	0.9881	0.9843	0.9894	0.9793	0.9919	0.9862	0.9894	0.9925
Congress	0.9734	0.9709	0.9727	0.9783	0.9679	0.9719	0.9646	0.9688	0.9783
Diabetes	0.8489	0.8417	0.8415	0.8506	0.8365	0.8485	0.8482	0.8469	0.8508
German	0.8325	0.8249	0.8206	0.8378	0.8099	0.8333	0.8082	0.8186	0.8377
Glass	0.8767	0.8569	0.8585	0.8659	0.8341	0.8640	0.8038	0.8072	0.867
Heart-C	0.6219	0.6443	0.6102	0.6027	0.5693	0.6527	0.628	0.6096	0.6369
Heart-StatLog	0.9116	0.9075	0.91	0.9137	0.8962	0.9162	0.8887	0.8901	0.9162
Hepatitis	0.8666	0.857	0.8407	0.855	0.8322	0.8678	0.8428	0.8501	0.8624
Hillvalley	0.8171	0.8193	0.7999	0.8189	0.7963	0.8108	0.8151	0.8043	0.8137
Ionosphere	0.9588	0.9538	0.942	0.9498	0.9312	0.9559	0.9525	0.9539	0.9562
Iris	0.9801	0.9801	0.9735	0.9758	0.9757	0.9757	0.9801	0.9712	0.9801
Lung-Cancer	0.9743	0.9615	0.9487	0.9359	0.9231	0.9615	0.9356	0.9743	0.9615
Lymphography	0.4929	0.4475	0.4464	0.5089	0.4578	0.5352	0.4229	0.4729	0.5115
Vowel	0.9059	0.8946	0.9026	0.9063	0.8885	0.9059	0.90088	0.8959	0.9059
WaveformEW	0.8799	0.8821	0.8874	0.9035	0.8444	0.9015	0.8496	0.8997	0.9018
WDBC	0.99	0.9831	0.9849	0.9888	0.9793	0.98934	0.9818	0.9894	0.9906
Wine	0.9981	0.9945	0.9943	1	0.9889	1	0.9983	0.9869	0.9981
Zoo	0.9791	0.9739	0.9403	0.9899	0.9567	0.9499	0.9342	0.9636	0.9836

Table 9. The number of selected features of B_HBO against other recent optimizers.

Data Set	Number of Selected Features								
	ALO	AOA	BSA	CSA	LFD	PSO	SMA	TSA	HBO
Arrhythmia	91	28	139	128	143	133	12	123	135
Breastcancer	9	6	6	8	6	7	8	5	5
BreastEW	14	10	20	19	18	16	16	14	9
Congress	8	9	8	12	10	9	11	8	4
Diabets	4	3	3	6	4	5	4	3	3
German	14	10	15	18	12	16	8	10	15
Glass	5	4	6	6	7	6	4	4	4
Heart-C	8	7	8	9	8	8	7	7	4
Heart-StatLog	12	8	9	12	9	11	6	9	6
Hepatitis	11	6	11	14	10	9	4	9	12
Hillvalley	16	12	46	51	50	53	5	46	53
Ionosphere	7	9	13	15	12	16	5	12	4
Iris	2	2	3	2	3	3	2	3	2
Lung-Cancer	29	17	27	34	32	26	12	26	12
Lymphography	11	10	13	11	11	11	8	10	8

Table 9. Cont.

Data Set	Number of Selected Features								
	ALO	AOA	BSA	CSA	LFD	PSO	SMA	TSA	HBO
Vowel	10	8	8	9	6	10	8	6	6
WaveformEW	27	17	24	25	21	24	6	19	19
WDBC	14	14	15	20	17	16	15	14	6
Wine	8	8	10	9	7	7	8	8	7
Zoo	12	8	11	10	8	10	8	9	8

In this paper, we can see that the performance of the proposed *B_HBO* algorithm for feature selection and classification was investigated using six different statistical metrics (e.g., average fitness value, average accuracy, average sensitivity, average precision, average F_score, and number of selected features over 51 runs for each algorithm) Table 5 displays the average accuracy for the proposed *B_HBO* and the eight other compared algorithms. As shown, the proposed *B_HBO* recorded the best fitness values in all the used datasets. The findings presented in Table 5 demonstrate that the suggested *B_HBO* outperformed competing methods in nearly all datasets. Additionally, this method has the highest accuracy rate in 80% of the dataset. ALO is ranked the second algorithm in performance after the proposed *B_HBO* while SMA takes the third rank in performance. The proposed *B_HBO* algorithm achieved the best result for most datasets.

Precision and sensitivity are shown in Tables 6 and 7. The higher the algorithm's precision and sensitivity, the better its performance. It is easy to see that the proposed *B_HBO* has high precision and sensitivity values in eight datasets. At the same time, the CSA algorithm provides better precision and sensitivity in only six datasets. The ALO algorithm takes the third level in performance in three datasets. Previous results ensure the superiority of the proposed *B_HBO* algorithm over other compared algorithms.

Table 8 shows that the proposed *B_HBO* provides a higher F-score rate than others. The proposed *B_HBO* algorithm provides a higher f-score by 160% than (CSA, PSO) while it is higher by 200% than (ALO). So HBO is ranked the first algorithm in performance, then CSA and PSO came in the second rank followed by ALO ranked the third in performance. We notice that BSA and LFD algorithms ranked last and performed worst.

Table 9 displays the number of selected features for each technique during its evaluation. The results demonstrate that B HBO is highly effective for the FS procedure.

5.4. Convergence Curve

This section is devoted to the asymptotic evaluation of the proposed *B_HBO* algorithm for the FS problem on various carefully chosen datasets. It illustrates the relationship between the number of optimization iterations, the prediction error attained thus far, and the graphical convergence curve of the proposed *B_HBO* technique. The convergence curve of *B_HBO* with varying MHs, such as ALO, AOA, BSA, CSA, LFD, PSO, SMA, and TSA on 20 medical benchmark datasets, is shown in Figures 2–5. It illustrated convergence curves of *B_HBO* with *k*-NN.

