Engineering Science and Technology, an International Journal 44 (2023) 101453

HOSTED BY

ELSEVIER

Contents lists available at ScienceDirect

Engineering Science and Technology, an International Journal

journal homepage: www.elsevier.com/locate/jestch

A binary chaotic horse herd optimization algorithm for feature selection

Esin Ayşe Zaimoğlu [a,*], Nilüfer Yurtay [a], Hüseyin Demirci [a], Yüksel Yurtay [a]

[a] Computer and Information Sciences, Sakarya University, Sakarya 54187, Turkey

ARTICLE INFO

Article history:
Received 9 December 2022
Revised 26 April 2023
Accepted 26 May 2023
Available online 12 June 2023

Keywords:
Horse herd optimization
Feature selection
Metaheuristic algorithm
Machine learning
Artificial intelligence
Classification
Chaos map

ABSTRACT

One of the most challenging and common problems in machine learning is the Feature Selection (FS) process, which reduces the dataset size by finding optimal subsets of features. The Horse Herd Optimization Algorithm (HOA) is a new metaheuristic algorithm created by modeling the herd behavior of horses and developed for large scale optimization problems. This paper proposes the binary version of the HOA as a wrapper FS method to solve the FS problem. The proposed algorithm is a binary chaotic horse herd optimization algorithm for feature selection (BCHOAFS). The proposed BCHOAFS is applied to select the optimal feature combination that maximizes classification accuracy while minimizing the number of selected features. Classifier algorithms from machine learning algorithms were used to test the accuracy of the reduced subsets. The proposed method was named binary horse herd optimization for feature selection (BHOAFS) before adding chaotic maps; the k-nearest neighbor (k-NN) and Support Vector Machine (SVM) were tested as separate classifiers. It has been seen that k-NN classification accuracy gives better results than SVM. The BHOAFS-kNN method using the k-NN classification was combined with five chaotic maps and named as BCHOAFS-Logistics, BCHOAFS-Piecewise, BCHOAFS-Singer, BCHOAFS-Sinusoidal, BCHOAFS-Tent. The BCHOAFS versions were run on datasets consisting of 18 different sizes and quality datasets (i.e., low scale, medium scale, and large scale) taken from the UCI repository and compared with state-of-the-art algorithms in previous studies. The results prove that the proposed version, especially with the BCHOAFS-Piecewise and the BCHOAFS-Singer chaotic map outperforms or competes with well-known methods. The proof of the proposed approach's statistical significance has been validated using the Friedman Signed Rank test and post hoc Wilcoxon test. The novelty of BCHOAFS is that HOA, which is an optimization algorithm specially designed for large scale data, is the first binary chaotic-based algorithm developed for feature selection problems. It also proposes a new local search strategy called Similarity Measurement Function (SMF). As a result, versions of the proposed algorithm BCHOAFS can be used for the FS problem.

1. Introduction

With the development of technology, a wide variety of real-world data has emerged in the fields of business, science, health, and engineering [1]. Information in growing raw data is unnecessary unless patterns are extracted. Therefore, feature selection (FS) algorithms have advantages such as eliminating irrelevant, redundant, or noisy data, improving the learning algorithm's performance, reducing the computational cost, providing a better understanding of datasets, and reducing the size of the data space requirements [2]. Classification accuracy is greatly affected by the quality of the input features used to construct a learned model. FS is an essential preprocessing task that aims to select the most effective subset of features from the original dataset. Therefore, the FS methods are necessary to increase the classification accuracy and reduce the complexity of the constructed model.

FS algorithms have been used in many fields, such as face recognition [3], text classification [4], cancer and gene classification [5], text categorization, medical diagnostic decision support systems [6], and network intrusion detection systems [7]. They use random or heuristic search strategies to find the optimal subset of features, reducing the computational cost. FS methods are divided into three main groups, filter, wrapper, and embedded [8]. The filter approach selects attributes using statistical methods without using any learning method. The filter approach's computational cost is low, but its performance is not at the desired level. There are many filtering methods, such as chi-square statistics, t statistics, information gain, minimum redundancy and maximum relevance, Relief, common information, and correlation scoring [9]. The wrapper ap-

* Corresponding author.
E-mail addresses: esinzaimoglu@sakarya.edu.tr (E.A. Zaimoğlu), nyurtay@sakarya.edu.tr (N. Yurtay), huseyind@sakarya.edu.tr (H. Demirci), yyurtay@sakarya.edu.tr (Y. Yurtay).

https://doi.org/10.1016/j.jestch.2023.101453
2215-0986/

proach selects the candidate feature set using a learning algorithm (classification, clustering, etc.) and operates on this set. Although the wrapper approach gives faster and better results than the filter approach, the computational costs are higher [10]. The wrapper-based techniques for classification algorithms have generally outperformed filter-based methods in literature [11]. The classification algorithms used in wrapper-based FS methods are Decision Trees, k-Nearest Neighbor (k-NN), ANN, and Support Vector Machines (SVM) [12,13]. The third FS approach is hybrid algorithms. To reduce the feature size in hybrid algorithms, a filtering method is applied to the data first, then the most suitable subset is found with the help of a wrapper among the remaining features [14]. The hybrid approach handles the filter and wrapper models as two separate steps. Since the hybrid approach combines the two models, the accuracy may be low [15]. Searching for a solution in a search space is a Non-polynomial Hard (NP-Hard) problem. NP-Hard problems are accepted as multi-objective optimization problems. In recent years, it has been seen that metaheuristic algorithms are used successfully in solving many complex optimization problems [16,17]. Also, solving the FS problem with metaheuristic methods has become a popular research topic with the improved computer performances and the emergence of newly developed metaheuristic algorithms. Therefore, research on applying metaheuristic algorithms to the FS problem has increased considerably, including theoretical and practical studies [18]. Well-known metaheuristic algorithms commonly used in the FS problem can be listed as Particle Swarm Optimization(PSO) [15], Genetic Algorithm (GA) [19], Gravitational Search Algorithm [20], Gray Wolf Optimizer (GWO) [21], Whale Optimization Algorithm (WOA) [22], Krill Herd Algorithm [23], Ant Colony Optimization Algorithm [24], Bat Algorithm [25], Grasshopper Algorithm [26], and Aquila Optimizer [27]. In addition to using well-known metaheuristic optimization algorithms in the FS problem, new optimization algorithms proposed in recent years have also been applied. Some of these studies can be summarized as follows, Tubishat et al. [28,29] have used Salp Swarm Algorithm (SSA) for FS as a two-way search strategy, then improved with the Singer Chaotic Map. The designed SSA showed a significant improvement over competing algorithms in statistical analysis. Sheth et al. [6] proposed a new multi-purpose FS method using the Jaya Optimization Algorithm for five datasets: lung cancer, breast cancer, diabetes, fertility, and immunotherapy. In another study, single and multi-objective versions of Artificial Butterfly Optimization were applied to the FS problem and tested with well-known algorithms [30]. Integrated Artificial Immune System and Artificial Bee Colony (ABC)-based breast cancer diagnosis (IAIS-ABC-CDS) has been proposed by Al- Turjman et al. for parallel processing of effective FS and parameter optimization in ANN [31]. A binary version of the Social Spider Algorithm (BinSSA) with a cross operator (BinSSA-CR) has been proposed by Ülker et al. [32] and compared with a BinSSA without a cross operator. It was observed that BinSSA-CR gave better results than BinSSA. Two new FS approaches based on PSO with a crossover operator and a feature vector using the k-NN algorithm are proposed by Hichem et al. [33]. In a study proposed by Dasgupta et al. [34], classification problems were solved with Random Forest, k-NN, and SVM algorithms using the Penguin Search Optimization algorithm, which was inspired by the random foraging behavior of penguins. Performance of the two multi-objective ABC algorithms (Bin-MOABC and Num-MOABC) proposed by Hancer et al. [35] have been compared with single-target ABC algorithms (ABC-ER and ABC-Fit2C), conventional algorithms, and multi-objective algorithms (i.e., NSGAII, NSSABC, and MOPSO). A deep learning-based approach has been used for FS that can be used in the early diagnosis of COVID-19 disease [36]. It has been proven that the proposed approach can assist professionals during COVID-19 diagnostic studies. The new optimization algorithms proposed in recent years have produced remarkable results by using them together with classical machine learning methods in the FS problem. A binary variant of the recently proposed Sailfish Optimization called Binary Sailfish [37], and Spider Monkey Optimization algorithms have been used for FS and feature reduction problems [7]. The Adaptive Genetics Algorithm developed for the reduction of DNA microarray data used SVM and NB classifier for performance measurement [5]. A new Chaotic Dragonfly Algorithm was proposed, in which chaotic maps were applied to the search parameters of the Dragonfly Algorithm [38]. A wrapper-based binary Sine–Cosine Algorithm has been proposed for FS named WBSCA [39]. In a study using a binary version of the recently released Harris Hawk Optimization Algorithm, the k-NN method was used as a wrapper classifier to generate optimal feature subsets [40]. In a study published in 2021, the proposed Dispersed Foraging Slime Mould Algorithm(DFSMA), developed from the Slimy Mold Algorithm, a swarm-based stochastic algorithm, is introduced. The proposed algorithm is compared with eleven metaheuristic algorithms, ten improved algorithms, and three different algorithms introduced recently. In addition, binary-DFSMA (BDFSMA) was obtained using the transform function, and the performance of BDFSMA was evaluated on 12 datasets in the UCI repository. Experimental results revealed that BDFSMA outperforms the original SMA, improves classification accuracy, and reduces the number of selected features compared to other optimization algorithms [41]. In another study designed based on the Forest Optimization Algorithm, two versions were developed for the FS algorithm using continuous and binary representations. According to the results, both proposed algorithms achieved the same performance as other multi-objective methods and showed even better performance than single-objective methods [42]. In a study published in 2022, a wrapper-based greedy crossover multi-objective Binary Bat Algorithm (BBA) is proposed to reset the sub-optimal solutions obtained due to early convergence. The selected features were evaluated using SVM with 10-fold cross-validation, with benchmark datasets available in the UCI repository[43].

Also, hybrid versions of metaheuristic algorithms have been developed in recent years for solving FS. It is a well-known fact in the literature that the efficiency of high-dimensional datasets decreases, and computational costs increase in wrapper meta-heuristic methods. To solve this problem, a study published in 2022 proposed a hybrid FS approach based on the Relief filter method and a new meta-heuristic Equilibrium Optimizer. An interactive filter-wrapper multi-objective evolutionary algorithm named GR-MOEA was proposed in 2021. Unlike other algorithms, the algorithm also developed two populations (filter population and wrapper population) simultaneously and interacted with each other during evolution to obtain a higher-quality final feature subset. Comparison results on different datasets demonstrated the superiority of GR-MOEA over the state-of-the-art available in terms of accuracy, and the number of features selected [11,12]. Another algorithm (HBCRO-BPSO), developed based on Binary Chemical Reaction Optimization (BCRO) and Binary Particle Swarm Optimization (BPSO), used the k-NN classifier to optimize the number of features and improve the classification accuracy [44]. In another study [45], hybrid-intelligent phishing website prediction using Deep Neural Networks with evolutionary algorithm-based FS and weighting methods has been proposed to improve phishing website prediction. In the proposed approach, the optimum weights of the features are heuristically defined using GA to help improve the accuracy of the phishing website prediction. Another study published in 2021 proposes a new hybrid filter wrapper feature. The proposed algorithm using the Multi-objective WOA, in which the filter and wrapper fitness functions are optimized simultaneously, has been tested with seven well-known algorithms. Experimental results show the ability of the proposed algorithm to obtain several subsets with fewer features with excellent classification accuracy [46].

When the studies above are examined in detail, it is seen that the possibility of developing more efficient techniques for solving the FS problem is still possible. Even though the selection of the best subset of features with the best performance has been greatly developed, studies with newly published optimization techniques are still open to improvement. In recent years, studies on the discovery and selection process of the FS problem have encouraged our efforts to present an improved version of the Horse Herd Optimization Algorithm (HOA) [47] to solve this problem. In this study, a binary horse herd optimization algorithm, which is a binary version of HOA, was applied to the FS problem to select the appropriate feature subset. The recommended method was called BHOAFS before adding chaotic maps and tested as the k-nearest neighbor (k-NN) and Support Vector Machine (SVM). The experimental results showed that BHOAFS-kNN gave better results than BHOAFS-SVM. Binary Chaotic Horse Herd Optimization Algorithm For Feature Selection (BCHOAFS) is proposed by adding five one-dimensional chaotic maps (Logistic, Tent, Piecewise, Singer, and Sinusoidal) to BHOAFS-kNN. The proposed chaotic search-based BCHOAFS by adding chaotic maps is adopted to maximize the multi-objective classification accuracy, select the most suitable features that minimize the features of the datasets, and increase the algorithm's stability. The performance of the proposed FS-based BCHOAFS methods was evaluated using 18 different benchmark datasets. The results show that the proposed BCHOAFS achieves better results than other similar algorithms (i.e., proposed BHOAFS-kNN, BHOAFS-SVM, and previous studies published in the literature).

Also, the main contributions of this article are as follows:

(a) The original HOA is a special algorithm that can also work effectively on large scale data. In the study, the HOA is converted based on the binary search field. Then, the binary horse herd algorithm was applied to the feature selection problem with the wrapper method (BHOAFS). A detailed literature study was conducted to convert the HOA to a binary version and apply it to the feature selection problem;
(b) Thanks to Age Determination, it is ensured that the horses do a global search in the search space according to their solution. As a local search strategy, the SMF operator was developed to increase both the exploitation and exploration capabilities of BHOAFS;
(c) In previous feature selection studies in the literature, low or medium-sized data sets were generally used for testing. Large scale datasets were also tested in this study. The CNAE dataset with a maximum of 856 features was used;
(d) In this study, while trying to maximize the classification accuracy, it was aimed to minimize the number of selected features. In this context, many criteria were evaluated (e.g., average fitness values, average accuracy values, average number of selected features, and CPU time).
(e) Friedman Signed Rank test and post hoc Wilcoxon test was used to prove that the results were significant.
(f) Initial values in feature selection problems in the literature are usually determined randomly. To increase the algorithm's stability, chaotic maps are used as Random Number Generators. The proposed algorithm was retested on 18 UCI data sets by adding five different chaotic maps;
(g) The most important contribution of this work is the proposal of the BCHOAFS algorithm to increase BHOAFS stability. In addition, although optimization-based feature selection algorithms have been proposed before, the version in which a binary chaotic HOA algorithm specially designed for large scale data is used has not been proposed as far as we know. The rest of this paper is organized as follows: The foundations of the research and the proposed binary BHOAFS method are given in Section 2. The proposed FS-based binary chaotic BCHOAFS methods are described in Section 3. Experimental results and discussions are given in Section 4. Finally, the results and conclusions are presented in Section 5.

2. Material and methods

2.1. Horse herd optimization

The HOA is an optimization algorithm that was first created in 2021 by imitating the social performance of horses [48]. Horses exhibit different behaviors at different ages. The lifespan of a horse is approximately 25–30 years. Horse ages are categorized as 0–5, 5–10, 10–15, and older than 15. Horses are grouped with the following names according to various age ranges.

- δ horses $\rightarrow$ horses between the ages of 0–5
- γ horses $\rightarrow$ horses between the ages of 5–10
- β horses $\rightarrow$ horses between the ages of 10–15
- α horses $\rightarrow$ symbolizes horses older than 15 years

When horses are ranked from best to worst according to their age, the top 10% are selected as α horses. The following 20% are in the group of β horses. δ and γ horses represent 30% and 40% of the remaining horses, respectively. The algorithm is based on horses' behavior patterns in their environments and their social performance at different ages. Behavior patterns of horses generally include Grazing (G), Hierarchy (H), Sociability (S), Imitation (I), Defense mechanism (D), and Circulation (R) [48].

These six behaviors represent aging in horses. These behaviors are described below.

- Grazing: It is the behavior in which horses spend 16 −20 h a day moving freely.
- Hierarchy: Some horses living in herds are leaders, and the rest is followers. Strong horses are responsible for the horses in the herd.
- Sociability: It is the behavior in which horses live together and are in contact with other animals. Some horses live in groups, while others fight with other animals.
- Imitation: Horses imitate each other. In general, young horses need to find the right position of pasture, defense mechanism, etc. They learn such behaviors from adult horses. These behaviors allow them to survive safely.
- Defense: The defense mechanism of horses is in 2 ways: Horses either flee from the attack or live in harmony with the aggressor.
- Roaming: Horses explore new places out of curiosity and in search of nearby pastures.