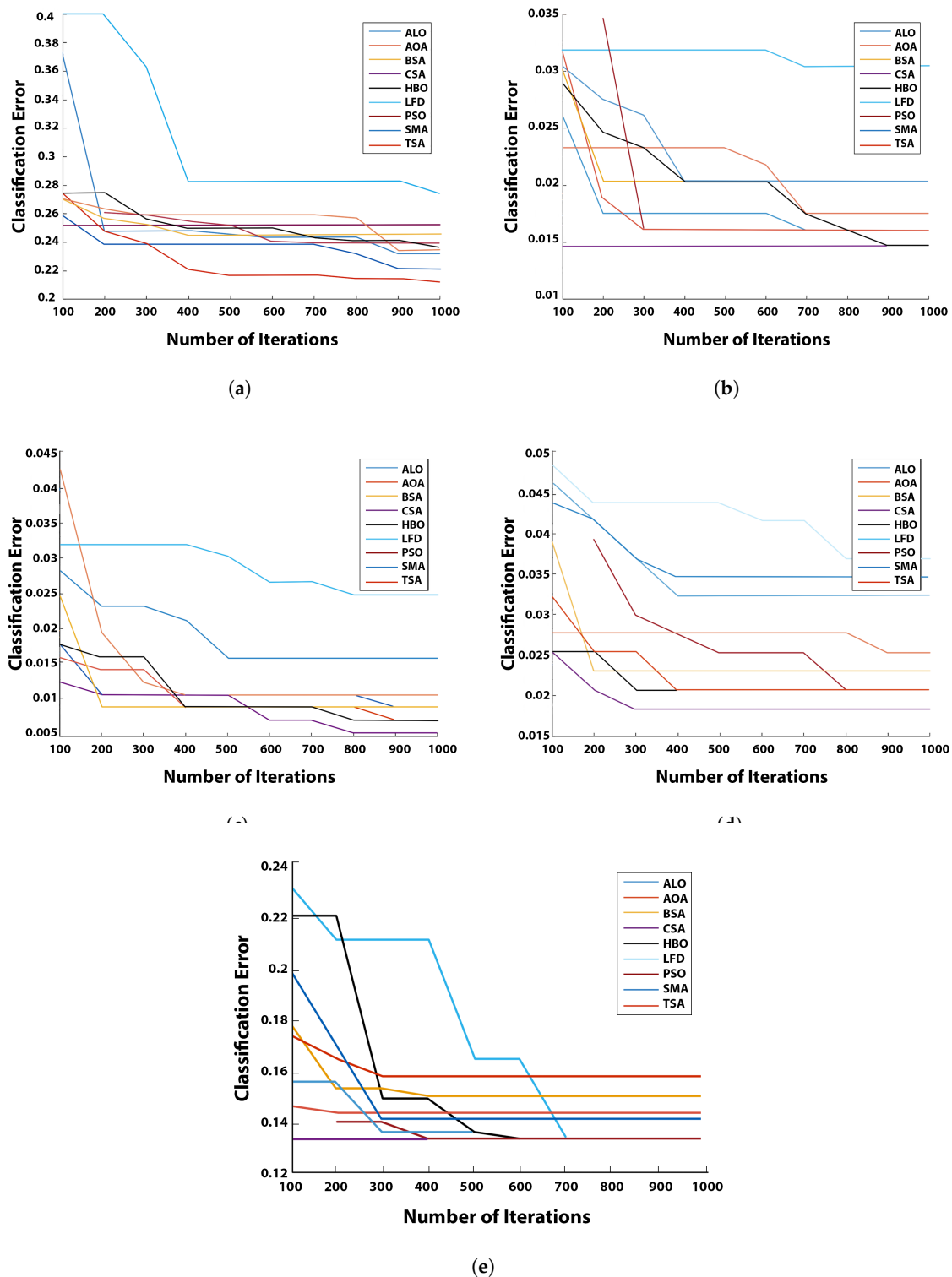


Figure 2. (a) Convergence Curve of the proposed approach for Arrhythmia dataset; (b) Convergence Curve of the proposed approach for Breast-cancer dataset; (c) Convergence Curve of the proposed approach for BreastEW dataset; (d) Convergence Curve of the proposed approach for CongressEW dataset; (e) Convergence Curve of the proposed approach for Diabetes dataset. Convergence Curve of the proposed approach (B_HBO) over 1000 iterations as a stop criterion for Arrhythmia, Breastcancer, BreastEW, CongressEW, and Diabetes datasets.

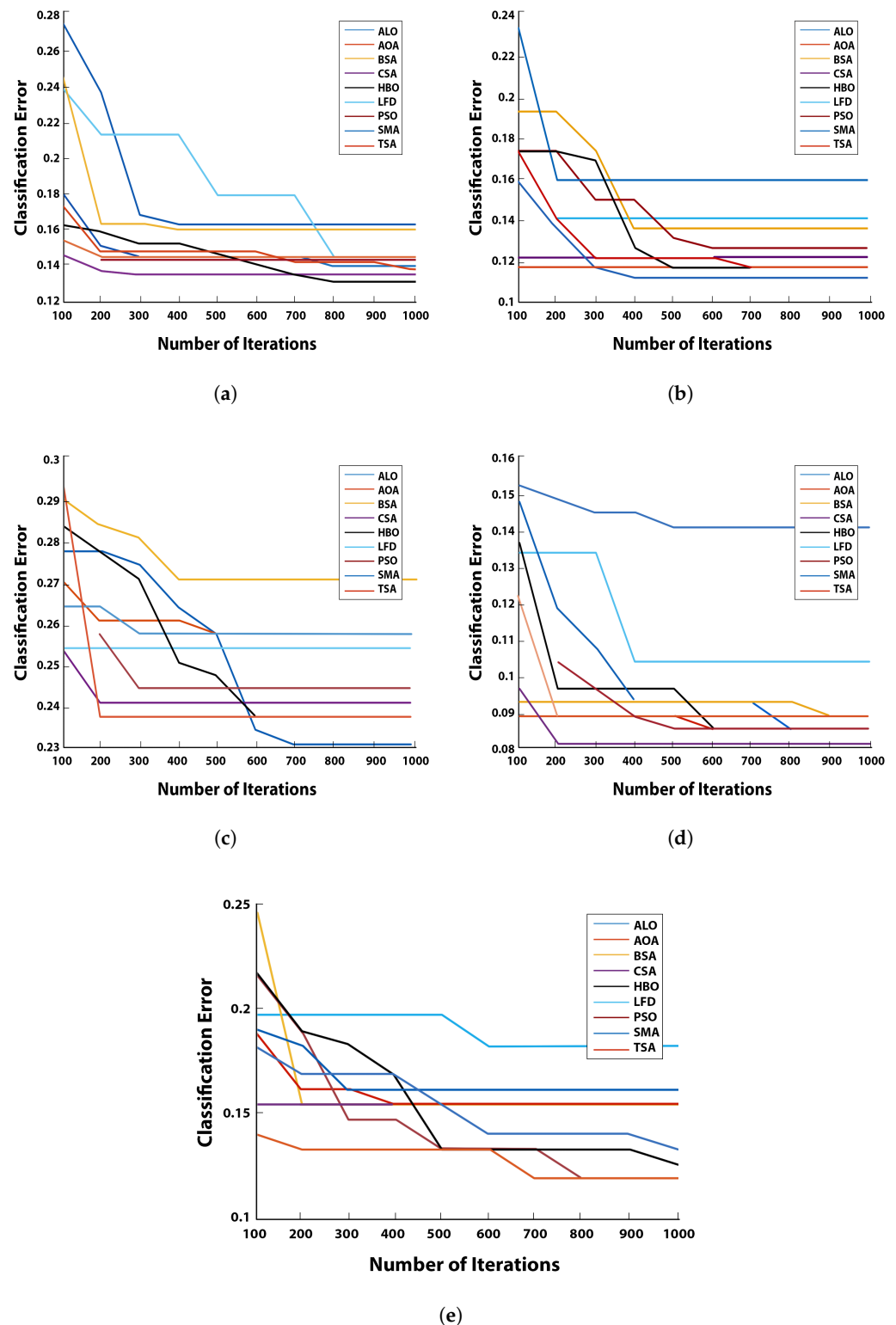


Figure 3. (a) Convergence Curve of the proposed approach for German dataset; (b) Convergence Curve of the proposed approach for Glass dataset; (c) Convergence Curve of the proposed approach for Heart-C dataset; (d) Convergence Curve of the proposed approach for Heart-StatLog dataset; (e) Convergence Curve of the proposed approach for Hepatitis dataset.