The fitness value is calculated according to the positions of the horses in the herd. Since the fitness value represents the age of the horses, their behavior is updated with each iteration in order from best to worst. As a result, the first 10% of the horses in the ranked population with good fitness values are selected as α horses. The following 20% β horses, γ, and δ horses are made up of 30% and 40% of the remaining horses, respectively. Therefore, this algorithm is inspired by the X general behaviors of horses at different ages. The movement applied to the horses at each iteration is according to Eq. 1.

$$X_{mIter,AGE} = V_{mIter,AGE} + X_{m(Iter-1),AGE} \quad AGE = \alpha, \beta, \delta, \gamma \tag{1}$$

The following mathematical equations are used to calculate the velocity vector of horses at different ages with the above six behavior patterns.

$$\overrightarrow{V_m^{t,\alpha}} = \overrightarrow{G_m^{t,\alpha}} + \overrightarrow{D_m^{t,\alpha}} \tag{2}$$

$$\overrightarrow{V_m^{t,\beta}} = \overrightarrow{G_m^{t,\beta}} + \overrightarrow{D_m^{t,\beta}} + \overrightarrow{H_m^{t,\beta}} + \overrightarrow{S_m^{t,\beta}} \tag{3}$$

$$\overrightarrow{V_m^{t,\delta}} = \overrightarrow{G_m^{t,\delta}} + \overrightarrow{D_m^{t,\delta}} + \overrightarrow{H_m^{t,\delta}} + \overrightarrow{S_m^{t,\delta}} + \overrightarrow{I_m^{t,\delta}} + \overrightarrow{R_m^{t,\delta}} \tag{4}$$

$$\overrightarrow{V_m^{t,\gamma}} = \overrightarrow{G_m^{t,\gamma}} + \overrightarrow{I_m^{t,\gamma}} + \overrightarrow{R_m^{t,\gamma}} \tag{5}$$

2.2. Binary-Based Horse Herd Optimization Algorithm (BHOAFS)

S-shaped (sigmoid), V-shaped [49,50], and U-shaped [51] transfer functions are commonly used transfer functions. The feature selection problem works in a discrete space, while HOA works in a continuous space. To apply the feature selection problem to HOA, continuous space must be converted to discrete space. This transformation is done by using the U-Shaped Transfer function in the proposed algorithm. U-shape transfer function: δ is the slope of the transfer function, and α is the width; the transfer function is calculated as in Eq. 6.

$$U\left(X_{m,j}^t\right) = \delta|X_{m,j}^t|^\alpha \tag{6}$$

$$X_{m,j}^t = \begin{cases} 1 & rand < U\left(X_{m,j}^t\right), \\ 0 & rand \geqslant U\left(X_{m,j}^t\right). \end{cases} \tag{7}$$

where $U\left(X_{m,j}^t\right)$identifies the probability value of the U-shape. The values of the elements are changed to 1's or 0's using Eq. 7.

The following equation Eq. 8 calculates the movement of horses in BHOAFS in the search space represented as binary.

$$x_i^{k+1} = x_i^k + \theta * \left(X_{BestHorse}^k - X_i^k\right) \quad i = 1,2,\ldots,N \tag{8}$$

where x_i^k and $x_i^{(k+1)}k+1$ are the positions of the ith horse in iterations k and $k+1$, respectively. $x_BestHorse^k$ is the position of the best horse in the search space. θ is a random number evenly distributed between 0 and 1, and N is the number of horses [52]. The process of simulating the BHOAFS begins mathematically by initially creating a population of randomly generated horses. In this research area, each $X_d = (X_{d1}, X_{d2}, X_{d3}, \ldots, X_{dn})$ n represents the horse in position.$d = 1,2,\ldots,N, j = 1,2,\ldots,D, U(0,1)$ is a function that generates a random value evenly distributed between 0 and 1, X_j^{min} ve X_j^{max} as the boundaries of the search space in the j position;

$$X_{d,j} = X_j^{min} + \left(X_j^{max} - X_j^{min}\right) * U(0,1) \tag{9}$$

$X_{d,j}$ is expressed as in Eq. 9.

Solutions in BHOAFS are updated in the discrete search space. In BHOAFS, the FS is represented as a binary vector with values of 0 or 1. A binary bit string of length N for an N-dimensional data set represents the position of the horses. While each bit represents a feature, a value of "1" means that feature is selected, and "0" means that feature is not selected. Each position vector generated is a subset of features generated from the original dataset. A balance must be struck between exploration and exploitation to improve algorithm performance and obtain the best solution [53]. $X_i, X_2, X_3, X_4 \ldots X_n$ represents a vector array representing the position of each X horse, $x_i^j X^j$ represents the ith bit of X_j horse. x_i^j is defined as in Eq. 7.

Eq. 7 is used better to present the binary representation of the search space. The solution for X_i is represented as $BinaryX_i = [1,0,0,1,1,0,1]$ where $X_i = [0.59, 0.170.32, 0.95, 0.85, 0.50, 0.77]$. If we explain this representation of $BinaryX_i$ more clearly, a new feature subset is created by selecting the first, fourth, fifth, and seventh features from the 1's and 0's array. However, the second, third, and sixth features should not be selected. The second, third, and sixth features must be eliminated.

2.3. Position update strategy (age determination)

The Hamming distance between two strings of equal length is the number of positions where the corresponding symbols differ. Therefore, the Hamming distance can be used to measure the distance between two horses. X and Y are binary bit strings representing the position of two horses; The distance between the X and Y horses is equal to the result of the $X \oplus Y$ operation in Eq. 10 [54].

$$h(X,Y) = \sum_{i=1}^{N} (x_i \oplus y_i) \tag{10}$$

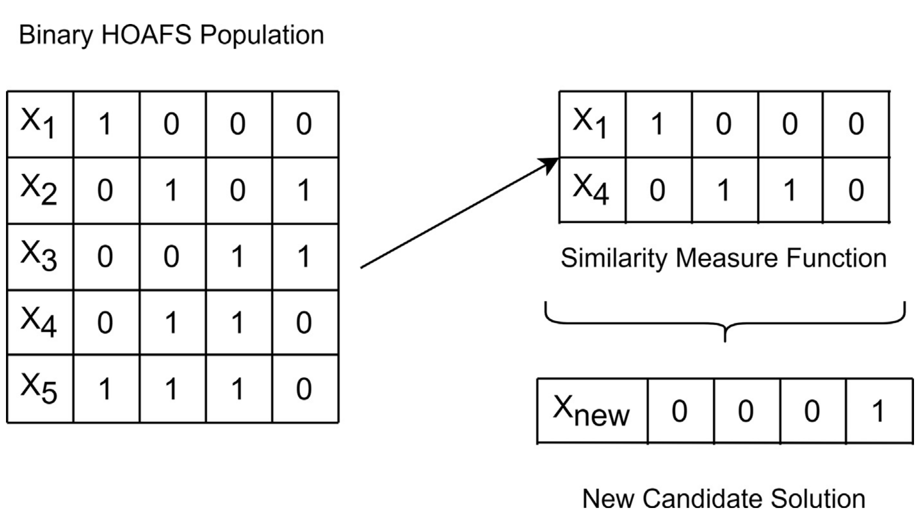

Fig. 1. Similarity Measure Function.

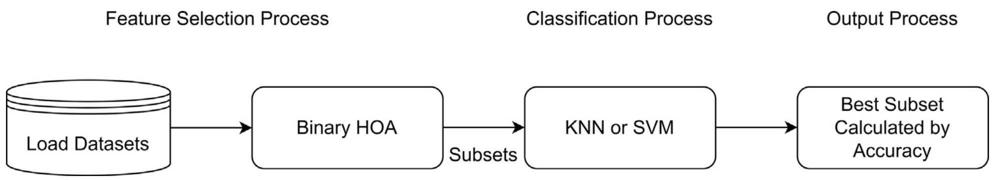

Fig. 2. Using k-NN and SVM Classifier.

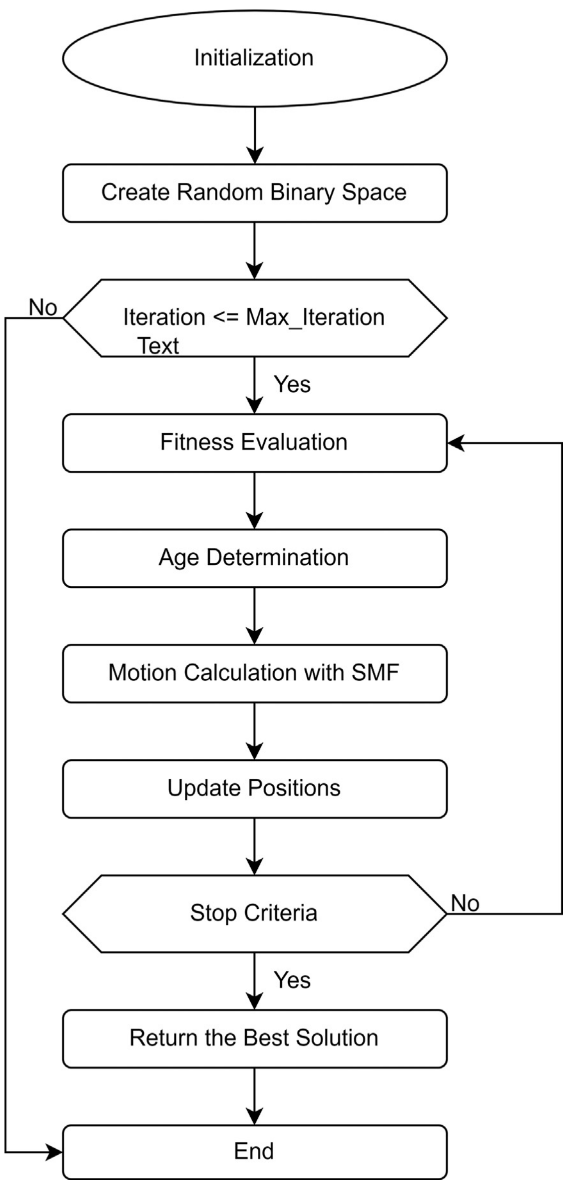

Fig. 3. Proposed Flowchart for BHOAFS.

$X^1, X^2, \ldots X^n$ is a vector array representing the position of each X horse, $x_i^j j$ to represent the ith bit of X^j i of the horse; X_c represents the cluster center of n horses and is represented by Eq. 11 below. The created cluster center is used to move horses in the search space.

$$X_c = \left\{ C_1, C_2, \ldots, C_N \right| \qquad If \frac{1}{n} \sum_{j=1}^{N} \left(X_i^j > 0.5 \right) \qquad C_i = 1$$
$$otherwise \quad C_i = 0 \} \tag{11}$$

Horses start the iteration by randomly positioning. Horses mature as they age and change their positions. The next solution in the search space changes according to each horse's age. Each horse performs the operations of the behavior (Eq. 1) mentioned above. Thanks to Age Determination, it is ensured that the horses do a global search in the search space according to their solution. The fitness function evaluates these behaviors and determines the next position. In the designed BHOAFS, each horse represents a possible solution to the FS problem. The accuracy of the classification algorithm obtained from this solution's horse position indicates the solution's quality. Each solution is associated with the population's quality, while horses offer local solutions based on their position. Horses mature as they age and change their positions. The next solution in the search space changes according to each horse's age. Alpha and Beta horses participate in the exploitation process, while Theta and Gamma horses participate in the exploration process.

2.3.1. Similarity measurement function (SMF)

The SMF, designed for the local search strategy for the proposed algorithm, aims to improve the current positions of the individuals with good positions in the herd and forward them to the next iteration. SMF is a function that measures the similarity/proximity of the two candidate horses that make up the population. Its main function is to calculate the distance between solutions. Then, according to this distance, it generates a candidate horse solution that is different from the existing solutions but is more likely to be in a better position.

If the distance of the two nearest or adjacent sources is less than a specified value of k, the lower fitness value from these sources is reconstructed. (According to the literature, the k value has been determined as $N/3$ or $N/2$, with the size of the dataset to be selected for FS. In the proposed algorithm, $N/3$ was used for cases where the dataset is high dimensional [55].) This process is done for the selected individuals in each iteration. The aim is to regenerate the available resource during the initialization of the population without losing the potential of the optimal point reached so far, not randomly. Thus, early convergence, the most crucial problem related to the herd, is prevented, and diversity is increased. The SMF process is shown in Fig. 1. In addition, this function provides a more comprehensive local search without reducing the global search capacity. Thus, the probability of finding a better solution by avoiding local pitfalls in the search space of BHOAFS is also increased. The balance of the proposed algorithm between exploration and exploitation is achieved with SMF.

2.3.2. Fitness function

In algorithms that convert from continuous search space to binary search space, the number of features in the subset should also be considered when calculating the fitness of a candidate solution to the objective function. However, since it is aimed to reduce the number of features by increasing the classification accuracy in the datasets, the fitness function is created using both the accuracy and the number of reduced features. Eq. 12 shows the fitness function.

The fitness function to be applied for the algorithm is expressed as in the Eq. 12 [56]:

Table 1
The five adapted chaotic maps in the paper.

Map	Definition (0,1)
C1 = Logistic Map (BCHOAFS1)	$c_{i+1} = wc_i \quad (1 - c_i)$
C2 = Tent Map (BCHOAFS2)	$c_{i+1} = c_i/0.7, \quad c_i < 0 \quad \left(\frac{10}{3}\right)(1 - c_i), \quad c_i \geqslant 0$
C3 = Piecewise Map (BCHOAFS3)	$c_{i+1} = c_i/p \quad 0 < c_i < p\frac{(c_i - p)}{(0.5 - p)}, \quad p < c_i < 0.5\frac{(1 - p - c_i)}{(0.5 - p)}, \quad 0.5 < c_i < 1 - p\frac{(1 - c_i)}{p}, \quad 1 - p < c_i < 1, \quad p = 0.2$
C4 = Singer Map (BCHOAFS4)	$c_{i+1} = \mu(7.86c_i - 23.31c_i^2 + 28.75c_i^3 - 13.302875c_i^4), \quad \mu = 1.07$
C5 = Sinusoidal Map (BCHOAFS5)	$c_{i+1} = wc_i^2 \sin(\pi c_i), \quad w = 2.3$

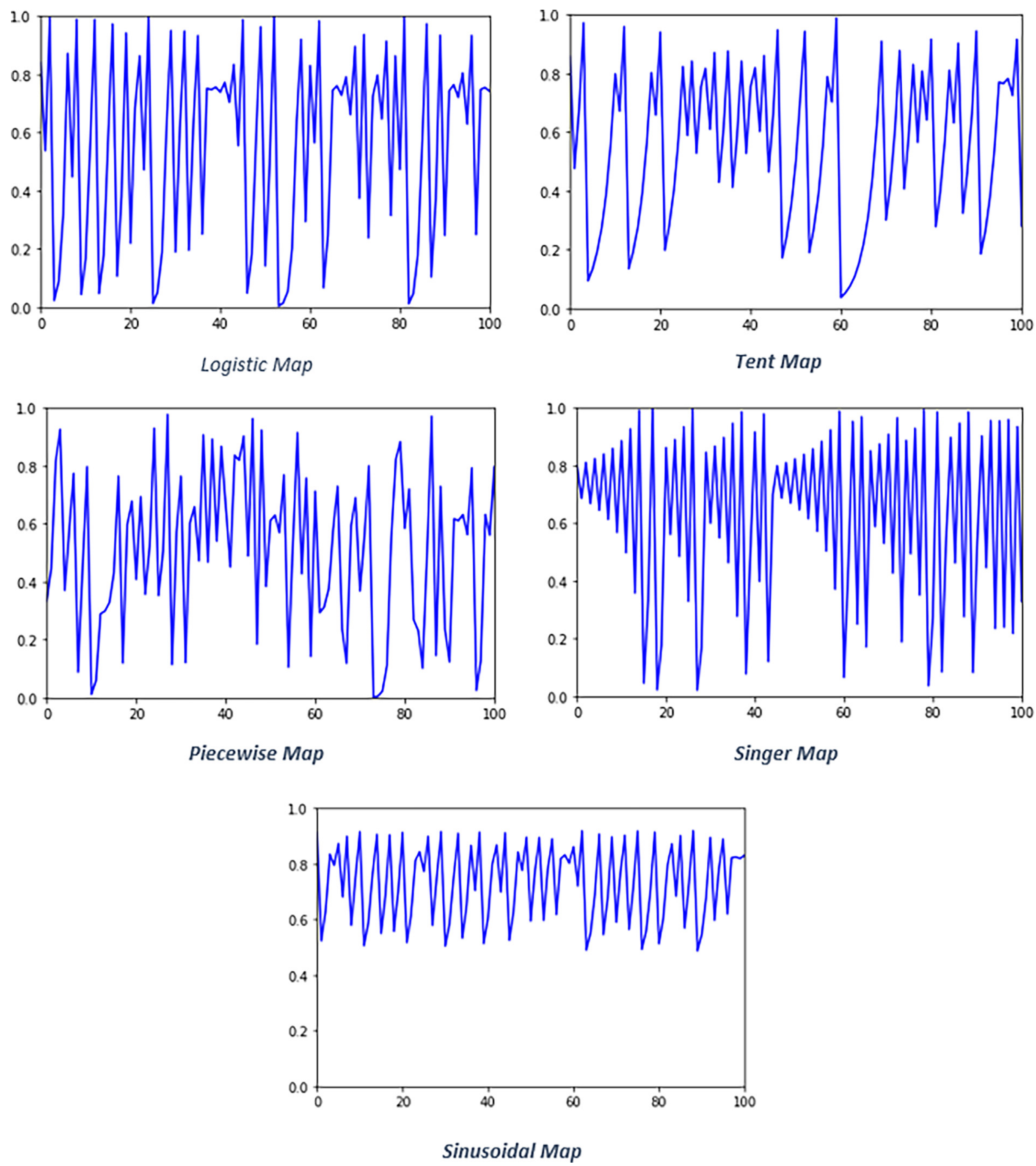

Fig. 4. Visualization of Chaotic Maps.