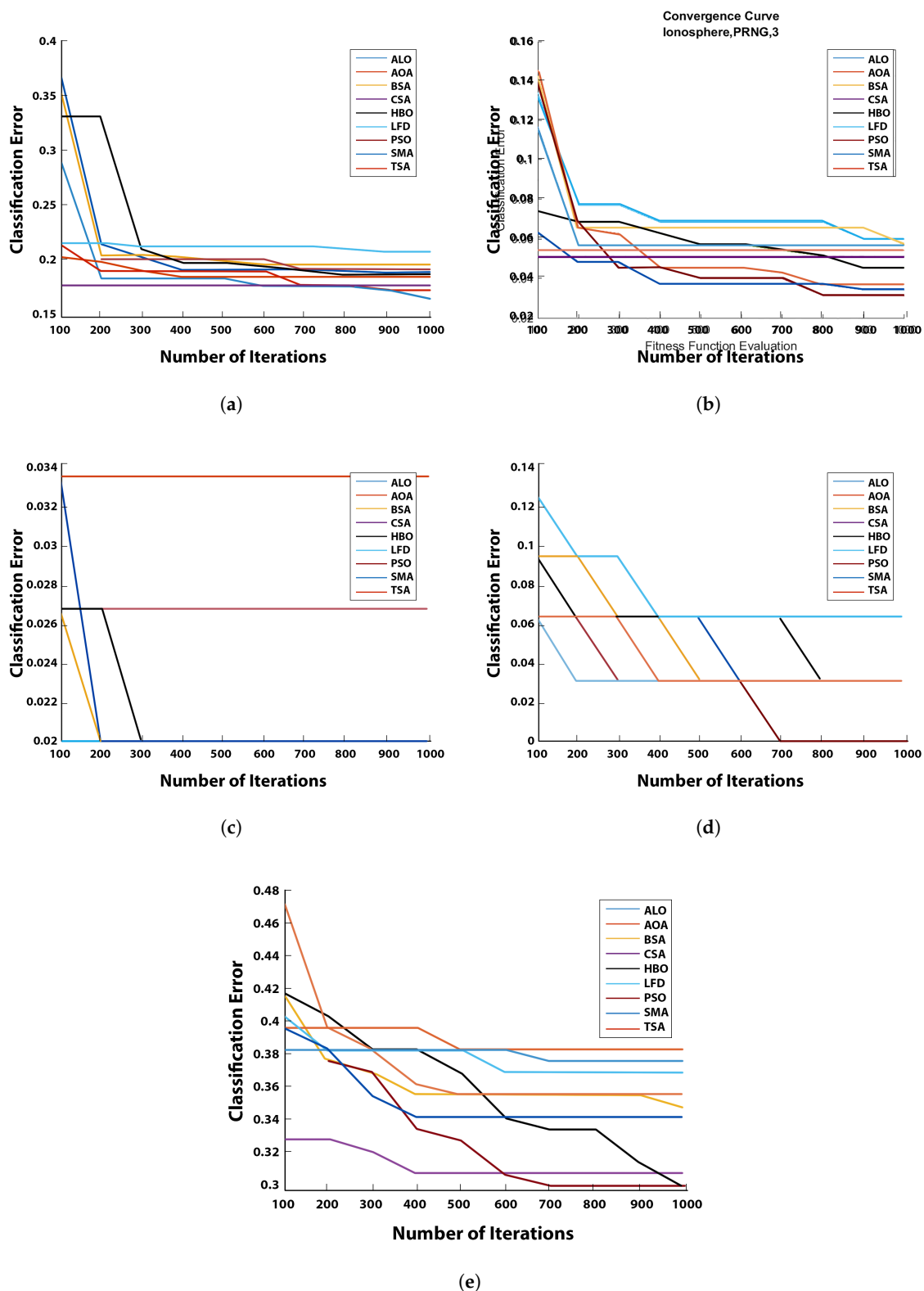


Figure 4. (a) Convergence Curve of the proposed approach for Hillvalley dataset; (b) Convergence Curve of the proposed approach for Ionosphere dataset; (c) Convergence Curve of the proposed approach for Iris dataset; (d) Convergence Curve of the proposed approach for Lung-Cancer dataset; (e) Convergence Curve of the proposed approach for Lymphography dataset.

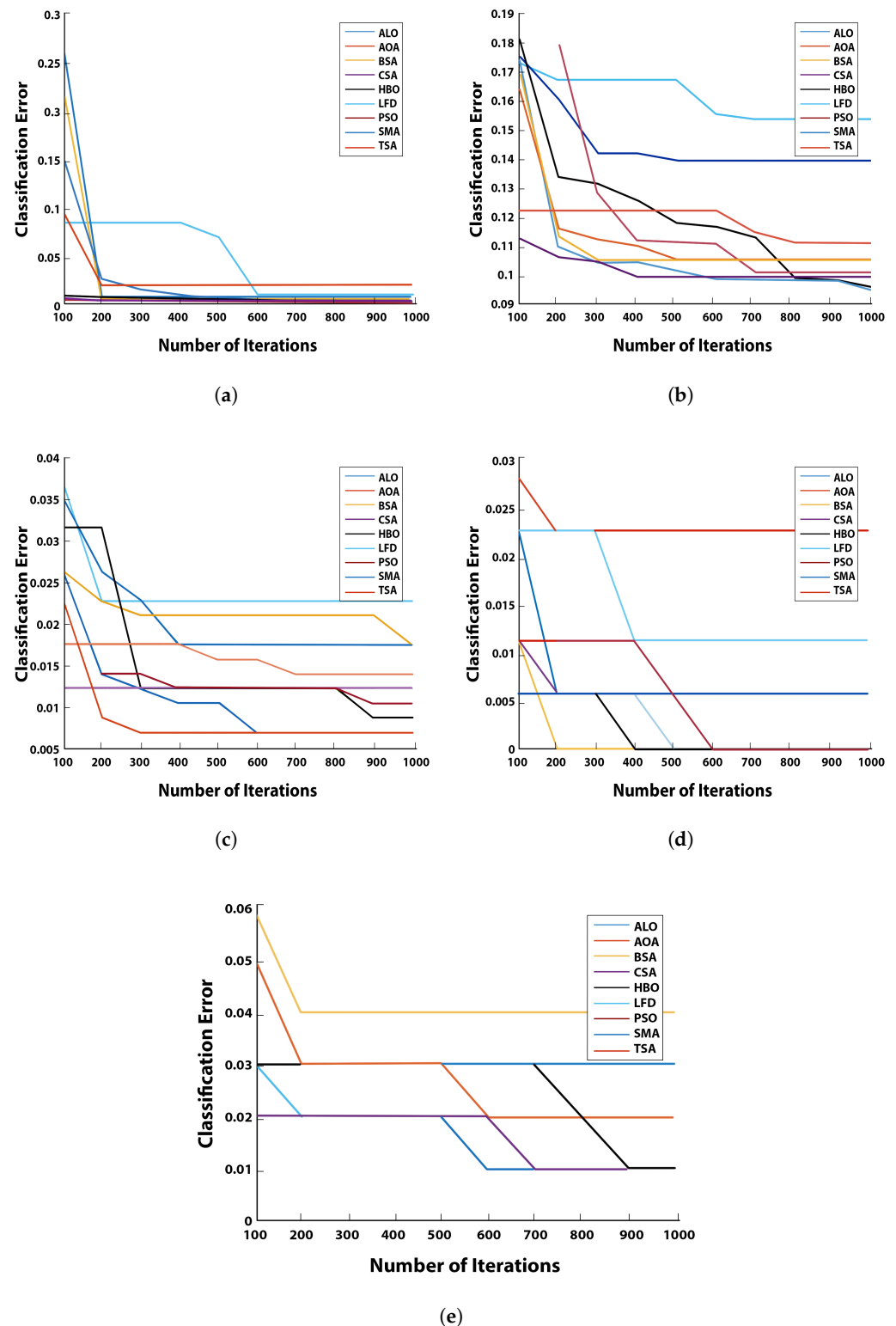


Figure 5. (a) Convergence Curve of the proposed approach for Vowel dataset; (b) Convergence Curve of the proposed approach for WaveformEW dataset; (c) Convergence Curve of the proposed approach for WDBC dataset; (d) Convergence Curve of the proposed approach for Wine dataset; (e) Convergence Curve of the proposed approach for Zoo dataset.

As can be observed in the graphs, almost every B_HBO had better outcomes than the others because their curves were higher than the other algorithms. It can be shown

that *B_HBO* causes an increase in the convergence rate toward the optimal solutions. For example, this can be noticed in the Diabetes, German, Iris, Lymphography, Vowel, Waveform_EW, Wine, and Zoo datasets. Most of these high rates of convergence are obtained at high dimension datasets.

5.5. Boxplot

The boxplot is used to analyze further the behavior of *B_HBO* in terms of different performance measures. Figures 6–9 show boxplots of the accuracies for all datasets achieved by the optimizers; ALO, AOA, BSA, CSA, LFD, PSO, SMA, and TSA on 20 medical benchmark datasets, and the proposed method *B_HBO* with *k*-NN. The minimum, maximum, median, first quartile (Q_1), and third quartile (Q_3) of the data are the five elements of a boxplot. In addition, the red line inside the box indicated the median value representing the algorithms' categorization accuracy. Compared to the other algorithms, *B_HBO* has a higher number of boxplots.

It is evident that the *B_HBO* has the lowest boxplot for fitness value in most tested datasets, especially those with high dimensions, except for four datasets (arrhythmia, Hepatitis, Hillvalley, and Lymphography). By analyzing the boxplot results, the following points can be reached: First, the presented *B_HBO* has a lower boxplot at 80% of the datasets. In addition, there are some datasets' boxplot plots that indicate that the competitive FS methods nearly have the same statistical description. In addition, most of the obtained results belong to the first quartile, which indicates that the proposed *B_HBO* obtained a small classification error.

We can conclude that the *B_HBO* with *k*-NN has the best boxplots for most datasets compared with the other algorithms. The *B_HBO* algorithm's median has a greater value. Depending on the dataset, the second-best algorithm is ALO.

Finally, it is clear that:

- **box plots** Let us note that *B_HBO* outperforms Ant Lion Optimizer (ALO), Archimedes Optimization Algorithm (AOA), Backtracking Search Algorithm (BSA), Crow Search Algorithm (CSA), Levy flight distribution (LFD), Particle Swarm Optimization (PSO), Slime Mold Algorithm (SMA), and Tree Seed Algorithm (TSA) (TSA). In final, it is easy to note that:
- The performance of the proposed method is compared to the performance of eight different algorithms. The results reveal the higher categorization, accuracy, number of selected characteristics, sensitivity, and specificity of our proposed method.

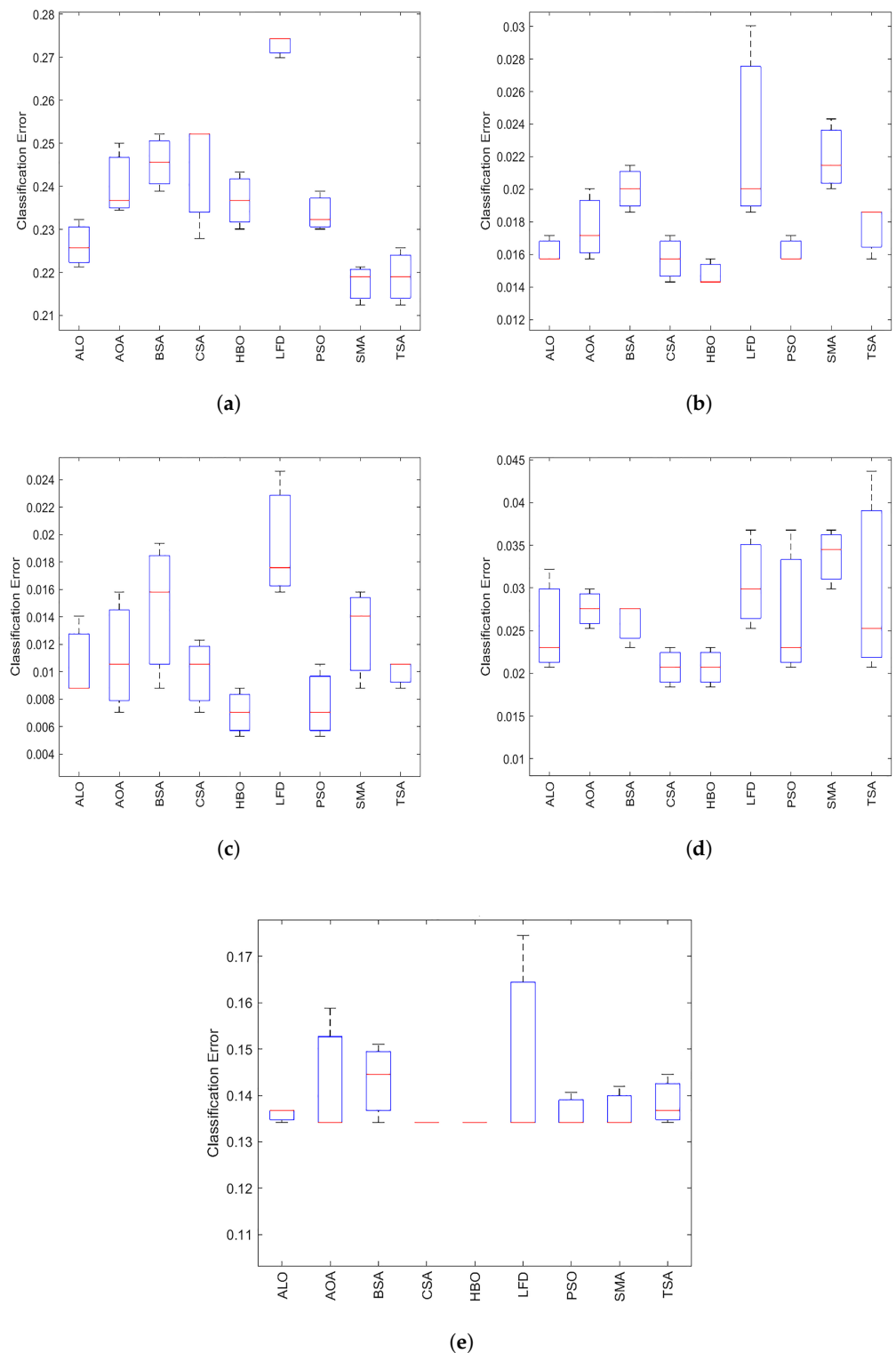


Figure 6. (a) Classification error of the proposed approach for Arrhythmia dataset; (b) Classification error of the proposed approach for Breastcancer dataset; (c) Classification error of the proposed approach for BreastEW dataset; (d) Classification error of the proposed approach for Congress dataset; (e) Classification error of the proposed approach for Diabetes dataset.

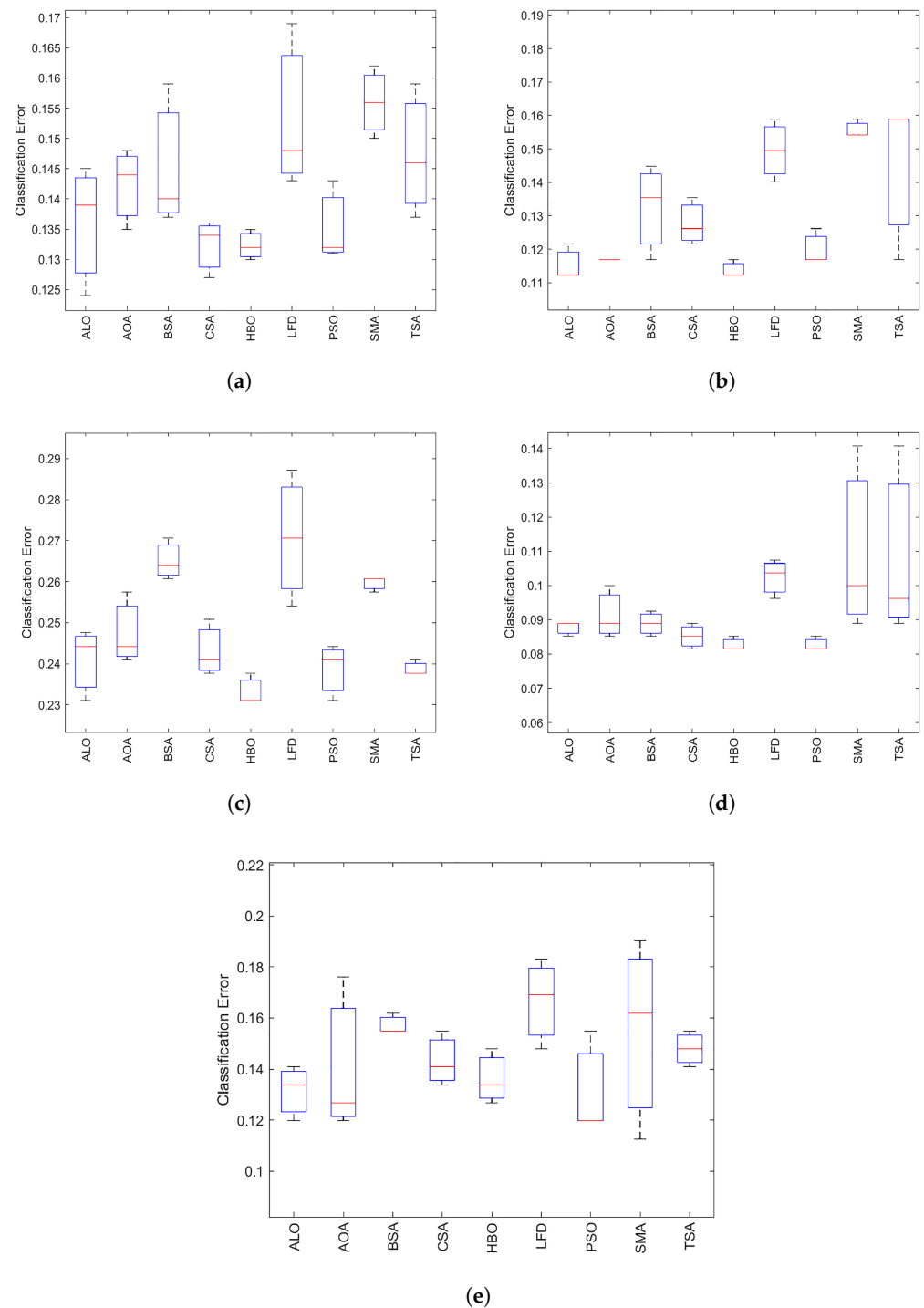


Figure 7. Boxplots of the results achieved by the B_HBO regarding classification error over German, Glass, Heart-C, Heart-StatLog, and Hepatitis datasets regarding classification error. (a) Classification error of the proposed approach for German dataset. (b) Classification error of the proposed approach for Glass dataset. (c) Classification error of the proposed approach for Heart-C dataset. (d) Classification error of the proposed approach for Heart-StatLog dataset. (e) Classification error of the proposed approach for Hepatitis dataset.