$$FitnessFunction = S * D + B * \frac{C - R}{C} \tag{12}$$

where S is a fixed value between [0,1], D is the accuracy value of the relevant classification algorithm, B is 1-S, C is the total number of features, and R is the number of selected features, respectively.

```
Start
Generate chaotic maps as C_map and Initialize populations with candidate horse positions X_i , i = 1,2…,n
Generate a population of n horse (X_n )
Calculate fitness value of each horse solution (Eq. 12)
Choose the horse which has the best fitness value as a best horse solution (X_best )
while (iter<= Max_iter ) do
        for each horse on population do
            Evaluate each horse X by a fitness function as Fitness (X_i) (Eq. 12)
            if Fitness (X_i) > = Fitness (X_best) and | X_i | < | X_best |
                    Set X_best = X_i and Fitness (X_best) = Fitness (X_i) and calculate Distance (Eq. 10)
            end if
            if Distance (X_best - X_i) < Distance (X_best)
                    Replace X with the new horse in search area
            end if
        end for
        for j = 1: numbers of horse
            for k = 1: numbers of horse
                    Calculate X_i^(k+1) = X_i^k + C_map * (X_best - X_i^k), i = 1,…n
                    if C_map <0.5
                            Set X^new = 0
                    else
                            Set X^new = 0
                    end if
            end for;
        end for;
        Set iter = iter +1
end while
return best solution
end
```

Fig. 5. The proposed BCHOAFS.

Table 2
Parameter setting for BCHOAFS.

Parameters	Values
Population size(n)	20
Maximum iteration	100
Number of runs	20
Problem dimension (N)	Number of features in the dataset
K	5 (K value in K-NN)
SVM	RBF Kernel
k	10 (k-value in k-fold)
Search domain	[0 1]

Table 3
Parameter setting for experiments.

Algorithms	Parameters	Values
PSO	w	2
	w_{max}	0.9
	w_{min}	0.4
	c1 and c2	2
GWO	α	[0 2]
ALO	Mutation rate	[0 0.9]
SSA	r_a, p_c and p_m	1, 0.7 and 0.1
	Cross_val and mut_val	0.9 and 0.1
	w	0.7
GA	Cross_val and mut_val	0.9 and 0.1
BA	Qmin and Qmax	0 and 2
	A and r	0.5 and 0.5
K-NN	K	5

2.3.3. BHOAFS fundamentals

In the wrapper-based FS algorithm problem, a classifier is needed to evaluate the selected feature. After determining which features to choose, a classification process is performed. Effective classification method selection can increase the prediction accuracy of the algorithm and reduce the required computational cost. KNN and SVM, which are widely used in the data mining community and machine learning, are the most effective classifier algorithms [57]. In the studies, different classifiers were used for different data types in feature selection algorithms [58–62]. In recent studies, there are algorithms that prefer the SVM if the dataset contains two classes and the k-NN classifier if it contains more than two classes [32].

Therefore, the proposed algorithm BHOAFS is tested separately on low, medium, and large scale well-known datasets in the UCI repository using k-NN and SVM. Thus, the most successful version of BHOAFS in terms of classification performance was tried to be determined.

k-NN Classifier: The data were classified with k = 5 for k-NN classification, and the training set results were obtained by 10-fold cross-validation. SVM Classifier: The Kernel Trick method calculates how similar each point is to a certain point by calculating with a normal distribution is called RBF Kernel. For SVM, the RBF Kernel method was used. For multi-class datasets, SVM was applied with a one-versus-all strategy. The basic representation of the BHOAFS is shown in Fig. 2.

The experimental results showed that the classification performance of k-NN was superior to SVM. The BHOAFS starts with the preprocessing step for the selected datasets. After the relevant data is appropriate, it is divided into two parts, the training set and the test set, and a subset of features is created from the original dataset, as in Fig. 2. The algorithm cycle continues by evaluating the selected classification algorithm and fitness function during the determined iteration. As in Fig. 3, when the stopping criterion of the algorithm is met, the most optimal features are accepted, and the reduced version of these features is applied to the test dataset. The size of the test dataset is reduced by the size of the subset produced by the training dataset. The reduced test set is also classified

Table 4
The UCI benchmark datasets details used for proposed BHOAFS.

Type	Dataset	#Features	#Samples	#Classes
Low Dimension	Iris	5	150	3
	Diabetes	9	768	2
	Wine	13	178	3
	Hepatitis	19	155	2
	Vehicle	18	846	4
	Zoo	16	101	7
	Heart	13	270	2
	Breast Cancer	11	699	2
Medium Dimension	Ionosphere	34	351	2
	Soybean Small	35	47	4
	Lung-Cancer	56	32	3
	Sonar	60	208	2
	BreastEW	30	569	2
	Movementlibras	90	360	15
	WaveformEW	40	5000	3
High Dimension	Arrhythmia	279	452	16
	Hillvalley	101	606	2
	Clean	166	476	2
	Semeion	256	1593	10
	CNAE	856	1080	9

with the classification algorithm used, and the results were compared.

2.4. Chaotic maps for BHOAFS

Chaotic maps are deterministic systems sensitive to initial conditions and vary according to disordered/random behavior. Chaos theory, on the other hand, examines deterministic rules and dynamic system behaviors [63]. According to the initial conditions, chaotic variables can pass all states in certain intervals without repeating. Configuring and creating chaotic number sequences is quick and easy. Because of these behaviors, chaos search is better at climbing the hill and escaping the local optimum than random search. For this reason, many optimization algorithms have been applied, representing chaos maps to update a basic motion mode or stochastic variables used in equations [64]. In previous studies, chaotic maps were used in optimization algorithms in two ways: **Random Number Generator:** Values produced from chaotic maps can be used in creating the initial population, as coefficients in intuitive equations, or as probability values when selecting individuals. **Local Search Technique:** Chaotic search is incorporated into metaheuristic procedures to enrich the search behavior, improve global convergence, and not get stuck in the local optimum [65].

In our experiments, five chaotic maps were used, namely Logistic, Tent, Piecewise, Singer, and Sinusoidal Maps, with the mathematical equations shown in Table 1. These chaotic maps are visualized in Fig. 4 for five different chaotic map values. In the proposed algorithm, one-dimensional chaotic maps are used as random number generators at the beginning of the search process to improve the FS performance of BHOAFS, find effective solutions in the search space, and increase the stability of the algorithm. The initial value was set to 0.7 in all chaotic maps selected for the proposed algorithm [66].

3. Proposed binary chaotic horse herd optimization algorithm (BCHOAFS)

The BCHOAFS generates a random number sequence containing N horses using chaotic maps. Each horse represents a solution in the population for the given FS problem. The population is created using Eq. 13:

Table 5
Comparison between the proposed approaches based on accuracy and the number of features for k-NN and SVM classifier of datasets.

		BHOAFS-kNN		BHOAFS-SVM	
Benchmark	#Features	MeanAcc	#Selected Features	MeanAcc	#Selected Features
Iris	5	**0.98000**	**3**	0.97333	**3**
Diabetes	9	**0.77662**	**5**	0.77623	**5**
Wine	13	**0.95556**	**5**	0.93889	7
Hepatitis	19	0.74000	**5**	**0.76000**	6
Vehicle	18	**0.75529**	**4**	0.65059	6
Zoo	16	0.94545	7	**0.96364**	**6**
Heart	13	**0.85926**	**3**	0.85556	6
Breast Cancer	11	**0.97857**	**6**	**0.97857**	7
Ionosphere	34	0.93056	**6**	**0.96944**	12
Soybean Small	35	**1.00000**	**15**	0.80000	23
Lung-Cancer	56	**0.97500**	13	**0.97500**	**11**
Sonar	60	**0.94762**	21	0.92857	**17**
BreastEW	30	**0.96140**	11	0.95611	**2**
Movementlibras	90	**0.83611**	27	0.83333	**26**
WaveformEW	40	0.84580	**18**	**0.87520**	26
Arrhythmia	279	0.70000	138	**0.70870**	**83**
Hillvalley	101	**0.65574**	**14**	0.48033	25
Clean	166	**0.95417**	131	0.93958	**102**
Semeion	256	**0.98188**	**61**	0.98063	117
CNAE	856	**0.99815**	663	**0.99815**	**661**

Table 6
Comparison between the proposed approaches by means of fitness for k-NN and SVM classifiers of datasets.

Benchmark	BHOAFS-kNN	BHOAFS-SVM	GA	PSO	ALO	GWO	SSA
Wine	**0.05015**	0.06512	0.23000	0.54000	0.37200	0.43000	0.23000
Hepatitis	0.26477	**0.24444**	1.60000	1.50000	1.29300	2.19500	1.26300
Vehicle	**0.25004**	0.35258	2.30000	2.10000	3.01900	2.94500	2.39300
Zoo	0.05963	**0.04225**	1.16000	2.00000	1.87500	1.02700	0.75800
Heart	**0.14703**	0.14838	1.25000	1.85000	2.02600	1.30100	1.34800
Breast Cancer	0.02576	**0.02485**	0.38000	0.37000	0.42500	0.41500	0.39900
Ionosphere	0.07698	**0.03672**	1.27000	1.76000	1.19900	1.41500	1.21400
Soybean Small	**0.00571**	0.20143	0.60000	0.40000	0.01900	2.91300	0.04900
Lung Cancer	**0.03243**	0.03279	1.50000	1.72400	1.90200	2.84400	1.29600
Sonar	**0.05836**	0.07788	1.69000	3.19000	1.76900	2.25600	0.06900
BreastEW	**0.04454**	0.05278	0.42000	0.54000	0.56400	0.65400	0.51200
Movementlibras	**0.16925**	0.17211	2.96000	3.33000	3.88900	2.78400	1.74900
WaveformEW	0.15816	**0.12705**	2.18000	2.40000	2.68100	2.64800	2.02000
Arrhythmia	0.30205	**0.29542**	0.60000	0.90000	0.37400	0.36300	2.93500
Hillvalley	**0.34943**	0.52200	2.45000	2.63000	4.05200	4.08800	2.92000
Clean	**0.04748**	0.06367	0.64000	0.92000	0.16100	0.18400	0.96500
Semeion	0.02556	**0.02461**	-	-	-	-	7.28600
CNAE	**0.00409**	0.00411	2.67000	3.27000	1.66200	2.00000	2.06900

Table 7
Comparison of BHOAFS with other well-known optimization algorithms by mean accuracy.

Benchmark	BHOAFS-kNN	BHOAFS-SVM	GA	PSO	ALO	GWO	SSA
Wine	0.95556	0.93889	0.95700	0.92300	0.94200	0.91200	**0.97500**
Hepatitis	0.74000	0.76000	0.87500	0.86050	0.88250	0.84790	**0.90000**
Vehicle	**0.75529**	0.65059	0.71630	0.66950	0.68140	0.61720	0.72010
Zoo	0.94545	0.96364	0.85400	0.82400	0.80500	0.87500	**0.96750**
Heart	**0.85926**	0.85556	0.82400	0.82200	0.80200	0.80700	0.85210
Breast Cancer	**0.97857**	**0.97857**	0.95500	0.95100	0.95000	0.95300	0.96850
Ionosphere	0.93056	**0.96944**	0.82400	0.84800	0.84300	0.81900	0.91000
Soybean Small	**1.00000**	0.80000	0.94380	0.86480	0.90980	0.92050	**1.00000**
Lung Cancer	**0.97500**	**0.97500**	0.48200	0.56270	0.50560	0.50140	0.90830
Sonar	**0.94762**	0.92857	0.71700	0.72300	0.71400	0.71400	0.93750
BreastEW	**0.96140**	0.95611	0.93500	0.94900	0.94200	0.94900	0.95200
Movementlibras	**0.83611**	0.83333	0.69020	0.64660	0.65970	0.68660	0.82220
WaveformEW	0.84580	**0.87520**	0.76200	0.76200	0.76900	0.76500	0.80080
Arrhythmia	0.70000	**0.70870**	0.58020	0.57070	0.54620	0.56410	0.68280
Hillvalley	0.65574	0.48033	0.56270	0.55070	0.57090	0.55440	**0.66580**
Clean	**0.95417**	0.93958	0.76480	0.77840	0.80980	0.79530	0.90680
Semeion	**0.98188**	0.98063					0.20750
CNAE	**0.99815**	**0.99815**	0.82460	0.81470	0.79620	0.84070	0.79000

Table 8
Comparison of BHOAFS with other well-known binary optimization algorithms by mean accuracy.

Benchmark	BHOAFS-kNN	BHOAFS-SVM	BGA	BPSO	BALO	BGWO	BSSA
Wine	0.95556	0.93889	0.95700	0.92300	0.94200	0.91200	**0.97500**
Hepatitis	0.74000	0.76000	0.87500	0.86050	0.88250	0.84790	**0.90000**
Vehicle	**0.75529**	0.65059	0.71630	0.66950	0.68140	0.61720	0.72010
Zoo	0.94545	0.96364	0.85400	0.82400	0.80500	0.87500	**0.96750**
Heart	**0.85926**	0.85556	0.82400	0.82200	0.80200	0.80700	0.85210
Breast Cancer	**0.97857**	**0.97857**	0.95500	0.95100	0.95000	0.95300	0.96850
Ionosphere	0.93056	**0.96944**	0.82400	0.84800	0.84300	0.81900	0.91000
Soybean Small	**1.00000**	0.80000	0.94380	0.86480	0.90980	0.92050	0.10000
Lung Cancer	**0.97500**	**0.97500**	0.48200	0.56270	0.50560	0.50140	0.90830
Sonar	**0.94762**	0.92857	0.71700	0.72300	0.71400	0.71400	0.93750
BreastEW	**0.96140**	0.95611	0.93500	0.94900	0.94200	0.94900	0.95200
Movementlibras	**0.83611**	0.83333	0.69020	0.64660	0.65970	0.68660	0.82220
WaveformEW	0.84580	**0.87520**	0.76200	0.76200	0.76900	0.76500	0.80080
Arrhythmia	0.70000	**0.70870**	0.58020	0.57070	0.54620	0.56410	0.68280
Hillvalley	0.65574	0.48033	0.56270	0.55070	0.57090	0.55440	**0.66580**
Clean	**0.95417**	0.93958	0.76480	0.77840	0.80980	0.79530	0.90680
Semeion	**0.98188**	0.98063					0.20750
CNAE	**0.99815**	**0.99815**	0.82460	0.81470	0.79620	0.84070	0.79000

$$x_i^{k+1} = x_i^k + Cmap * \left(x_{BestHorse}^k - x_i^k\right) \qquad i = 1, 2, \ldots, N \tag{13}$$

where x_i^k and $x_i^{k+1}$are the positions of the ith horse in iterations k and k + 1, respectively. $x_{BestHorse}^k$ is the position of the best horse in space, and N is the number of horses. Cmap is the value $\in [0, 1]$ produced by the chaotic map. In order to obtain good results in the exploration and exploitation stages and to improve the performance of the proposed algorithm while finding the optimal global solution, the θ parameter was replaced with the Cmap value (Eq. 13) ob-

Table 9
Comparison of BHOAFS with other well-known binary optimization algorithms by the number of selected features.