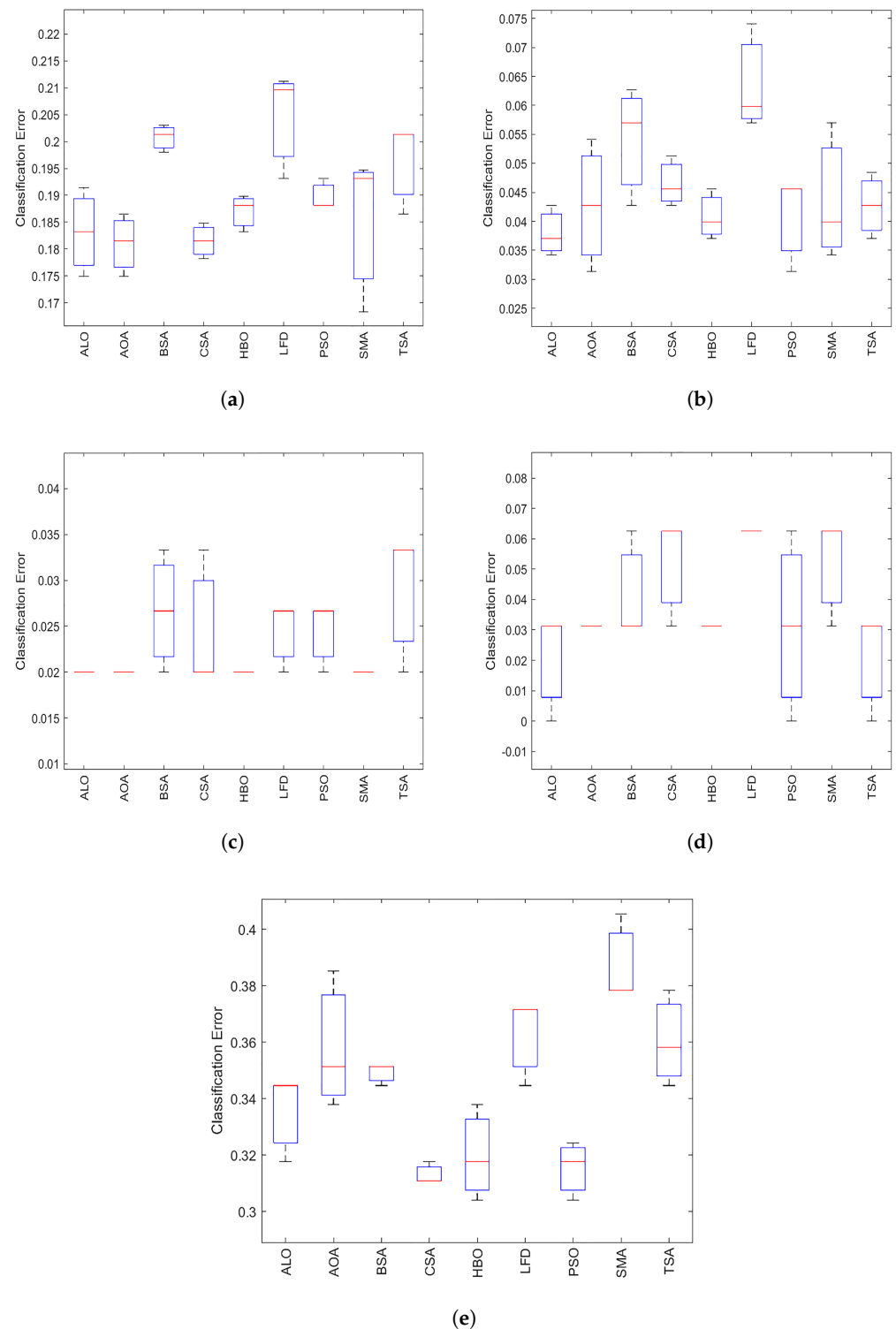


Figure 8. Boxplots of the results achieved by the B_HBO regarding classification error over Hillvalley, Ionosphere, Iris, Lung-Cancer, and Lymphography datasets regarding classification error. **(a)** Classification error of the proposed approach for Hillvalley dataset. **(b)** Classification error of the proposed approach for Ionosphere dataset. **(c)** Classification error of the proposed approach for Iris dataset. **(d)** Classification error of the proposed approach for Lung-Cancer dataset. **(e)** Classification error of the proposed approach for Lymphography dataset.

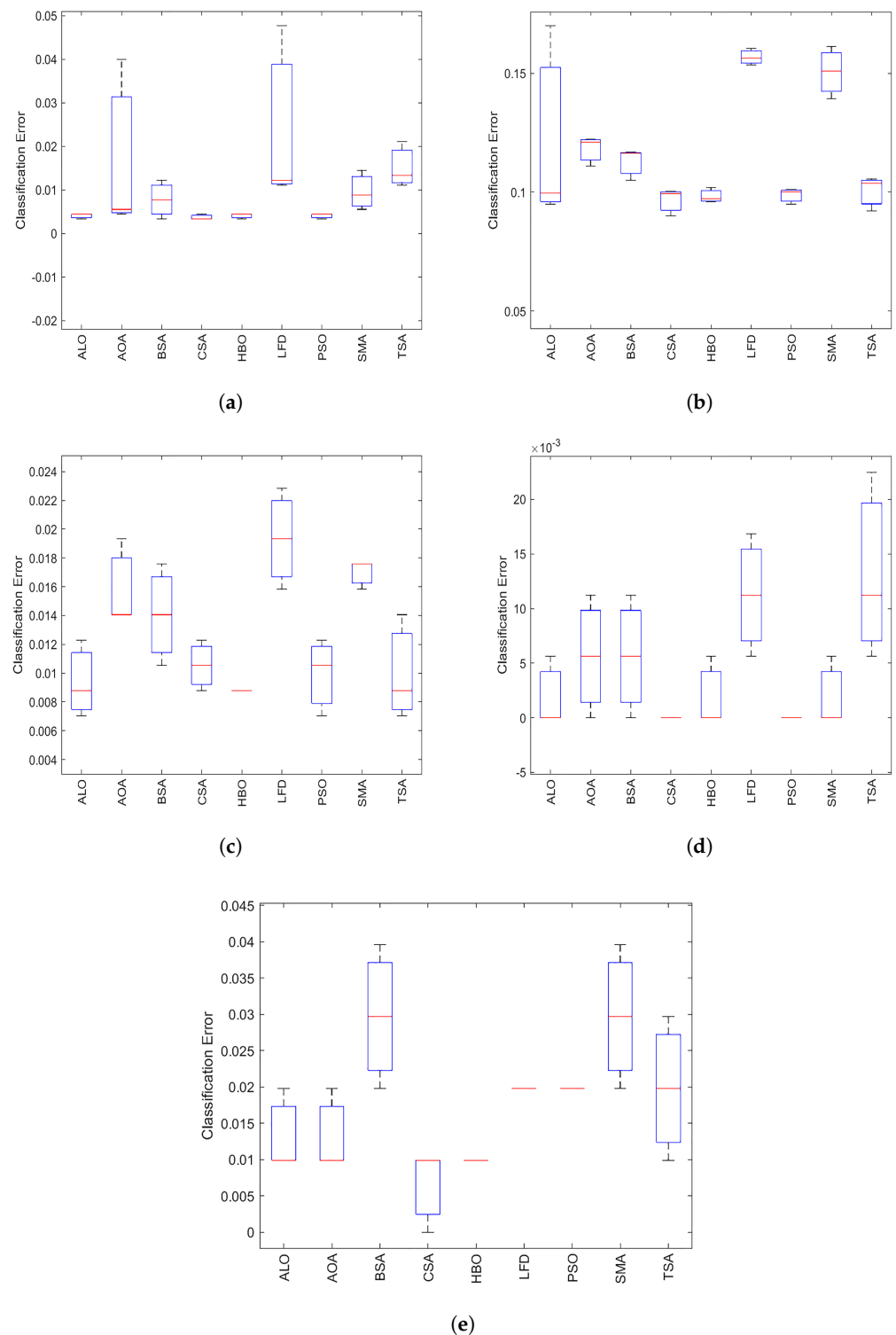


Figure 9. Boxplots of the results achieved by the B_HBO regarding classification error over Vowel, WaveformEW, WDBC, Wine, and Zoo datasets regarding classification error. (a) Classification error of the proposed approach for Vowel dataset. (b) Classification error of the proposed approach for WaveformEW dataset. (c) Classification error of the proposed approach for WDBC dataset. (d) Classification error of the proposed approach for Wine dataset. (e) Classification error of the proposed approach for Zoo dataset.