Benchmark	BHOAFS-kNN	BHOAFS-SVM	BSSA	BBA	BGWO	BPSO
Wine	**5**	7	**5**	**5**	6	**5**
Hepatitis	**5**	6	9	7	8	7
Vehicle	**4**	6	8	9	11	8
Zoo	7	**6**	**6**	**6**	**6**	7
Heart	**3**	6	5	6	6	5
Breast Cancer	6	7	**4**	**4**	**4**	**4**
Ionosphere	**6**	12	14	17	15	14
Soybean Small	15	23	**14**	**14**	18	16
Lung Cancer	13	**11**	24	28	25	28
Sonar	21	**17**	28	27	28	29
BreastEW	11	**2**	12	13	20	17
Movementlibras	27	**26**	40	39	54	40
WaveformEW	**18**	26	23	22	30	23
Arrhythmia	138	**83**	147	146	144	142
Hillvalley	**14**	25	49	49	65	51
Clean	131	102	87	87	111	**86**
Semeion	**61**	117	137	133	152	134
CNAE	663	661	**421**	428	482	433

Table 10
Comparison of BHOAFS with other well-known binary optimization algorithms by means of CPU time (in minutes)

Benchmark	BHOAFS-kNN	BHOAFS-SVM	BSSA	BGWO	BPSO	BGA
Wine	124.95	112.57	215.85	198.80	**103.87**	431.30
Hepatitis	306.15	175.48	224.77	228.27	**102.90**	375.33
Vehicle	344.40	1397.18	239.95	228.87	**158.47**	262.27
Zoo	238.55	185.73	192.67	192.07	**100.93**	188.13
Heart	180.75	**71.93**	336.83	266.53	128.30	225.00
Breast Cancer	148.97	**59.27**	310.47	334.03	122.53	217.87
Ionosphere	872.15	601.52	**55.38**	56.70	110.00	191.93
Soybean Small	170.75	**121.18**	420.03	1575.87	1567.47	2966.07
Lung-Cancer	323.55	597.77	181.48	260.87	**98.63**	183.60
Sonar	1541.08	504.73	131.42	271.37	**105.53**	195.90
BreastEW	329.85	3894.30	532.42	148.90	**142.47**	217.43
Movementlibras	1320.32	1727.52	700.43	439.20	**109.27**	651.70
WaveformEW	8570.32	34783.03	2610.60	315.33	**100.80**	609.37
Arrhythmia	5920.42	9302.28	126.60	680.03	**116.03**	401.37
Hillvalley	2101.37	1647.02	367.88	272.40	**103.57**	188.83
Clean	3700.52	6417.77	**550.62**	2781.57	1416.17	3044.73
Semeion	10828.32	9146.60	292.02	**106.13**	189.40	405.83
CNAE	26083.58	21789.80	**887.27**	–	–	–

Table 11
The accuracy comparison of different variations of BCHOAFS.

Benchmark	BCHOAFS1	BCHOAFS2	BCHOAFS3	BCHOAFS4	BCHOAFS5
Wine	**0.95556**	**0.95556**	**0.95556**	**0.95556**	**0.95556**
Hepatitis	0.72667	0.72667	0.72667	**0.74000**	0.72667
Vehicle	0.73882	**0.75765**	**0.75765**	0.75529	**0.75765**
Zoo	**0.96364**	**0.96364**	**0.96364**	0.94545	**0.96364**
Heart	**0.86296**	**0.86296**	**0.86296**	0.85926	**0.86296**
Breast Cancer	**0.97857**	**0.97857**	**0.97857**	**0.97857**	**0.97857**
Ionosphere	**0.93056**	**0.93056**	0.92778	0.92500	0.92222
Soybean Small	**1.00000**	**1.00000**	**1.00000**	**1.00000**	**1.00000**
Lung-Cancer	**0.97500**	**0.97500**	**0.97500**	**0.97500**	**0.97500**
Sonar	**0.93810**	0.93333	**0.93810**	0.92857	**0.93810**
BreastEW	**0.96140**	**0.96140**	**0.96140**	**0.96140**	**0.96140**
Movementlibras	0.83611	0.83333	**0.84167**	0.82778	0.83333
WaveformEW	0.84680	0.84540	0.84580	0.84840	**0.84880**
Arrhythmia	**0.73478**	0.70217	0.70435	0.70217	0.69783
Hillvalley	0.64098	0.63607	0.63443	0.65082	**0.65410**
Clean	**0.94792**	0.93542	0.94375	0.94375	0.93958
Semeion	0.98000	0.98000	**0.98063**	0.97938	0.97938
CNAE	**0.99815**	**0.99815**	**0.99815**	**0.99815**	**0.99815**

Table 12
The number of selected features comparison of different variations of BCHOAFS.

Benchmark	BCHOAFS1	BCHOAFS2	BCHOAFS3	BCHOAFS4	BCHOAFS5
Wine	**5**	6	**5**	**5**	**5**
Hepatitis	8	5	**4**	5	8
Vehicle	11	**4**	**4**	**4**	**4**
Zoo	**6**	**6**	**6**	7	**6**
Heart	4	4	4	**3**	4
Breast Cancer	**6**	**6**	7	**6**	**6**
Ionosphere	**6**	7	**6**	**6**	9
Soybean Small	17	**13**	15	19	16
Lung-Cancer	**15**	17	16	16	**15**
Sonar	**21**	28	32	26	29
BreastEW	7	14	12	12	**5**
Movementlibras	39	**33**	46	50	**33**
WaveformEW	21	**15**	18	22	18
Arrhythmia	160	177	158	150	**142**
Hillvalley	28	28	49	**21**	24
Clean	**107**	123	120	130	135
Semeion	87	93	**58**	59	82
CNAE	666	**659**	676	674	676

Table 13
According to the mean of accuracy and number of selected features, a comparison of BHOAFS with different variations of BCHOAFS.

Benchmark	BCHOAFS1		BCHOAFS2		BCHOAFS3		BCHOAFS4		BCHOAFS5	
	MeanAcc	MeanFea	MeanAcc	MeanFea	MeanAcc	MeanFea	MeanAcc	MeanFea	MeanAcc	MeanFea
Wine	**0.95556**	6.0	**0.95556**	6.2	0.95500	6.0	**0.95556**	**5.8**	0.95500	6.1
Hepatitis	0.71933	8.6	**0.72333**	7.8	0.71867	**7.6**	0.72067	7.7	0.70733	**7.6**
Vehicle	0.72954	10.3	0.73753	**8.6**	0.73612	9.0	0.73447	9.5	**0.73965**	9.0
Zoo	**0.94727**	7.9	0.93273	8.7	0.94364	8.9	0.93273	**7.8**	0.94182	9.2
Heart	0.86111	4.5	**0.86185**	4.3	**0.86185**	4.3	0.85926	**3.4**	0.86111	4.5
Breast Cancer	**0.97857**	6.6	**0.97857**	**6.0**	**0.97857**	7.0	0.97814	**6.0**	**0.97857**	6.2
Ionosphere	**0.91972**	7.0	0.91528	8.2	0.91806	8.1	0.91472	**6.8**	0.91444	8.4
Soybean Small	**1.00000**	23.2	**1.00000**	22.9	**1.00000**	22.8	**1.00000**	22.2	**1.00000**	**22.0**
Lung-Cancer	0.97250	24.7	**0.97500**	24.2	**0.97500**	22.6	**0.97500**	24.0	0.97000	**19.9**
Sonar	**0.92190**	**25.1**	0.91714	25.2	0.91952	28.0	0.91667	28.1	0.91667	27.7
BreastEW	0.96088	14.8	0.96070	15.8	**0.96140**	**14.1**	0.96070	15.0	0.96053	15.2
Movementlibras	**0.82472**	36.4	0.82222	**32.7**	0.82056	35.9	0.82056	35.6	0.82139	39.6
WaveformEW	0.84320	19.3	0.84322	20.1	0.84240	**18.7**	0.84220	22.5	**0.84378**	19.2
Arrhythmia	0.68783	155.1	0.68761	157.8	**0.69022**	158.3	0.68783	155.3	0.68630	**154.2**
Hillvalley	0.62721	**27.1**	0.62525	30.5	0.62492	37.7	0.62918	39.0	**0.63098**	29.1
Clean	**0.94062**	114.5	0.92813	109.2	0.93125	**106.9**	0.93354	108.5	0.93042	108.0
Semeion	0.97606	91.6	**0.97625**	85.5	0.97550	91.9	0.97594	**77.1**	0.97506	103.7
CNAE	**0.99815**	687.1	**0.99815**	**682.3**	**0.99815**	687.7	**0.99815**	688.1	0.99815	684.6

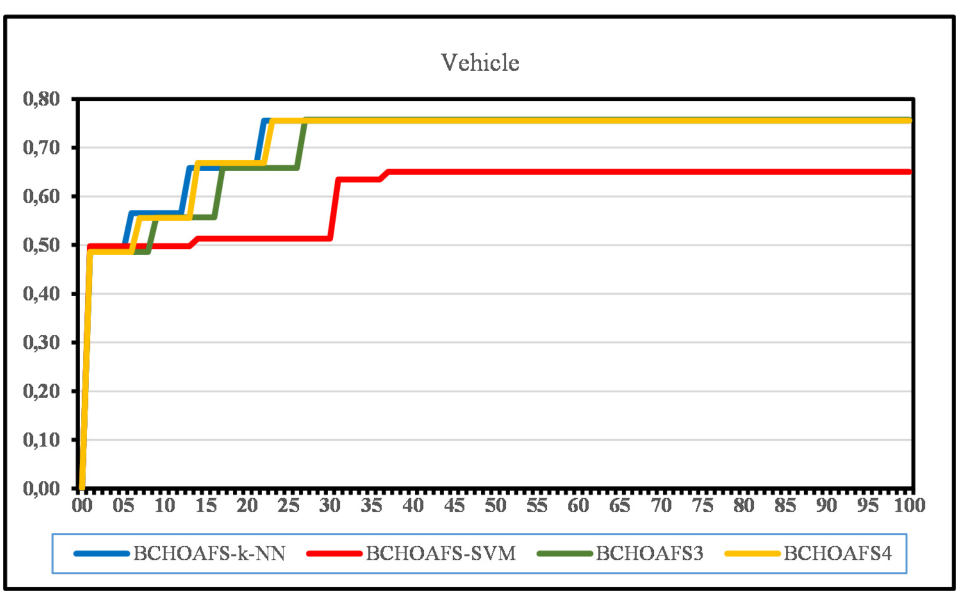

Fig. 6. Convergence Graphic for Vehicle Dataset.

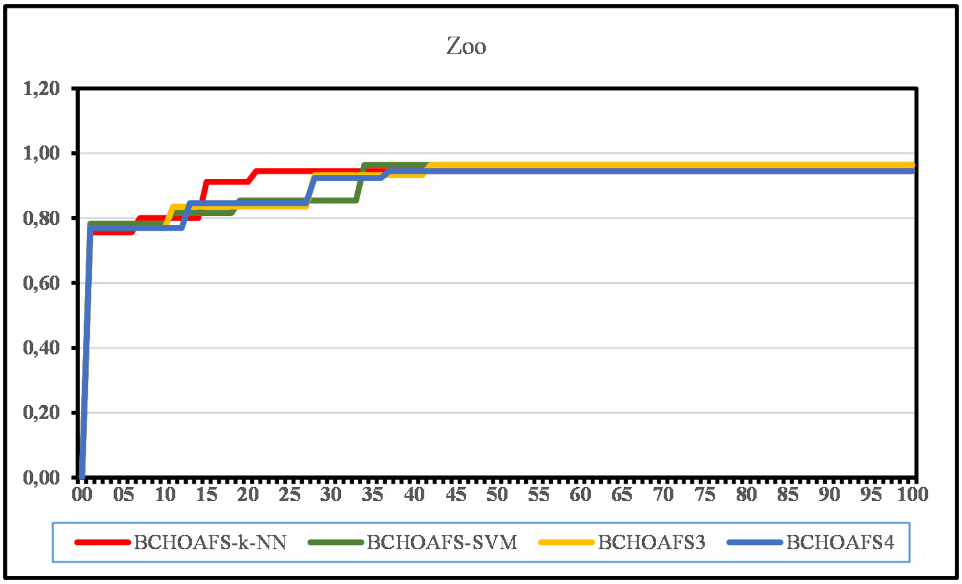

Fig. 7. Convergence Graphic for Zoo Dataset.

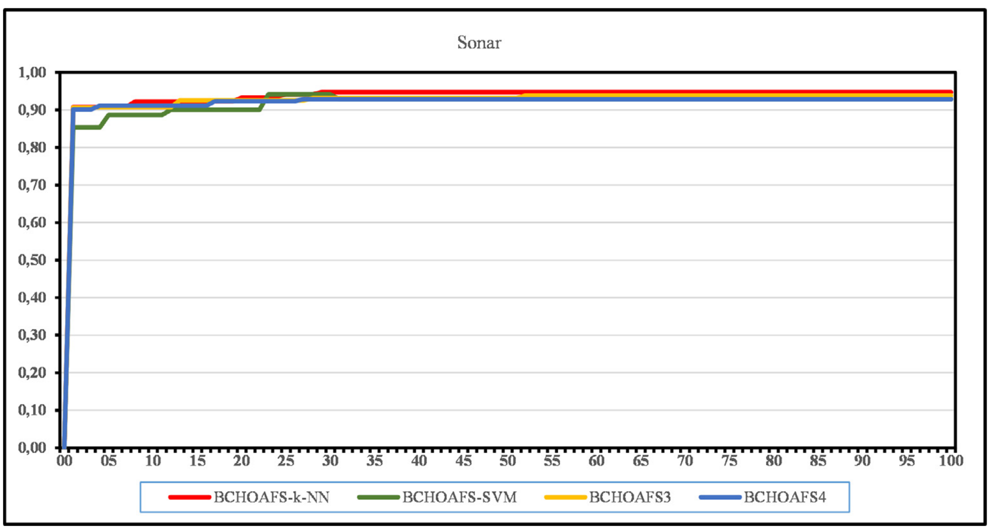

Fig. 8. Convergence Graphic for Sonar Dataset.

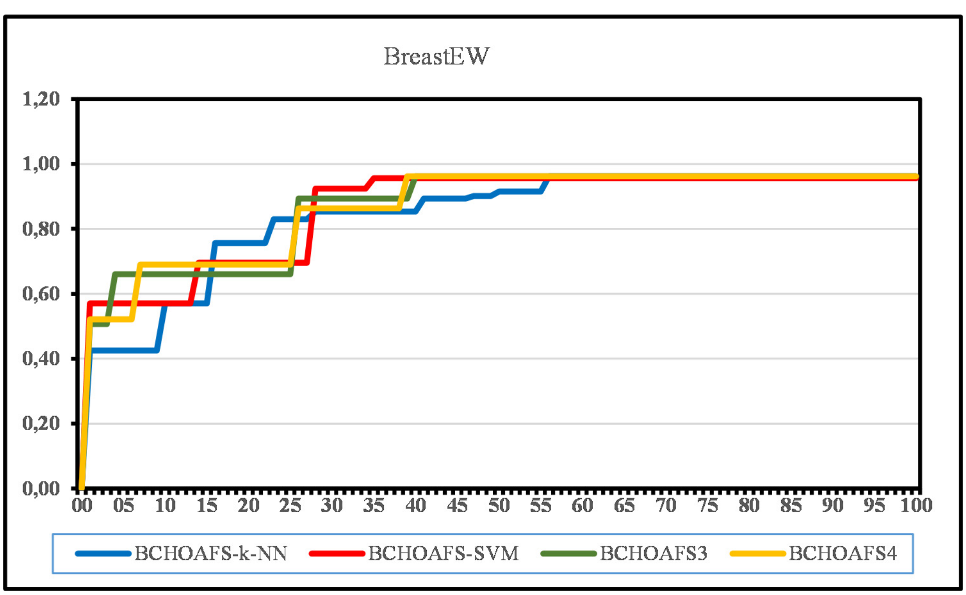

Fig. 9. Convergence Graphic for BreastEW Dataset.