5.6. The Wilcoxon Test

Statistical analysis is necessary to compare the efficiency of *B_HBO* to that of other competitive algorithms. Wilcoxon's test assesses the superiority of the presented *B_HBO* over the other FS methods. The main aim of using Wilcoxon's test is to determine whether there is a significant difference between *B_HBO* (as control group) and each of the tested FS methods. Since Wilcoxon's test is the pair-wise non-parametric statistical test, in this test, there are two hypotheses: the first one is called null and supposes there is no significant difference between the *B_HBO* and other methods. The second hypothesis is called the alternative, and it assumes there is a significant difference. The alternative hypothesis is accepted if the *p*-value is less than 0.05. Table 10 shows the *p*-value obtained using Wilcoxon's rank-sum test for the accuracy. From the results, it can be seen that *B_HBO* has a significant difference in accuracy value with ALO, AOA, BSA, CSA, LFD, PSO, SMA, and TSA at 17 datasets. In most cases, there is a significant difference with other methods, with nearly more than 14 datasets. The combination of MRFO and SCA enhances the performance of determining the relevant features with increasing classification accuracy. Following this criterion, *B_HBO* outperforms all other algorithms to varying degrees, indicating that *B_HBO* benefits from extensive exploitation. In general, *B_HBO* is statistically significant with 85% of algorithms. Therefore, we can conclude that *B_HBO* has a high exploration capability to investigate the most promising regions of the search space and provides superior results compared to competing algorithms.

Table 10. Wilcoxon ranksum Statistical test based on Accuracy.

Data Set	Accuracy								
	ALO	AOA	BSA	CSA	LFD	PSO	SMA	TSA	HBO
Arrhythmia	1.32×10^{-20}	1.35×10^{-20}	1.33×10^{-20}	1.28×10^{-20}	1.33×10^{-20}	1.31×10^{-20}	1.35×10^{-20}	1.36×10^{-20}	1.29×10^{-20}
Breastcancer	7.95×10^{-21}	1.10×10^{-20}	1.16×10^{-20}	3.41×10^{-21}	1.27×10^{-20}	6.50×10^{-21}	1.19×10^{-20}	1.10×10^{-20}	2.31×10^{-21}
BreastEW	9.24×10^{-21}	1.14×10^{-20}	1.12×10^{-20}	8.54×10^{-21}	1.19×10^{-20}	8.71×10^{-21}	1.21×10^{-20}	1.16×10^{-20}	4.83×10^{-21}
Congress	1.02×10^{-20}	1.20×10^{-20}	9.92×10^{-21}	5.33×10^{-21}	1.22×10^{-20}	9.95×10^{-21}	1.27×10^{-20}	1.27×10^{-20}	4.66×10^{-21}
Diabets	3.70×10^{-21}	2.28×10^{-21}	1.16×10^{-20}	6.91×10^{-22}	1.38×10^{-20}	1.11×10^{-21}	4.10×10^{-21}	8.74×10^{-21}	9.57×10^{-24}
German	1.35×10^{-20}	1.37×10^{-20}	1.36×10^{-20}	1.31×10^{-20}	1.37×10^{-20}	1.36×10^{-20}	1.38×10^{-20}	1.38×10^{-20}	1.33×10^{-20}
Glass	8.68×10^{-21}	1.09×10^{-20}	1.30×10^{-20}	1.04×10^{-20}	1.27×10^{-20}	9.44×10^{-21}	1.08×10^{-20}	1.26×10^{-20}	1.63×10^{-21}
Heart-C	1.14×10^{-20}	1.30×10^{-20}	1.28×10^{-20}	1.18×10^{-20}	1.27×10^{-20}	1.19×10^{-20}	1.32×10^{-20}	1.23×10^{-20}	7.77×10^{-21}
Heart-StatLog	1.10×10^{-20}	1.29×10^{-20}	1.25×10^{-20}	8.22×10^{-21}	1.29×10^{-20}	7.56×10^{-21}	1.25×10^{-20}	1.30×10^{-20}	8.39×10^{-21}
Hepatitis	1.08×10^{-20}	1.13×10^{-20}	1.20×10^{-20}	1.03×10^{-20}	1.08×10^{-20}	1.11×10^{-20}	1.21×10^{-20}	1.20×10^{-20}	8.71×10^{-21}
Hillvalley	1.35×10^{-20}	1.36×10^{-20}	1.34×10^{-20}	1.27×10^{-20}	1.32×10^{-20}	1.30×10^{-20}	1.33×10^{-20}	1.34×10^{-20}	1.23×10^{-20}
Ionosphere	1.07×10^{-20}	1.10×10^{-20}	1.28×10^{-20}	1.14×10^{-20}	1.24×10^{-20}	1.23×10^{-20}	1.23×10^{-20}	1.24×10^{-20}	1.10×10^{-20}
Iris	2.46×10^{-23}	2.45×10^{-23}	3.78×10^{-21}	1.55×10^{-23}	2.66×10^{-21}	1.58×10^{-21}	5.66×10^{-23}	6.24×10^{-21}	9.57×10^{-24}
Lung-Cancer	6.97×10^{-14}	3.86×10^{-19}	3.48×10^{-17}	1.19×10^{-17}	1.80×10^{-21}	7.42×10^{-12}	3.86×10^{-19}	4.35×10^{-14}	2.23×10^{-16}
Lymphography	1.22×10^{-20}	1.26×10^{-20}	1.28×10^{-20}	1.13×10^{-20}	1.27×10^{-20}	1.23×10^{-20}	1.25×10^{-20}	1.22×10^{-20}	8.79×10^{-21}
Vowel	4.40×10^{-21}	1.31×10^{-20}	1.00×10^{-20}	3.96×10^{-22}	1.36×10^{-20}	5.24×10^{-21}	1.29×10^{-20}	1.34×10^{-20}	3.02×10^{-22}
WaveformEW	1.38×10^{-20}	1.39×10^{-20}	1.38×10^{-20}	1.38×10^{-20}	1.39×10^{-20}	1.38×10^{-20}	1.39×10^{-20}	1.38×10^{-20}	1.35×10^{-20}
WDBC	9.74×10^{-21}	1.20×10^{-20}	1.14×10^{-20}	9.21×10^{-21}	1.21×10^{-20}	8.81×10^{-21}	1.10×10^{-20}	9.79×10^{-21}	5.33×10^{-21}
Wine	7.52×10^{-6}	1.73×10^{-15}	5.79×10^{-15}	3.66×10^{-6}	4.71×10^{-18}	2.67×10^{-8}	5.58×10^{-14}	5.93×10^{-14}	0.15933
Zoo	1.39×10^{-20}	1.22×10^{-19}	5.12×10^{-21}	1.22×10^{-18}	5.55×10^{-21}	2.80×10^{-17}	6.13×10^{-21}	5.71×10^{-18}	3.69×10^{-18}
Zoo	16/0/4	17/0/2	16/2/2	10/1/9	17/2/1	14/0/6	18/1/1	18/0/2	3/11/6

6. Conclusions

This paper presents a novel feature selection method based on the Heap-Based Optimizer (HBO). This paper proposes a new binary version of the basic Heap-based optimizer (HBO) called BHBO to solve the FS problem. The experiments are applied to 20 benchmark datasets from UCI datasets, and five evaluation criteria are performed to investigate the performance of the proposed algorithm. The experimental results revealed that the proposed algorithms achieved superior results versus eight of the recent state-of-the-art algorithms, including ALO, AOA, BSA, CSA, LFD, PSO, SMA, and TSA, according to the experimental results. Furthermore, the results proved that *B_HBO* had achieved the smallest number of features with better classification accuracy. The findings and results showed that the HBO achieved the minimum number of selected features with the best accuracy in a reasonable amount of time for most datasets. The HBO exhibited a considerable benefit for significantly big datasets. Regarding average accuracy, sensitivity, specificity, and feature size,

HBO came in first, with the least number of specified features. After, HBO, ALO and CSA are ranked second in terms of performance.

Author Contributions: All authors contributed equally to this paper, where; D.S.A.E.: Supervision, Methodology, Conceptualization, Formal analysis, Writing—review & editing. F.R.P.P.: Methodology, Formal analysis, Methodology, Writing—review & editing, Implementation the code and running. M.A.S.A.: Software, Formal analysis, Resources, Writing—original draft, Supervision, Conceptualization, Methodology, Writing—review & editing. All authors have read and agreed to the published version of the manuscript.

Funding: This work was funded by the Deanship of Scientific Research, Vice Presidency for Graduate Studies and Scientific Research, King Faisal University, Saudi Arabia Project No. AN000565 .

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: This work was supported by the Deanship of Scientific Research, Vice Presidency for Graduate Studies and Scientific Research, King Faisal University, Saudi Arabia [Project No. AN000565].

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

ACO	Ant Colony Optimization
ALO	Ant Lion Optimization
AOA	Archimedes Optimization Algorithm
BALO	Binary Ant Lion Optimization
BCFA	Binary Clonal Flower Pollination Algorithm
BGOA	Binary Grasshopper Optimization Algorithm
BGSA	Binary Gravitational Search Algorithm
BGWO	Binary Gray Wolf Optimization
B_HBO	Binary Heap Based Optimizer
BSA	Back Tracking Search Algorithm
BSSO	Binary Swallow Swarm Optimization
BSHO	Binary Spotted Hyena Optimizer
BPSO	Binary Particle Swarm Optimization
BWOA	Binary Whale Optimization Algorithm
CSA	Crow Search Algorithm
EO	Equilibrium Optimizer
FOA	Forest Optimization Algorithm
FPA	Flower Pollination Algorithm
FS	Feature Selection
GA	Genetic Algorithm
GBO	Gradient-Based Optimizer
GSA	Gravitational Search Algorithm
GWO	Gray wolf Optimizer
HBO	Heap-Based Optimizer
HGSO	Henry Gas Solubility Optimization Algorithm
LFD	Levy flight Distribution

PSO	Particle Swarm Optimization
SCA	Sine Cosine Algorithm
SMA	Slime Mold Algorithm
SSA	Salp Swarm Algorithm
TSA	Tree Seed Algorithm
WOA	Whale Optimization Algorithm

References

1. Zawbaa, H.M.; Emary, E.; Grosan, C. Feature selection via chaotic antlion optimization. *PLoS ONE* **2016**, *11*, e0150652.
2. Faris, H.; Mafarja, M.M.; Heidari, A.A.; Aljarah, I.; Ala'M, A.Z.; Mirjalili, S.; Fujita, H. An efficient binary salp swarm algorithm with crossover scheme for feature selection problems. *Knowl.-Based Syst.* **2018**, *154*, 43–67.
3. Huang, Y.; Jin, W.; Yu, Z.; Li, B. Supervised feature selection through Deep Neural Networks with pairwise connected structure. *Knowl.-Based Syst.* **2020**, *204*, 106202.
4. Mafarja, M.; Aljarah, I.; Heidari, A.A.; Faris, H.; Fournier-Viger, P.; Li, X.; Mirjalili, S. Binary dragonfly optimization for feature selection using time-varying transfer functions. *Knowl.-Based Syst.* **2018**, *161*, 185–204.
5. Zhang, J.; Hu, X.; Li, P.; He, W.; Zhang, Y.; Li, H. A hybrid feature selection approach by correlation-based filters and svm-rfe. In Proceedings of the 2014 22nd International Conference on Pattern Recognition, Stockholm, Sweden, 24–28 August 2014; pp. 3684–3689.
6. Teng, X.; Dong, H.; Zhou, X. Adaptive feature selection using v-shaped binary particle swarm optimization. *PLoS ONE* **2017**, *12*, e0173907.
7. Motoda, H.; Liu, H. *Feature Selection, Extraction and Construction*; Communication of IICM (Institute of Information and Computing Machinery Taiwan): 2002; Volume 5, p. 2.
8. Talbi, E.G. *Metaheuristics: From Design to Implementation*; John Wiley & Sons: Hoboken, NJ, USA, 2009; Volume 74.
9. Gnana, D.A.A.; Balamurugan, S.A.A.; Leavline, E.J. Literature review on feature selection methods for high-dimensional data. *Int. J. Comput. Appl.* **2016**, *975*, 8887.
10. Hussien, A.G.; Hassanien, A.E.; Houssein, E.H.; Bhattacharyya, S.; Amin, M. S-shaped binary whale optimization algorithm for feature selection. In *Recent Trends in Signal and Image Processing*; Springer: Berlin/Heidelberg, Germany, 2019; pp. 79–87.
11. Dhiman, G.; Kaur, A. Optimizing the design of airfoil and optical buffer problems using spotted hyena optimizer. *Designs* **2018**, *2*, 28.
12. Oh, I.S.; Lee, J.S.; Moon, B.R. Hybrid genetic algorithms for feature selection. *IEEE Trans. Pattern Anal. Mach. Intell.* **2004**, *26*, 1424–1437.
13. Kennedy, J.; Eberhart, R. Particle Swarm Optimisation. In Proceedings of the ICNN'95 - International Conference on Neural Networks, Perth, WA, Australia, 27 November–1 December 1995.
14. Kennedy, J.; Eberhart, R.C. A discrete binary version of the particle swarm algorithm. In Proceedings of the 1997 IEEE International Conference on Systems, Man, and Cybernetics. Computational Cybernetics and Simulation, Orlando, FL, USA, 12–15 October 1997; Volume 5, pp. 4104–4108.
15. Chakraborty, B. Feature subset selection by particle swarm optimization with fuzzy fitness function. In Proceedings of the 2008 3rd International Conference on Intelligent System and Knowledge Engineering, Xiamen, China, 17–19 November 2008; Volume 1, pp. 1038–1042.
16. Wang, X.; Yang, J.; Teng, X.; Xia, W.; Jensen, R. Feature selection based on rough sets and particle swarm optimization. *Pattern Recognit. Lett.* **2007**, *28*, 459–471.
17. Xue, B.; Zhang, M.; Browne, W.N. Particle swarm optimisation for feature selection in classification: Novel initialisation and updating mechanisms. *Appl. Soft Comput.* **2014**, *18*, 261–276.
18. Aghdam, M.H.; Ghasem-Aghaee, N.; Basiri, M.E. Text feature selection using ant colony optimization. *Expert Syst. Appl.* **2009**, *36*, 6843–6853.
19. Karaboga, D. *An Idea Based on Honey Bee Swarm for Numerical Optimization*; Technical Report; Citeseer: 2005.
20. Rashedi, E.; Nezamabadi-Pour, H.; Saryazdi, S. BGSA: Binary gravitational search algorithm. *Nat. Comput.* **2010**, *9*, 727–745.
21. Wang, J.; Hedar, A.R.; Wang, S.; Ma, J. Rough set and scatter search metaheuristic based feature selection for credit scoring. *Expert Syst. Appl.* **2012**, *39*, 6123–6128.
22. Hashim, F.A.; Hussain, K.; Houssein, E.H.; Mabrouk, M.S.; Al-Atabany, W. Archimedes optimization algorithm: A new metaheuristic algorithm for solving optimization problems. *Appl. Intell.* **2021**, *51*, 1531–1551.
23. Van Beek, P. Backtracking search algorithms. In *Foundations of Artificial Intelligence*; Elsevier: Amsterdam, The Netherlands, 2006; Volume 2, pp. 85–134.
24. Abd Elminaam, D.S.; Nabil, A.; Ibraheem, S.A.; Houssein, E.H. An efficient marine predators algorithm for feature selection. *IEEE Access* **2021**, *9*, 60136–60153.
25. Mirjalili, S.; Lewis, A. The whale optimization algorithm. *Adv. Eng. Softw.* **2016**, *95*, 51–67.
26. Sharma, M.; Kaur, P. A Comprehensive Analysis of Nature-Inspired Meta-Heuristic Techniques for Feature Selection Problem. *Arch. Comput. Methods Eng.* **2021**, *28*, 1103–1127.