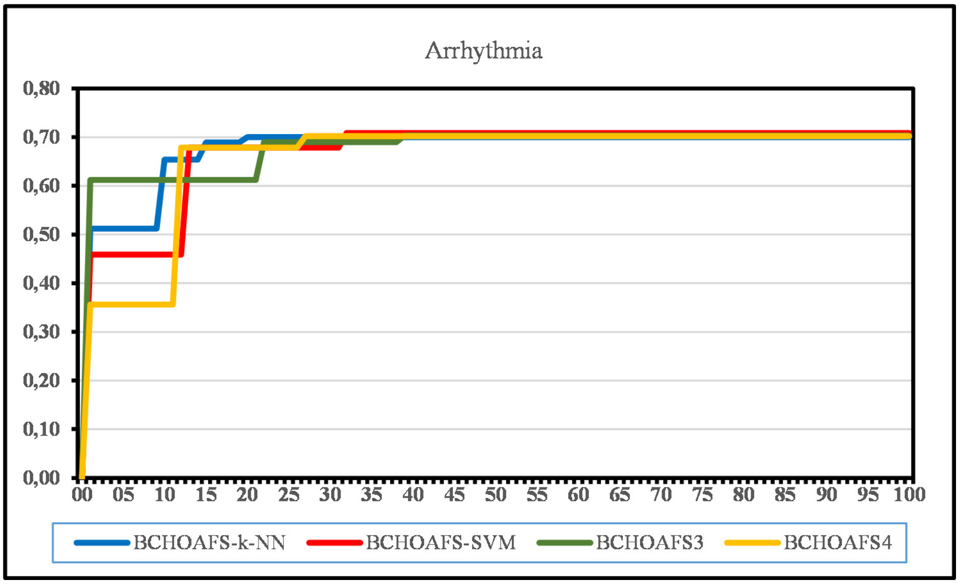

Fig. 10. Convergence Graphic for Arrhythmia Dataset.

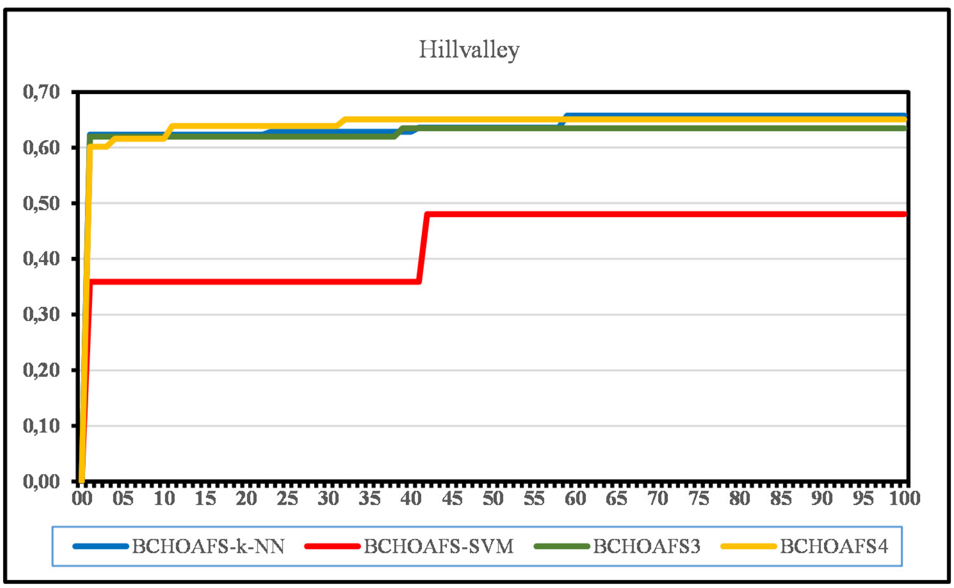

Fig. 11. Convergence Graphic for Hillvalley Dataset.

tained from the chaotic map. Five chaotic maps were used to manipulate the values of the random parameters of the BHOAFS. Fig. 5 shows the pseudo-code of the proposed BCHOAFS.

4. Experiments

4.1. Experimental initializations and parameter settings

Random population generation is provided by using Pareto Law rules in FS. According to this rule, the binary vector array is initially set to 80% zero and 20% one. 90% of the datasets were reserved for training and 10% for testing. k-NN and SVM classifiers were used as classifiers by randomly selecting the data in the training and test set. Results for the test set were obtained by ten-fold cross-validation. The accuracy average of this classifier was used. The fitness function in the FS problem depends not only on the classification accuracy but also on the number of selected features. The one with the higher accuracy value is preferred of the two subsets of features obtained. However, if two subsets have the same accuracy value, the solution with fewer features is preferred. Since the accuracy value is more important than the number of features, the S coefficient used in the fitness function (Eq. 12) was determined as 0.005 [32]. The termination condition of the algorithm can be achieved by reaching the maximum number of iterations or by obtaining the same feature reduction result three times in a row [67]. The size of all algorithms is fixed to the number of features in the original dataset and the number of populations to 10. 90% of the datasets are reserved for training and 10% for testing. The data in the training and test set were randomly selected. BHOAFS was classified with k-NN (k = 5) and SVM, and the results for the training set were obtained by ten-fold cross-validation, as shown in Table 2. Since BHOAFS classified with k-NN gives better results than SVM, only the k-NN classifier was used for BCHOAFS. While creating the BCHOAFS, five chaotic maps were used to manipulate the val-

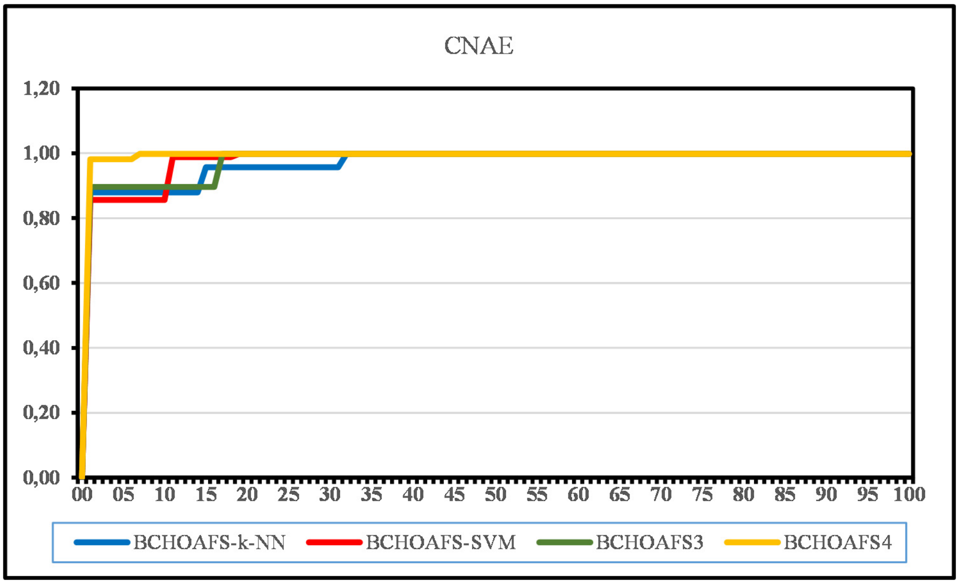

Fig. 12. Convergence Graphic for CNAE Dataset.

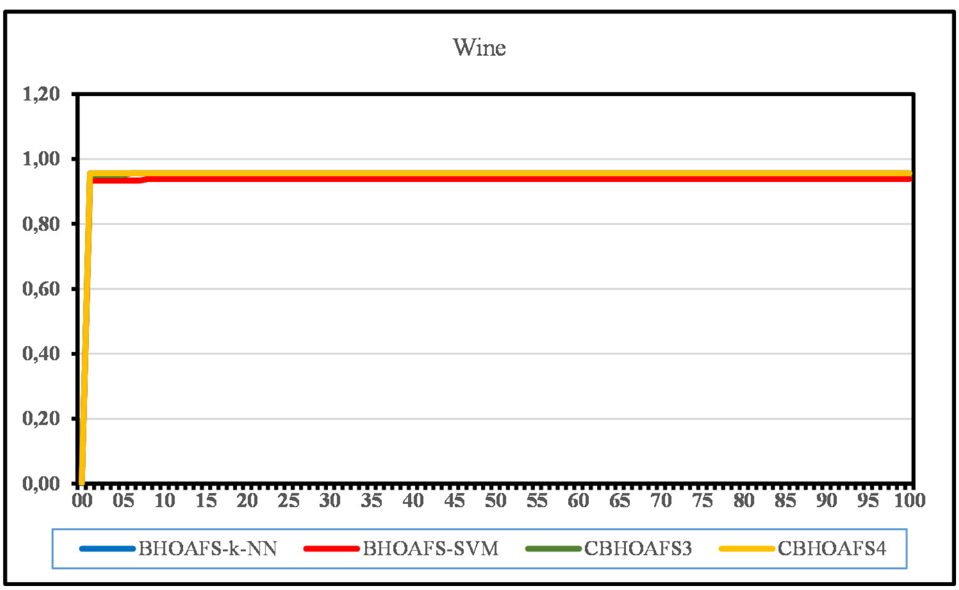

Fig. 13. Convergence Graphic for Wine Dataset.

ues of the random parameters of the BHOAFS. The proposed BHOAFS and BCHOAFS were run using a computer with 4 GB RAM, Intel(R) Core(TM) i5-2450 M 2.5 GHz CPU, coded with Python version 3.9, according to the initial conditions that stated in Table 2. The proposed algorithm has been compared with other well-known optimization algorithms. The applied parameters are the same as their own parameter settings, as stated in Table 3 [68,69].

4.2. Datasets

BHOAFS was run for datasets of 20 different sizes and qualities taken from the UCI data repository. Eighteen of these data sets were used for the BCHOAFS. The definitions of datasets are shown in Table 4.

4.3. Evaluation measures

The datasets are randomly divided into two parts (i.e., training and test datasets). The data splitting is repeated many times to ensure the robustness and measurability of the results. The following statistical measures are tested from validation data at each run.

4.3.1. Mean of classification accuracy

When the algorithm is run N times, the selected feature set is an indicator of how accurate the given classifier is and is calculated as follows in Eq. 14:

$$MeanAcc = \frac{1}{N}\sum_{i=1}^{N}\left(Acc^{k}\right) \tag{14}$$

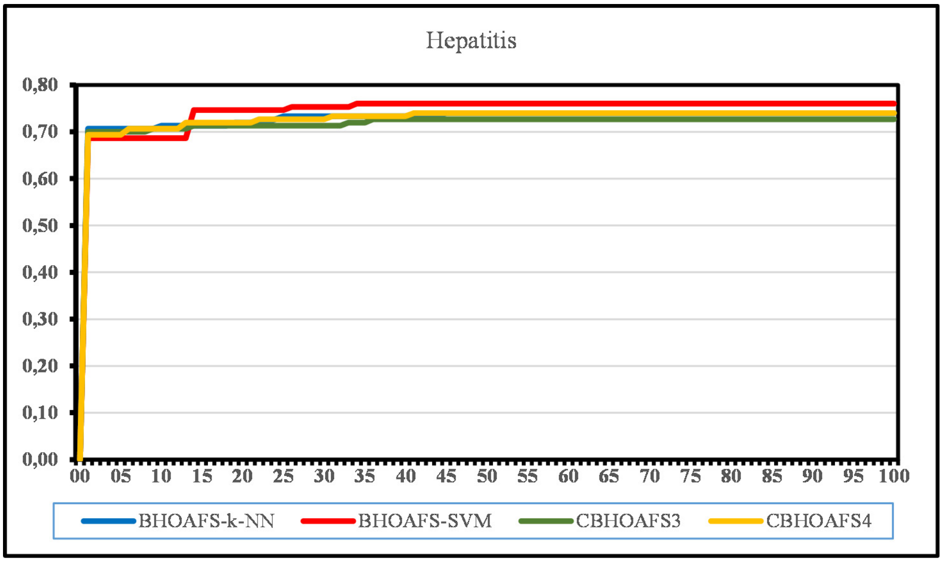

Fig. 14. Convergence Graphic for Hepatitis Dataset.

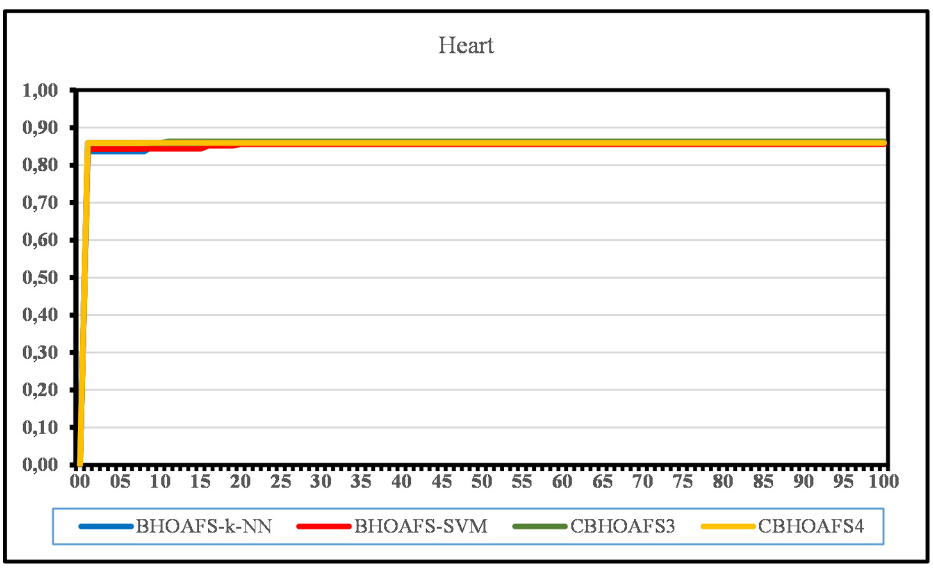

Fig. 15. Convergence Graphic for Heart Dataset.

4.3.2. Mean of selected features

According to Eq. 15, MeanFea is an indicator of the average of the selected features. When the algorithm is run N times, MeanFea is calculated as follows:

$$MeanFea = \frac{1}{N}\sum_{i=1}^{N}\left(\frac{MeanFea^{k}}{D}\right) \tag{15}$$

where $MeanFea^k$ is the selected features at run k, and D shows the dataset's total number of features.

4.3.3. Mean of fitness value

.

$$MeanFit = \frac{1}{N}\sum_{i=1}^{N}(fit_k) \tag{16}$$

It is an indicator of the average value of the fitness function obtained when the algorithm is run N times and is calculated as in Eq. 16. Fit_k represents the fitness function in the kth iteration.

4.4. Comparisons between the proposed optimizers

4.4.1. BHOAFS results

The proposed algorithm was run for 20 datasets from the UCI repository. However, eighteen of 20 datasets are compared with well-known algorithms in the literature. [32]. According to the algorithm results in Table 5, the BHOAFS produced similar results for both k-NN and SVM for most of the datasets. It was observed that k-NN gave better results for Iris, Diabetes, Wine, Vehicle, Heart, Soybean Small, Sonar, BreastEW, Movementlibras, Hillvalley, Clean, and Semeion datasets. For Hepatitis, Zoo, Ionosphere, WaveformEW, and Arrhythmia datasets, SVM produced better accuracy. For the Breast Cancer, Lung-Cancer, and CNAE datasets, both k-NN

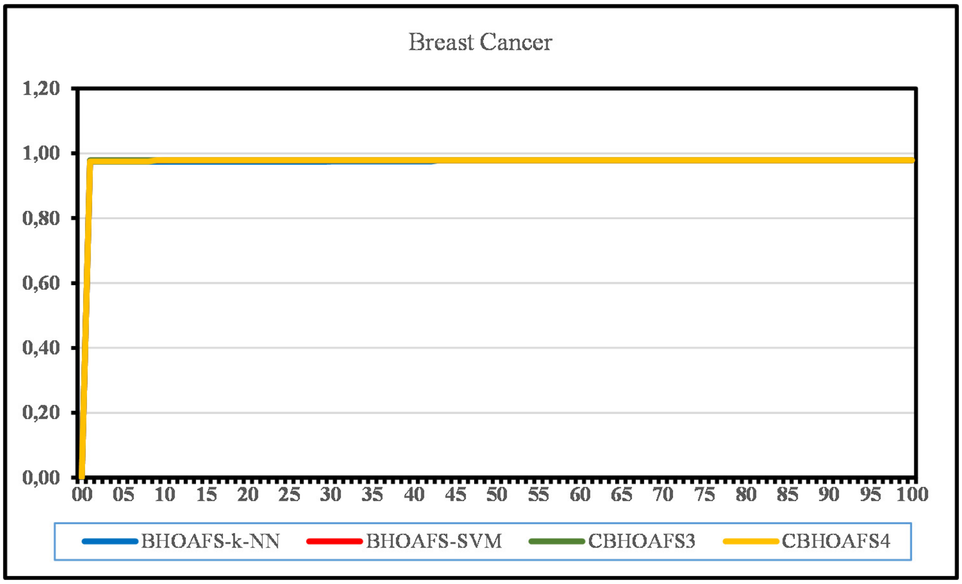

Fig. 16. Convergence Graphic for Breast Cancer Dataset.