27. Xue, Y.; Xue, B.; Zl, M. Self-Adaptive particle swarm optimization for large-scale feature selection in classification. *Acm Trans. Knowl. Discov. Data* **2019**, *13*, 50. <https://doi.org/10.1145/3340848>.
28. Askari, Q.; Saeed, M.; Younas, I. Heap-based optimizer inspired by corporate rank hierarchy for global optimization. *Expert Syst. Appl.* **2020**, *161*, 113702.
29. AbdElminaam, D.S.; Houssein, E.H.; Said, M.; Oliva, D.; Nabil, A. An efficient heap-based optimizer for parameters identification of modified photovoltaic models. *Ain Shams Eng. J.* **2022**, *13*, 101728.
30. Elsayed, S.K.; Kamel, S.; Selim, A.; Ahmed, M. An improved heap-based optimizer for optimal reactive power dispatch. *IEEE Access* **2021**, *9*, 58319–58336.
31. Zarshenas, A.; Suzuki, K. Binary coordinate ascent: An efficient optimization technique for feature subset selection for machine learning. *Knowl.-Based Syst.* **2016**, *110*, 191–201.
32. Chuang, L.Y.; Tsai, S.W.; Yang, C.H. Improved binary particle swarm optimization using catfish effect for feature selection. *Expert Syst. Appl.* **2011**, *38*, 12699–12707.
33. Zhang, Y.; Wang, S.; Phillips, P.; Ji, G. Binary PSO with mutation operator for feature selection using decision tree applied to spam detection. *Knowl.-Based Syst.* **2014**, *64*, 22–31.
34. Mafarja, M.; Aljarah, I.; Faris, H.; Hammouri, A.I.; Ala'M, A.Z.; Mirjalili, S. Binary grasshopper optimisation algorithm approaches for feature selection problems. *Expert Syst. Appl.* **2019**, *117*, 267–286.
35. Emary, E.; Zawbaa, H.M.; Hassanien, A.E. Binary ant lion approaches for feature selection. *Neurocomputing* **2016**, *213*, 54–65.
36. Asgarnezhad, R.; Monadjemi, S.A.; Soltanaghaei, M. An application of MOGW optimization for feature selection in text classification. *J. Supercomput.* **2021**, *77*, 5806–5839.
37. Neggaz, N.; Ewees, A.A.; Abd Elaziz, M.; Mafarja, M. Boosting salp swarm algorithm by sine cosine algorithm and disrupt operator for feature selection. *Expert Syst. Appl.* **2020**, *145*, 113103.
38. Kumar, V.; Kaur, A. Binary spotted hyena optimizer and its application to feature selection. *J. Ambient. Intell. Humaniz. Comput.* **2020**, *11*, 2625–2645.
39. Nakamura, R.Y.; Pereira, L.A.; Costa, K.A.; Rodrigues, D.; Papa, J.P.; Yang, X.S. BBA: A binary bat algorithm for feature selection. In Proceedings of the 2012 25th SIBGRAPI Conference on Graphics, Patterns and Images, Ouro Preto, Brazil, 22–25 August 2012; pp. 291–297.
40. Mafarja, M.M.; Eleyan, D.; Jaber, I.; Hammouri, A.; Mirjalili, S. Binary dragonfly algorithm for feature selection. In Proceedings of the 2017 International Conference on New Trends in Computing Sciences (ICTCS), Amman, Jordan, 11–13 October 2017; pp. 12–17.
41. Neggaz, N.; Houssein, E.H.; Hussain, K. An efficient henry gas solubility optimization for feature selection. *Expert Syst. Appl.* **2020**, *152*, 113364.
42. Jiang, Y.; Luo, Q.; Wei, Y.; Abualigah, L.; Zhou, Y. An efficient binary Gradient-based optimizer for feature selection. *Math. Biosci. Eng.* **2021**, *18*, 3813–3854.
43. Ouadfel, S.; Abd Elaziz, M. Enhanced crow search algorithm for feature selection. *Expert Syst. Appl.* **2020**, *159*, 113572.
44. Chaudhuri, A.; Sahu, T.P. Feature selection using Binary Crow Search Algorithm with time varying flight length. *Expert Syst. Appl.* **2021**, *168*, 114288.
45. Too, J.; Mirjalili, S. General learning equilibrium optimizer: A new feature selection method for biological data classification. *Appl. Artif. Intell.* **2021**, *35*, 247–263.
46. Hamidzadeh, J.; Kelidari, M. Feature selection by using chaotic cuckoo optimization algorithm with levy flight, opposition-based learning and disruption operator. *Soft Comput.* **2021**, *25*, 2911–2933.
47. Sayed, S.A.F.; Nabil, E.; Badr, A. A binary clonal flower pollination algorithm for feature selection. *Pattern Recognit. Lett.* **2016**, *77*, 21–27.
48. Moorthy, U.; Gandhi, U.D. Forest optimization algorithm-based feature selection using classifier ensemble. *Comput. Intell.* **2020**, *36*, 1445–1462.
49. Hodashinsky, I.; Sarin, K.; Shelupanov, A.; Slezkin, A. Feature selection based on swallow swarm optimization for fuzzy classification. *Symmetry* **2019**, *11*, 1423.
50. Ghosh, M.; Guha, R.; Alam, I.; Lohariwal, P.; Jalan, D.; Sarkar, R. Binary genetic swarm optimization: A combination of GA and PSO for feature selection. *J. Intell. Syst.* **2019**, *29*, 1598–1610.
51. Liu, M.K.; Tran, M.Q.; Weng, P.Y. Fusion of vibration and current signatures for the fault diagnosis of induction machines. *Shock Vib.* **2019**, *2019*, 7176482.
52. Tran, M.Q.; Elsis, M.; Liu, M.K. Effective feature selection with fuzzy entropy and similarity classifier for chatter vibration diagnosis. *Measurement* **2021**, *184*, 109962.
53. Tran, M.Q.; Li, Y.C.; Lan, C.Y.; Liu, M.K. Wind Farm Fault Detection by Monitoring Wind Speed in the Wake Region. *Energies* **2020**, *13*, 6559.
54. Aljarah, I.; Habib, M.; Faris, H.; Al-Madi, N.; Heidari, A.A.; Mafarja, M.; Elaziz, M.A.; Mirjalili, S. A dynamic locality multi-objective salp swarm algorithm for feature selection. *Comput. Ind. Eng.* **2020**, *147*, 106628.
55. Alweshah, M.; Khalailah, S.A.; Gupta, B.B.; Almomani, A.; Hammouri, A.I.; Al-Betar, M.A. The monarch butterfly optimization algorithm for solving feature selection problems. *Neural Comput. Appl.* **2020**. <https://doi.org/10.1007/s00521-020-05210-0>.
56. Arora, S.; Anand, P. Binary butterfly optimization approaches for feature selection. *Expert Syst. Appl.* **2019**, *116*, 147–160.

57. Gao, Y.; Zhou, Y.; Luo, Q. An Efficient Binary Equilibrium Optimizer Algorithm for Feature Selection. *IEEE Access* **2020**, *8*, 140936–140963.
58. Ghosh, K.K.; Singh, P.K.; Hong, J.; Geem, Z.W.; Sarkar, R. Binary social mimic optimization algorithm with X-shaped transfer function for feature selection. *IEEE Access* **2020**, *8*, 97890–97906.
59. Ghosh, K.K.; Ahmed, S.; Singh, P.K.; Geem, Z.W.; Sarkar, R. Improved binary sailfish optimizer based on adaptive β -Hill climbing for feature selection. *IEEE Access* **2020**, *8*, 83548–83560.
60. Guha, R.; Ghosh, M.; Kapri, S.; Shaw, S.; Mutsuddi, S.; Bhateja, V.; Sarkar, R. Deluge based Genetic Algorithm for feature selection. *Evol. Intell.* **2021**, *14*, 357–367.
61. Hammouri, A.I.; Mafarja, M.; Al-Betar, M.A.; Awadallah, M.A.; Abu-Doush, I. An improved Dragonfly Algorithm for feature selection. *Knowl.-Based Syst.* **2020**, *203*, 106131.
62. Han, C.; Zhou, G.; Zhou, Y. Binary Symbiotic Organism Search Algorithm for Feature Selection and Analysis. *IEEE Access* **2019**, *7*, 166833–166859.
63. Nouri-Moghaddam, B.; Ghazanfari, M.; Fathian, M. A novel multi-objective forest optimization algorithm for wrapper feature selection. *Expert Syst. Appl.* **2021**, *175*, 114737.
64. Yan, C.; Ma, J.; Luo, H.; Patel, A. Hybrid binary Coral Reefs Optimization algorithm with Simulated Annealing for Feature Selection in high-dimensional biomedical datasets. *Chemom. Intell. Lab. Syst.* **2019**, *184*, 102–111.
65. Alweshah, M.; Alkhalaileh, S.; Albashish, D.; Mafarja, M.; Bsoul, Q.; Dorgham, O. A hybrid mine blast algorithm for feature selection problems. *Soft Comput.* **2021**, *25*, 517–534.
66. Anand, P.; Arora, S. A novel chaotic selfish herd optimizer for global optimization and feature selection. *Artif. Intell. Rev.* **2020**, *53*, 1441–1486.
67. Anter, A.M.; Ali, M. Feature selection strategy based on hybrid crow search optimization algorithm integrated with chaos theory and fuzzy c-means algorithm for medical diagnosis problems. *Soft Comput.* **2020**, *24*, 1565–1584.
68. Qasim, O.S.; Al-Thanoon, N.A.; Algamal, Z.Y. Feature selection based on chaotic binary black hole algorithm for data classification. *Chemom. Intell. Lab. Syst.* **2020**, *204*, 104104.
69. Ahmady, G.A.; Mehrpour, M.; Nikooravesh, A. Organizational structure. *Procedia-Soc. Behav. Sci.* **2016**, *230*, 455–462.
70. Dheeru, D.; Karra Taniskidou, E. *UCI Machine Learning Repository*; Irvine, School of Information and Computer Sciences, University of California: Irvine, CA, USA, 2017.
71. Jiao, Y.; Du, P. Performance measures in evaluating machine learning based bioinformatics predictors for classifications. *Quant. Biol.* **2016**, *4*, 320–330.