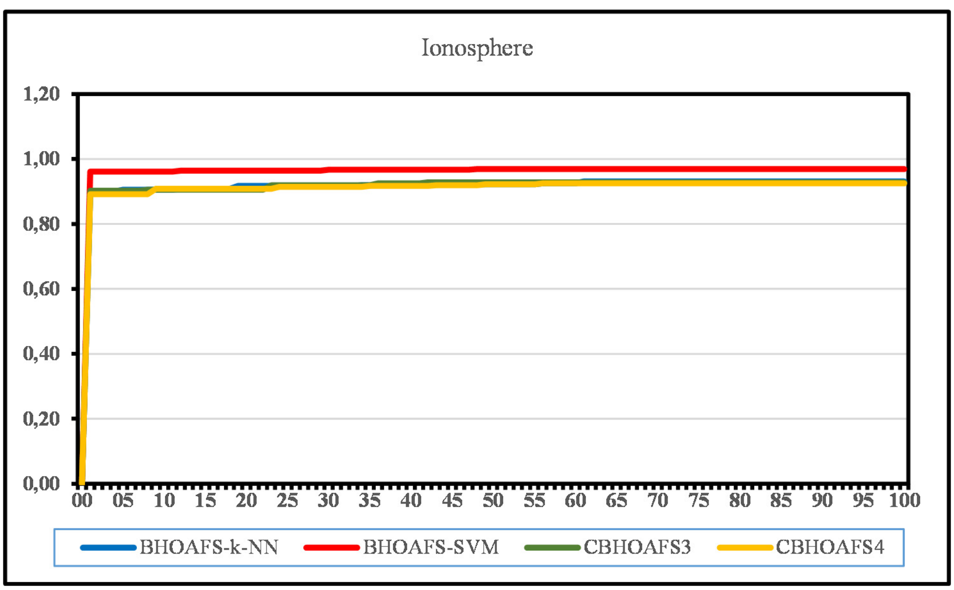

Fig. 17. Convergence Graphic for Ionosphere Dataset.

and SVM produced the same results. According to the results seen in Table 5, BHOAFS-kNN is more effective for large scale data.

The mentioned performance measure has a higher accuracy value and a lower number of FS. This situation can be clearly seen in Table 6, which shows the fitness values calculated according to Eq. 16. Table 6 shows the fitness function comparison of the BHOAFS with the well-known optimization methods (GA, PSO, ALO, GWO, SSA) in the literature [69,68] for both k-NN and SVM classifiers. Wine, Vehicle, Heart, Soybean Small, Lung-Cancer, Sonar, BreastEW, Movementlibras, Hillvalley, Clean, and CNAE datasets for BHOAFS-kNN produced lower fitness results. For BHOAFS-SVM, Hepatitis, Zoo, Breast Cancer, Ionosphere, WaveformEW, Arrhythmia, and Semeion datasets produced lower fitness results.

Similarly, the accuracy values of these fitness values are shown in Table 7. As can be seen in Table 7, for Vehicle, Heart, Breast Cancer, Soybean Small, Lung-Cancer, Sonar, BreastEW, Movementlibras, Clean, Semeion, CNAE datasets, BHOAFS-kNN produced good results among other compared algorithms. BHOAFS-SVM produced the best results for Breast Cancer, Ionosphere, Lung-Cancer, WaveformEW, Arrhythmia, and CNAE datasets.

Table 8 shows the proposed algorithm's comparison with well-known binary optimization methods in the literature (i.e., BGA, BPSO, BALO, BGWO, BSSA). As can be seen in Table 8, for Vehicle, Heart, Breast Cancer, Soybean Small, Lung Cancer, Sonar, BreastEW, Movementlibras, Clean, Semeion, and CNAE datasets, BHOAFS-kNN produced good results among other compared algorithms. BHOAFS-SVM produced the best results for Breast Cancer, Ionosphere, Lung-Cancer, WaveformEW, Arrhythmia, and CNAE datasets. Experimental results proved that the BHOAFS classified by k-NN (BHOAFS-kNN) outperformed the BHOAFS version classified by SVM (BHOAFS-SVM).

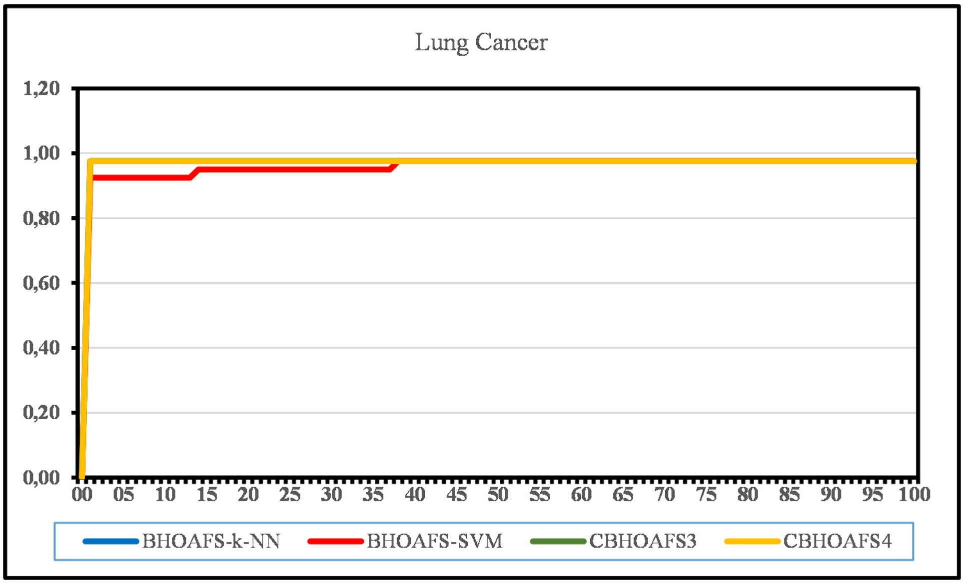

Fig. 18. Convergence Graphic for Lung Cancer Dataset.

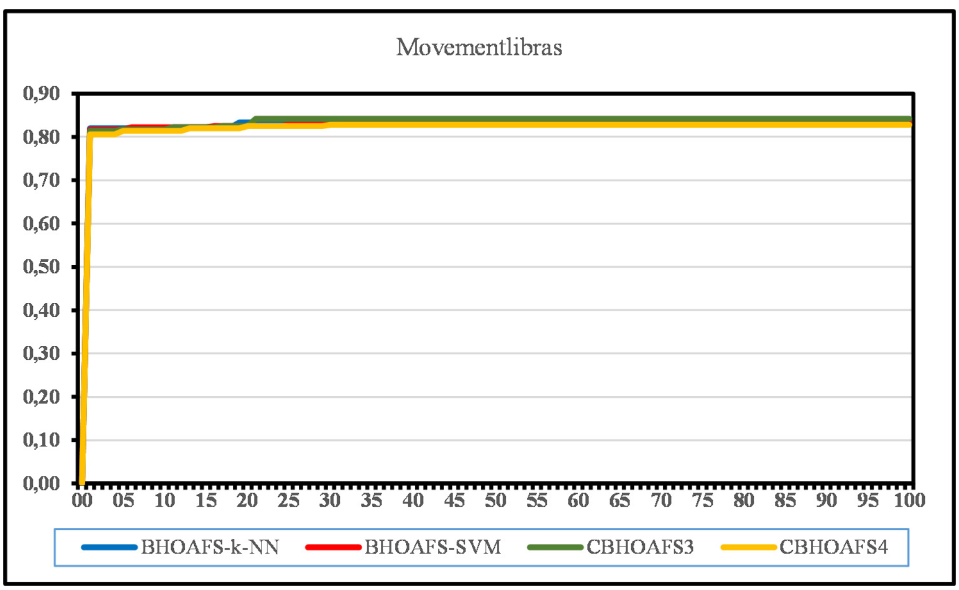

Fig. 19. Convergence Graphic for Movementlibras Dataset.

In Table 9, the comparison of the selected features of the algorithm can be seen in Table 10, where the running time of the algorithm is shown for k-NN and SVM. As can be seen in Table 9, BHOAFS-kNN has the least feature selection among the other compared algorithms for Wine, Hepatitis, Vehicle, Heart, Ionosphere, WaveformEW, Hillvalley, and Semeion datasets. For Zoo, Lung Cancer, Sonar, BreastEW, Movementlibras, and Arrhythmia datasets, BHOAFS-SVM made the least feature selection. Table 10 shows that BHOAFS-SVM for Datasets for Heart, Breast Cancer, and Soybean Small datasets produces better results in less time. It was observed that the study time for some datasets was lower than previous studies [32,69]. The best results in the tables are boldfaced.

As the accuracy value increases and the number of features decreases, the run time for some data sets is also lower than the run times in the literature [32,69,68]. This shows that the proposed algorithm is multi-objective.

4.4.2. BCHOAFS results

The performance of the five chaotic optimizers proposed in this section Logistics BHOAFS (BCHOAFS1), Tent BHOAFS (BCHOAFS2), Piecewise BHOAFS (BCHOAFS3), Singer BHOAFS (BCHOAFS4), Sinusoidal BHOAFS (BCHOAFS5) is given for 18 datasets. The five proposed algorithms are designed separately and compared with the BHOAFS-kNN proposed in the previous section with well-known optimization algorithms that solve FS problems. In this study, all experiments are performed with similar initial conditions. Table 11 shows the best accuracy values obtained during the entire iteration of the proposed BCHOAFS (i.e., BCHOAFS1, BCHOAFS2, BCHOAFS3,

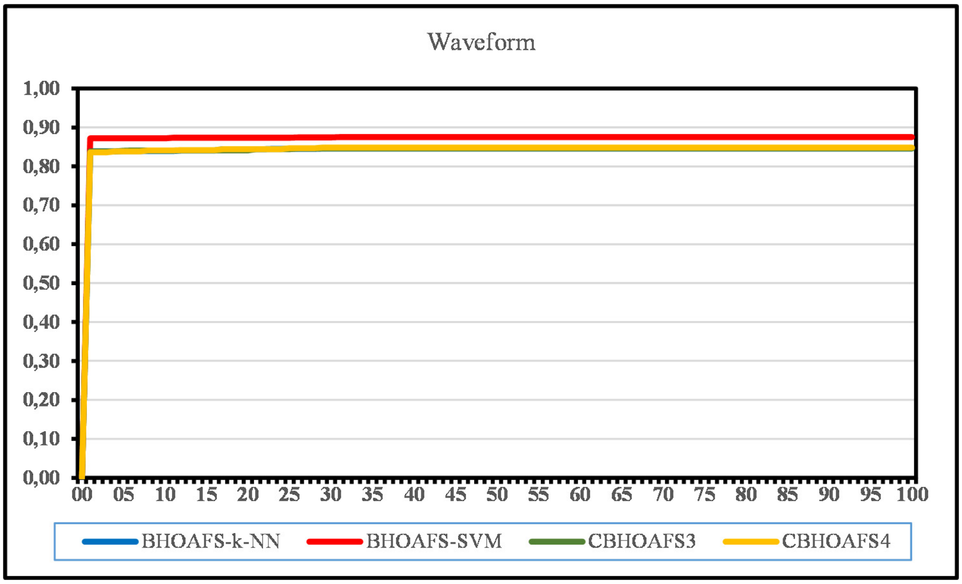

Fig. 20. Convergence Graphic for Waveform Dataset.

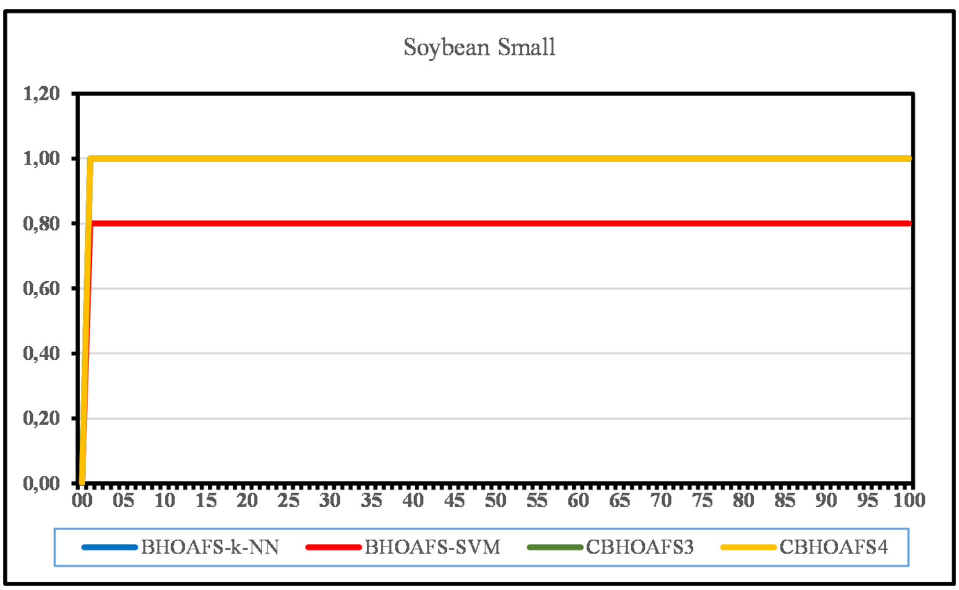

Fig. 21. Convergence Graphic for Soybean Small Dataset.

BCHOAFS4, BCHOAFS5). In approximately 70% of the datasets, the BCHOAFS3 and BCHOAFS4 yielded either better or the same results as BHOAFS-kNN. Table 12 gives information about the selected feature numbers of the BCHOAFS versions. The average accuracy and the number of selected features of the BCHOAFS can be seen in Table 13. In Table 11–13, the best results are boldfaced.

When chaotic maps are ranked in terms of obtained accuracy, and the number of selected features, Piecewise and Singer chaotic maps gave the best results. In contrast, Logistic, Tent, and Sinusoidal maps followed them. All these results prove that Piecewise and Singer Map-based BCHOAFS (BCHOAFS3 and BCHOAFS4) reduces the number of selected features while increasing the quality compared to the proposed BCHOAFS versions.

Convergence graphics of the datasets can be shown in Fig. 6–23.

4.4.3. Friedman test results

Hypothesis testing has been used extensively to infer the algorithms' comparability [70]. To draw a conclusion, it is essential to identify the alternative hypothesis H1 and the null hypothesis H0. A claim known as the null hypothesis frequently indicates that there are no differences between the algorithms under comparison. On the other hand, the alternative hypothesis reflects the variations. For our needs: H0: The compared algorithms do not differ from one another. H1: The compared algorithms differ from one another. The statistical test probability value, which determines whether the hypothesis is rejected or not, is α. Our test has a tolerance threshold of $\alpha = 0.05$.

If we look at the [71] for the expected X^2 value for df = 10 and $\alpha = 0.05$ is 18.31. The null hypothesis can be rejected if the calculated value of X^2 is higher than the expected value and the p-value

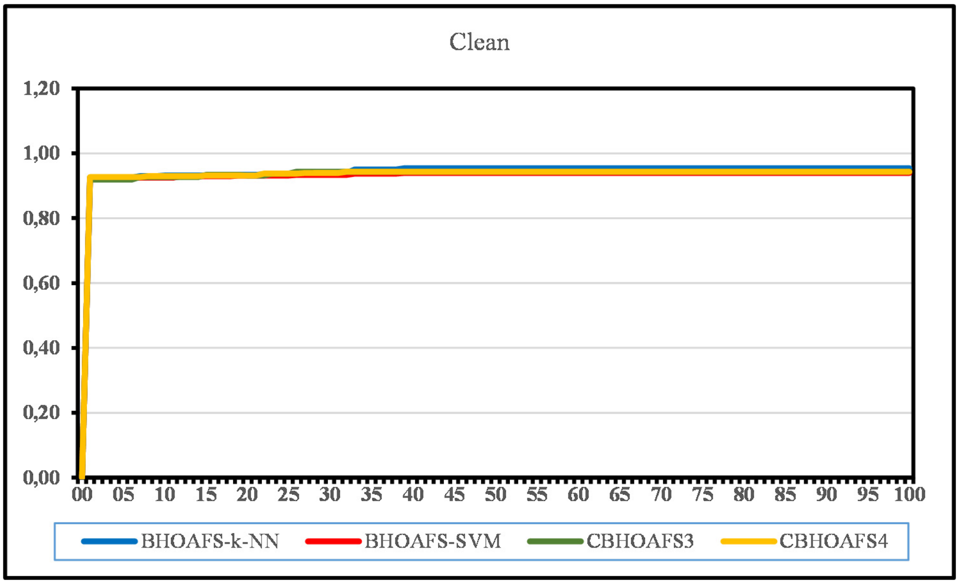

Fig. 22. Convergence Graphic for Clean Dataset.

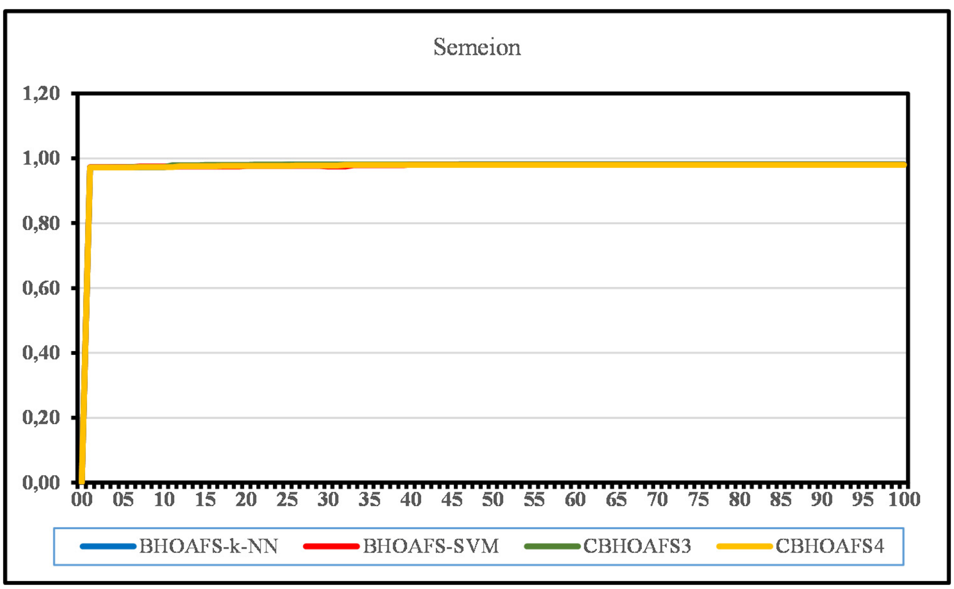

Fig. 23. Convergence Graphic for Semeion Dataset.

Table 14
Friedman test statistics of the algorithms.

Friedman test statistics	
N	17
Chi-Square X^2	55.611
df	10
p	$2.4274E-8$

is smaller than the $\alpha = 0.5$. We can see in Table 14 the calculated value of X^2 is 55.611 and $p = 2.4274E-8$, which the calculated X^2 is higher than the expected value and p is smaller than the α. That means the null hypothesis must be rejected. We can say that our proposed algorithms are different than the compared algorithms.

4.4.4. Wilcoxon test results

For the Wilcoxon test, the differences in the algorithms are described by p values. If the p-value is lower than the significance value $\alpha = 0.05$, we can say that the results of the algorithms are different from the compared algorithms. If the p-value is higher, that means the algorithm has similar results.

We can see in the Table 15 that p values are smaller than the significance value $\alpha = 0.05$, which means that our proposed algorithm has different results than most of the compared algorithms. In the case of the BSSA algorithm, the proposed algorithm has similar results. However, if we look at the accuracy results in Table 7 and Table 11, we can see our proposed algorithm has yielded better results.

Table 15
p-Values for the Wilcoxon signed-rank test.

Algorithm	BHOAFS-kNN	BCHOAFS1	BCHOAFS2	BCHOAFS3	BCHOAFS4	BCHOAFS5	BGA	BPSO	BALO	BGWO	BSSA
BHOAFS-kNN	1	0.646	0.328	0.594	0.050	0.388	0.022	0.006	0.010	0.009	0.177
BCHOAFS1		1	0.176	0.484	0.433	0.674	0.017	0.004	0.010	0.007	0.149
BCHOAFS2			1	0.161	0.790	0.612	0.025	0.006	0.011	0.010	0.163
BCHOAFS3				1	0.328	0.398	0.019	0.006	0.010	0.010	0.163
BCHOAFS4					1	0.534	0.022	0.007	0.015	0.011	0.227
BCHOAFS5						1	0.022	0.006	0.011	0.009	0.136

5. Conclusions

This article proposes the binary version of HOA, which mimics the life cycles and searching behaviors of horses, has been applied to a wrapper-based FS problem using classification algorithms (BHOAFS-kNN and BHOAFS-SVM). Experimental results show that BHOAFS-kNN results outperform BHOAFS-SVM. Then BHOAFS-kNN was combined with five chaotic maps and named as BCHOAFS-Logistics (BCHOAFS1), BCHOAFS - Tent (BCHOAFS2), BCHOAFS - Piecewise (BCHOAFS3), BCHOAFS - Singer (BCHOAFS4), and BCHOAFS - Sinusoidal (BCHOAFS5). The Age Determination process used in the proposed algorithm allows the horses to do a global search in the search space. The SMF operator, which is proposed as a local search technique, has been developed to overcome the disadvantages, such as early convergence while preserving population diversity. A comprehensive study was conducted using 18 standard datasets from the UCI repository of varying sizes and characters to evaluate the effectiveness of the proposed BHOAFS and BCHOAFS versions. The performance of the proposed methods was compared with various methods such as well-known optimization algorithms in the literature such as GA, PSO, ALO, GWO, SSA [69], and binary optimization algorithms such as BGA, BPSO, BALO, BGWO, and BSSA [68]. When the FS problem is considered a multi-objective problem, the classification accuracy should be at the highest level, and the number of selected features should be at the lowest level. When the results were examined, the best performance among the recommended algorithms was found to belong to BCHOAFS3 and BCHOAFS4 algorithms (BCHOAFS-Piecewise and BCHOAFS-Singer) in terms of classification accuracy and FS.

In general, applying chaotic maps to optimization algorithms is easy and practical. According to this study, chaotic maps can improve results when applied to previously proposed algorithms. The main reason for this improvement is that the algorithm can be compared more accurately since the initial conditions are designed in advance. Thus, bit-changing operations during the iteration will result in more specific exploitation and exploration, and a balance will be established between the algorithm's local and global search operations. Taking full advantage of this situation, our proposed algorithm proves that chaotic maps are one of the best enhancement options for currently proposed algorithms when the results obtained are examined. Based on the promising results of BCHOAFS, there is enough evidence to compare its performance with similar methods.

Finally, the results show that the proposed BCHOAFS achieves high competitive results and can be expressed as a multi-objective optimization problem. The most important result of this study is that BCHOAFS versions are useful in systems that require FS preprocessing, as it is the first chaotic-based algorithm designed specifically for large scale data among optimization-based feature selection algorithms. When the SMF operator designed as a local search strategy and chaotic maps are combined, the results prove the novelty of the proposed algorithm.

Declaration of Competing Interest

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

References

[1] J.N.P., R. Aruna, Big data analytics in health care by data mining and classification techniques, ICT Express 8 (2) (2022) 250–257. doi: 10.1016/j.icte.2021.07.001. URL: https://linkinghub.elsevier.com/retrieve/pii/S2405959521000849.

[2] B.H. Nguyen, B. Xue, M. Zhang, A survey on swarm intelligence approaches to feature selection in data mining, Swarm Evolut. Comput. 54 (2020), https://doi.org/10.1016/j.swevo.2020.100663. URL: https://linkinghub.elsevier.com/retrieve/pii/S2210650219303104 100663.

[3] H.R. Kanan, K. Faez, An improved feature selection method based on ant colony optimization (ACO) evaluated on face recognition system, Appl. Math. Comput. 205 (2) (2008) 716–725, https://doi.org/10.1016/j.amc.2008.05.115. URL: https://linkinghub.elsevier.com/retrieve/pii/S0096300308003500.

[4] H. Uguz, A two-stage feature selection method for text categorization by using information gain, principal component analysis and genetic algorithm, Knowl.-Based Syst. 24 (7) (2011) 1024–1032, https://doi.org/10.1016/j.knosys.2011.04.014. URL: https://linkinghub.elsevier.com/retrieve/pii/S0950705111000803.

[5] A.K. Shukla, P. Singh, M. Vardhan, A hybrid gene selection method for microarray recognition, Biocybern. Biomed. Eng. 38 (4) (2018) 975–991, https://doi.org/10.1016/j.bbe.2018.08.004. URL: https://linkinghub.elsevier.com/retrieve/pii/S0208521618302316.

[6] P.D. Sheth, S.T. Patil, M.L. Dhore, Evolutionary computing for clinical dataset classification using a novel feature selection algorithm, J. King Saud Univ. – Comput. Inform. Sci. 34 (8) (2022) 5075–5082, https://doi.org/10.1016/j.jksuci.2020.12.012. URL: https://linkinghub.elsevier.com/retrieve/pii/S1319157820306200.

[7] N. Khare, P. Devan, C. Chowdhary, S. Bhattacharya, G. Singh, S. Singh, B. Yoon, SMO-DNN: Spider Monkey Optimization and Deep Neural Network Hybrid Classifier Model for Intrusion Detection, Electronics 9 (4) (2020) 692, https://doi.org/10.3390/electronics9040692. URL: https://www.mdpi.com/2079-9292/9/4/692.

[8] A. Jovic, K. Brkic, N. Bogunovic, A review of feature selection methods with applications, in: 2015 38th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO), IEEE, Opatija, Croatia, 2015, pp. 1200–1205. doi: 10.1109/MIPRO.2015.7160458. URL: http://ieeexplore.ieee.org/document/7160458/.

[9] N. Sánchez-Maroño, A. Alonso-Betanzos, M. Tombilla-Sanromán, Filter Methods for Feature Selection – A Comparative Study, in: H. Yin, P. Tino, E. Corchado, W. Byrne, X. Yao (Eds.), Intelligent Data Engineering and Automated Learning – IDEAL 2007, vol. 4881, Springer, Berlin Heidelberg, Berlin, Heidelberg, 2007, pp. 178–187. doi: 10.1007/978-3-540-77226-_19. URL: http://link.springer.com/10.1007/978-3-540-77226-2_19.

[10] R. Kohavi, G.H. John, Wrappers for feature subset selection, Artif. Intell. 97 (1–2) (1997) 273–324, https://doi.org/10.1016/S0004-3702(97)00043-X. URL: https://linkinghub.elsevier.com/retrieve/pii/S000437029700043X.

[11] Z. Liu, B. Chang, F. Cheng, An interactive filter-wrapper multi-objective evolutionary algorithm for feature selection, Swarm Evolut. Comput. 65 (2021), https://doi.org/10.1016/j.swevo.2021.100925. URL: https://linkinghub.elsevier.com/retrieve/pii/S2210650221000869 100925.

[12] H. Liu, H. Motoda, R. Setiono, Z. Zhao, Feature Selection: An Ever Evolving Frontier in Data Mining, in: Proceedings of the Fourth International Workshop on Feature Selection in Data Mining 10, 2010, pp. 4–13. URL: https://proceedings.mlr.press/v10/liu10b.html.

[13] H. Faris, A.A. Heidari, A.M. Al-Zoubi, M. Mafarja, I. Aljarah, M. Eshtay, S. Mirjalili, Time-varying hierarchical chains of salps with random weight networks for feature selection, Expert Syst. Appl. 140 (2020), https://doi.org/10.1016/j.eswa.2019.112898. URL: https://linkinghub.elsevier.com/retrieve/pii/S0957417419306141 112898.

[14] S. Garg, K. Kaur, N. Kumar, G. Kaddoum, A.Y. Zomaya, R. Ranjan, A Hybrid Deep Learning-Based Model for Anomaly Detection in Cloud Datacenter Networks,

IEEE Trans. Netw. Serv. Manage. 16 (3) (2019) 924–935, https://doi.org/10.1109/TNSM.2019.2927886. URL: https://ieeexplore.ieee.org/document/8758843/.

[15] P. Moradi, M. Gholampour, A hybrid particle swarm optimization for feature subset selection by integrating a novel local search strategy, Appl. Soft Comput. 43 (2016) 117–130, https://doi.org/10.1016/j.asoc.2016.01.044. URL: https://linkinghub.elsevier.com/retrieve/pii/S1568494616300321.

[16] X. Xia, L. Gui, F. Yu, H. Wu, B. Wei, Y.L. Zhang, Z.H. Zhan, Triple Archives Particle Swarm Optimization, IEEE Trans. Cybern. 50 (12) (2020) 4862–4875, https://doi.org/10.1109/TCYB.2019.2943928. URL: https://ieeexplore.ieee.org/document/8866749/.

[17] B. Wei, X. Xia, F. Yu, Y. Zhang, X. Xu, H. Wu, L. Gui, G. He, Multiple adaptive strategies based particle swarm optimization algorithm, Swarm Evolut. Comput. 57 (2020), https://doi.org/10.1016/j.swevo.2020.100731. URL: https://linkinghub.elsevier.com/retrieve/pii/S2210650220303849 100731.

[18] L. Brezocnik, I. Fister, V. Podgorelec, Swarm Intelligence Algorithms for Feature Selection: A Review, Appl. Sci. 8 (9) (2018) 1521, https://doi.org/10.3390/app8091521. URL: https://www.mdpi.com/2076-3417/8/9/1521.

[19] P. Ghamisi, J.A. Benediktsson, Feature Selection Based on Hybridization of Genetic Algorithm and Particle Swarm Optimization, IEEE Geosci. Remote Sens. Lett. 12 (2) (2015) 309–313, https://doi.org/10.1109/LGRS.2014.2337320. URL: https://ieeexplore.ieee.org/document/6866865.

[20] X. Han, X. Chang, L. Quan, X. Xiong, J. Li, Z. Zhang, Y. Liu, Feature subset selection by gravitational search algorithm optimization, Inf. Sci. 281 (2014) 128–146, https://doi.org/10.1016/j.ins.2014.05.030. URL: https://linkinghub.elsevier.com/retrieve/pii/S0020025514005829.

[21] S. Medjahed, T. Ait Saadi, A. Benyettou, M. Ouali, Gray Wolf Optimizer for hyperspectral band selection, Appl. Soft Comput. 40 (2016) 178–186, https://doi.org/10.1016/j.asoc.2015.09.045. URL: https://linkinghub.elsevier.com/retrieve/pii/S1568494615006250.

[22] Q. Al-Tashi, S.J. Abdul Kadir, H.M. Rais, S. Mirjalili, H. Alhussian, Binary Optimization Using Hybrid Grey Wolf Optimization for Feature Selection, IEEE Access 7 (2019) 39496–39508, https://doi.org/10.1109/ACCESS.2019.2906757. URL: https://ieeexplore.ieee.org/document/8672550/.

[23] L. Mohammad, Q. Abualigah, Feature selection and enhanced krill herd algorithm for text document clustering, Springer, Berlin Heidelberg, New York, NY, 2019.

[24] M.M. Kabir, M. Shahjahan, K. Murase, A new hybrid ant colony optimization algorithm for feature selection, Expert Syst. Appl. 39 (3) (2012) 3747–3763, https://doi.org/10.1016/j.eswa.2011.09.073. URL: https://linkinghub.elsevier.com/retrieve/pii/S0957417411013960.

[25] R.Y.M. Nakamura, L.A.M. Pereira, D. Rodrigues, K.A.P. Costa, J.P. Papa, X.S. Yang, Binary Bat Algorithm for Feature Selection, in, Swarm Intelligence and Bio-Inspired Computation, Elsevier (2013) 225–237, https://doi.org/10.1016/B978-0-12-405163-8.00009-0. URL: https://linkinghub.elsevier.com/retrieve/pii/B9780124051638000090.

[26] M. Mafarja, I. Aljarah, H. Faris, A.I. Hammouri, A.M. Al-Zoubi, S. Mirjalili, Binary grasshopper optimisation algorithm approaches for feature selection problems, Expert Syst. Appl. 117 (2019) 267–286, https://doi.org/10.1016/j.eswa.2018.09.015. URL: https://linkinghub.elsevier.com/retrieve/pii/S0957417418305864.

[27] L. Abualigah, D. Yousri, M. Abd Elaziz, A.A. Ewees, M.A. Al-qaness, A.H. Gandomi, Aquila Optimizer: A novel meta-heuristic optimization algorithm, Comput. Ind. Eng. 157 (2021), https://doi.org/10.1016/j.cie.2021.107250. URL: https://linkinghub.elsevier.com/retrieve/pii/S0360835221001546 107250.

[28] M. Tubishat, N. Idris, L. Shuib, M.A. Abushariah, S. Mirjalili, Improved Salp Swarm Algorithm based on opposition based learning and novel local search algorithm for feature selection, Expert Syst. Appl. 145 (2020), https://doi.org/10.1016/j.eswa.2019.113122. URL: https://linkinghub.elsevier.com/retrieve/pii/S0957417419308395 113122.

[29] M. Tubishat, S. Ja'afar, M. Alswaitti, S. Mirjalili, N. Idris, M.A. Ismail, M.S. Omar, Dynamic Salp swarm algorithm for feature selection, Expert Syst. Appl. 164 (2021), https://doi.org/10.1016/j.eswa.2020.113873. URL: https://linkinghub.elsevier.com/retrieve/pii/S0957417420306801 113873.

[30] D. Rodrigues, V.H.C. de Albuquerque, J.P. Papa, A multi-objective artificial butterfly optimization approach for feature selection, Appl. Soft Comput. 94 (2020), https://doi.org/10.1016/j.asoc.2020.106442. URL: https://linkinghub.elsevier.com/retrieve/pii/S1568494620303823 106442.

[31] P.S., F. Al-Turjman, T. Stephan, An automated breast cancer diagnosis using feature selection and parameter optimization in ANN, Comput. Electr. Eng. 90 (2021) 106958. doi: 10.1016/j.compeleceng.2020.106958. URL: https://linkinghub.elsevier.com/retrieve/pii/S0045790620308041

[32] E. Bas, E. Ülker, An efficient binary social spider algorithm for feature selection problem, Expert Syst. Appl. 146 (2020), https://doi.org/10.1016/j.eswa.2020.113185. URL: https://linkinghub.elsevier.com/retrieve/pii/S0957417420300117 113185.

[33] H. Hichem, M. Rafik, M.T. Mesaaoud, Pso with crossover operator applied to feature selection problem in classification, Informatica 42(2) (2018) 189–198. URL: https://www.informatica.si/index.php/informatica/article/view/1373.

[34] A. Dasgupta, A. Banerjee, A.G. Dastidar, A. Barman, S. Chakraborty, A Study and Analysis of a Feature Subset Selection Technique Using Penguin Search Optimization Algorithm, Data Science, CRC Press, in, 2019, pp. 47–65.

[35] E. Hancer, B. Xue, M. Zhang, D. Karaboga, B. Akay, Pareto front feature selection based on artificial bee colony optimization, Inf. Sci. 422 (2018) 462–479, https://doi.org/10.1016/j.ins.2017.09.028. URL: https://linkinghub.elsevier.com/retrieve/pii/S0020025516312609.

[36] M. Canayaz, MH-COVIDNet: Diagnosis of COVID-19 using deep neural networks and meta-heuristic-based feature selection on X-ray images, Biomed. Signal Process. Control 64 (2021), https://doi.org/10.1016/j.bspc.2020.102257. URL: https://linkinghub.elsevier.com/retrieve/pii/S1746809420303840 102257.

[37] K.K. Ghosh, S. Ahmed, P.K. Singh, Z.W. Geem, R. Sarkar, Improved Binary Sailfish Optimizer Based on Adaptive β -Hill Climbing for Feature Selection, IEEE Access 8 (2020) 83548–83560, https://doi.org/10.1109/ACCESS.2020.2991543. URL: https://ieeexplore.ieee.org/document/9082591/.

[38] G.I. Sayed, A. Tharwat, A.E. Hassanien, Chaotic dragonfly algorithm: an improved metaheuristic algorithm for feature selection, Appl. Intell. 49 (1) (2019) 188–205, https://doi.org/10.1007/s10489-018-1261-8. URL: http://link.springer.com/10.1007/s10489-018-1261-8.

[39] Shokooh Taghian, M.H. Nadimi-Shahraki, A Binary Metaheuristic Algorithm for Wrapper Feature Selection, Int. J. Comput. Sci. Eng. (IJCSE). doi: 10.13140/RG.2.2.34937.90722. URL: http://rgdoi.net/10.13140/RG.2.2.34937.90722.

[40] J. Piri, P. Mohapatra, An analytical study of modified multi-objective Harris Hawk Optimizer towards medical data feature selection, Comput. Biol. Med. 135 (2021), https://doi.org/10.1016/j.compbiomed.2021.104558. URL: https://linkinghub.elsevier.com/retrieve/pii/S0010482521003528 104558.

[41] J. Hu, W. Gui, A.A. Heidari, Z. Cai, G. Liang, H. Chen, Z. Pan, Dispersed foraging slime mould algorithm: Continuous and binary variants for global optimization and wrapper-based feature selection, Knowl.-Based Syst. 237 (2022), https://doi.org/10.1016/j.knosys.2021.107761. URL: https://linkinghub.elsevier.com/retrieve/pii/S0950705121009850 107761.

[42] B. Nouri-Moghaddam, M. Ghazanfari, M. Fathian, A novel multi-objective forest optimization algorithm for wrapper feature selection, Expert Syst. Appl. 175 (2021), https://doi.org/10.1016/j.eswa.2021.114737. URL: https://linkinghub.elsevier.com/retrieve/pii/S0957417421001780 114737.

[43] S. Akila, S. Allin Christe, A wrapper based binary bat algorithm with greedy crossover for attribute selection, Expert Syst. Appl. 187 (2022), https://doi.org/10.1016/j.eswa.2021.115828. URL: https://linkinghub.elsevier.com/retrieve/pii/S0957417421011933 115828.

[44] P. Srinivasa Rao, A. Sravan Kumar, Q. Niyaz, P. Sidike, V.K. Devabhaktuni, Binary chemical reaction optimization based feature selection techniques for machine learning classification problems, Expert Syst. Appl. 167 (2021), https://doi.org/10.1016/j.eswa.2020.114169. URL: https://linkinghub.elsevier.com/retrieve/pii/S0957417420309076 114169.

[45] W. Ali, A.A. Ahmed, Hybrid intelligent phishing website prediction using deep neural networks with genetic algorithm-based feature selection and weighting, IET Inf. Secur. 13 (6) (2019) 659–669, https://doi.org/10.1049/iet-ifs.2019.0006. URL: https://onlinelibrary.wiley.com/doi/10.1049/iet-ifs.2019.0006.

[46] A. Got, A. Moussaoui, D. Zouache, Hybrid filter-wrapper feature selection using whale optimization algorithm: A multi-objective approach, Expert Syst. Appl. 183 (2021), https://doi.org/10.1016/j.eswa.2021.115312. URL: https://linkinghub.elsevier.com/retrieve/pii/S0957417421007417 115312.

[47] Y. Zhang, D.w. Gong, X.z. Gao, T. Tian, X.y. Sun, Binary differential evolution with self-learning for multi-objective feature selection, Inform. Sci. 507 (2020) 67–85. doi: 10.1016/j.ins.2019.08.040. URL: https://linkinghub.elsevier.com/retrieve/pii/S0020025519307819.

[48] F. MiarNaeimi, G. Azizyan, M. Rashki, Horse herd optimization algorithm: A nature-inspired algorithm for high-dimensional optimization problems, Knowl.-Based Syst. 213 (2021), https://doi.org/10.1016/j.knosys.2020.106711. URL: https://linkinghub.elsevier.com/retrieve/pii/S0950705120308406 106711.

[49] A. Chaudhuri, T.P. Sahu, Binary Jaya algorithm based on binary similarity measure for feature selection, J. Ambient Intell. Humaniz. Comput. 13 (12) (2022) 5627–5644, https://doi.org/10.1007/s12652-021-03226-5. URL: https://link.springer.com/10.1007/s12652-021-03226-5.

[50] A. Chaudhuri, T.P. Sahu, A hybrid feature selection method based on Binary Jaya algorithm for micro-array data classification, Comput. Electr. Eng. 90 (2021), https://doi.org/10.1016/j.compeleceng.2020.106963. URL: https://linkinghub.elsevier.com/retrieve/pii/S0045790620308089 106963.

[51] M. Abdel-Basset, K.M. Sallam, R. Mohamed, I. Elgendi, K. Munasinghe, O.M. Elkomy, An improved binary grey-wolf optimizer with simulated annealing for feature selection, IEEE Access 9 (2021) 139792–139822, https://doi.org/10.1109/ACCESS.2021.3117853.

[52] M.A. Awadallah, A.I. Hammouri, M.A. Al-Betar, M.S. Braik, M.A. Elaziz, Binary Horse herd optimization algorithm with crossover operators for feature selection, Comput. Biol. Med. 141 (2022), https://doi.org/10.1016/j.compbiomed.2021.105152. URL: https://linkinghub.elsevier.com/retrieve/pii/S001048252100946X 105152.

[53] O.S. Qasim, N.A. Al-Thanoon, Z.Y. Algamal, Feature selection based on chaotic binary black hole algorithm for data classification, Chemometr. Intell. Labor. Syst. 204 (2020), https://doi.org/10.1016/j.chemolab.2020.104104. URL: https://linkinghub.elsevier.com/retrieve/pii/S0169743920303075 104104.

[54] Y. Chen, Q. Zhu, H. Xu, Finding rough set reducts with fish swarm algorithm, Knowl.-Based Syst. 81 (2015) 22–29, https://doi.org/10.1016/j.knosys.2015.02.002. URL: https://linkinghub.elsevier.com/retrieve/pii/S0950705115000337.

[55] E.A. Zaimoglu, N. Celebi, N. Yurtay, Binary-Coded Tug of War Optimization Algorithm for Attribute Reduction Based on Rough Set, Journal of Multiple-

Valued Logic and Soft Computing. URL: https://acikerisim.sakarya.edu.tr/handle/20.500.12619/96117.

[56] E.A. Zaimoglu, N. Çelebi, N. Yurtay, An Intelligent Feature Selection Method for Finding Rough Set Reducts Based on Tug of War Optimization Algorithm, in: C. Kahraman, S. Cebi, S. Cevik Onar, B. Oztaysi, A.C. Tolga, I.U. Sari (Eds.), Intelligent and Fuzzy Techniques in Big Data Analytics and Decision Making, Advances in Intelligent Systems and Computing, Springer International Publishing, Cham, 2020, pp. 1225–1234. doi: 10.1007/978-3-030-23756-1_144.

[57] X. Wu, V. Kumar, J. Ross Quinlan, J. Ghosh, Q. Yang, H. Motoda, G.J. McLachlan, A. Ng, B. Liu, P.S. Yu, Z.H. Zhou, M. Steinbach, D.J. Hand, D. Steinberg, Top 10 algorithms in data mining, Knowl. Inform. Syst. 14(1) (2008) 1–37. doi: 10.1007/s10115-007-0114-2. URL: http://link.springer.com/10.1007/s10115-007-0114-2.

[58] D. Albashish, A.I. Hammouri, M. Braik, J. Atwan, S. Sahran, Binary biogeography-based optimization based SVM-RFE for feature selection, Appl. Soft Comput. 101 (2021), https://doi.org/10.1016/j.asoc.2020.107026. URL: https://linkinghub.elsevier.com/retrieve/pii/S1568494620309650 107026.

[59] S. Sahran, D. Albashish, A. Abdullah, N.A. Shukor, S. Hayati Md Pauzi, Absolute cosine-based SVM-RFE feature selection method for prostate histopathological grading, Artif. Intell. Med. 87 (2018) 78–90, https://doi.org/10.1016/j.artmed.2018.04.002. URL: https://linkinghub.elsevier.com/retrieve/pii/S0933365717302026.

[60] S. Sarafrazi, H. Nezamabadi-pour, Facing the classification of binary problems with a GSA-SVM hybrid system, Math. Comput. Modell. 57 (1–2) (2013) 270–278, https://doi.org/10.1016/j.mcm.2011.06.048. URL: https://linkinghub.elsevier.com/retrieve/pii/S089571771100389X.

[61] X. Sun, Y. Liu, J. Li, J. Zhu, H. Chen, X. Liu, Feature evaluation and selection with cooperative game theory, Pattern Recogn. 45 (8) (2012) 2992–3002, https://doi.org/10.1016/j.patcog.2012.02.001. URL: https://linkinghub.elsevier.com/retrieve/pii/S0031320312000623.

[62] S.F. Da Silva, M.X. Ribeiro, J.D.E. Batista Neto, C. Traina-Jr., A.J. Traina, Improving the ranking quality of medical image retrieval using a genetic feature selection method, Decision Support Syste. 51(4) (2011) 810–820. doi: 10.1016/j.dss.2011.01.015. URL: https://linkinghub.elsevier.com/retrieve/pii/S0167923611000443.

[63] H.M. Zawbaa, E. Emary, C. Grosan, Feature Selection via Chaotic Antlion Optimization, PLOS ONE 11 (3) (2016), https://doi.org/10.1371/journal.pone.0150652. URL: https://dx.plos.org/10.1371/journal.pone.0150652 e0150652.

[64] O. Abdel-Raouf, M. Abdel-Baset, I. El-henawy, An Improved Chaotic Bat Algorithm for Solving Integer Programming Problems, Int. J. Modern Educ. Comput. Sci. 6 (8) (2014) 18–24, https://doi.org/10.5815/ijmecs.2014.08.03. URL: http://www.mecs-press.org/ijmecs/ijmecs-v6-n8/v6n8-3.html.

[65] A.A. Ewees, M.A. El Aziz, A.E. Hassanien, Chaotic multi-verse optimizer-based feature selection, Neural Comput. Appl. 31 (4) (2019) 991–1006, https://doi.org/10.1007/s00521-017-3131-4. URL: http://link.springer.com/10.1007/s00521-017-3131-4.

[66] G.I. Sayed, A.E. Hassanien, A.T. Azar, Feature selection via a novel chaotic crow search algorithm, Neural Comput. Appl. 31 (1) (2019) 171–188, https://doi.org/10.1007/s00521-017-2988-6. URL: http://link.springer.com/10.1007/s00521-017-2988-6.

[67] L. Abualigah, A. Diabat, Chaotic binary Group Search Optimizer for feature selection, Expert Syst. Appl. 192 (2022), https://doi.org/10.1016/j.eswa.2021.116368. URL: https://linkinghub.elsevier.com/retrieve/pii/S0957417421016626 116368.

[68] E. Bas, E. Ülker, An efficient binary social spider algorithm for feature selection problem, Expert Syst. Appl. 146 (2020), https://doi.org/10.1016/j.eswa.2020.113185. URL: https://linkinghub.elsevier.com/retrieve/pii/S0957417420300117 113185.

[69] S. Arora, P. Anand, Binary butterfly optimization approaches for feature selection, Expert Syst. Appl. 116 (2019) 147–160, https://doi.org/10.1016/j.eswa.2018.08.051. URL: https://linkinghub.elsevier.com/retrieve/pii/S0957417418305669.

[70] W.J. Conover, Practical nonparametric statistics, vol. 350, john wiley & sons, 1999.

[71] D.J. Sheskin, Handbook of parametric and nonparametric statistical procedures, Chapman and Hall/CRC, 2003.