Journal of Intelligent Systems and Internet of Things
Vol. 16, No. 01, PP. 86-101, 2025
DOI: https://doi.org/10.54216/JISIoT.160108

Enhanced Feature Selection Approach using Artificial Hummingbirds with Genetic Algorithm

Ismael Salih Aref [1,*], Dheyab Salman Ibrahim[1], Bashar Talib AL-Nuaimi[1]

[1]Department of Computer Science - College of Science University of Diyala, Iraq

Emails: asmaelsalih@uodiyala.edu.iq; dr.dheyab@uodiyala.edu.iq; alnuaimi_bashar@uodiyala.edu.iq

Abstract

Feature selection (FS) is a crucial preprocessing step in data mining to eliminate redundant or irrelevant features from high-dimensional data. Many optimization algorithms for FS often lack balance in their search processes. This paper proposes a hybrid algorithm, the Artificial Hummingbird Algorithm based on the Genetic Algorithm (AHA-GA), to address this imbalance and solve the FS problem. The main goal of AHA-GA is to select the most crucial characteristics to improve overall model categorization. The UCI datasets are used to assess the performance of the proposed FS method. The proposed feature selection algorithm was compared with five feature selection optimization algorithms: BWOAHHO, HSGW, WOA-CM, BDA-SA, and ASGW. AHA-GA achieved a classification accuracy of 96% across 18 datasets, which was higher than BWOAHHO (93.2%), HSGW (92.5%), WOA-CM (94.4%), BDA-SA (93%), and ASGW (91.6%). When comparing the proposed AHA-GA algorithm to the results obtained by the other five algorithms in terms of selected attribute size, the average feature sizes were as follows: AHA-GA (15.10889), BWOAHHO (16.74222), HSGW (19.43111), WOA-CM (17.05389), BDA-SA (17.275), and ASGW (19.7585). The statistical and experimental tests demonstrated that the proposed AHA-GA performs better than competitive algorithms in selecting effective features.

Received: November 15, 2024 Revised: January 14, 2025 Accepted: February 11, 2025

Keywords: Feature Selection; Artificial Hummingbirds algorithm; Genetic Algorithm; Classification

1. Introduction

Over the past decade, the dimensions of datasets have significantly increased due to the rapid expansion of information science. Additionally, many dataset attributes often contain redundant or unnecessary information, which negatively affects the performance of learning algorithms. As a result, it is crucial to apply dimensionality reduction techniques to address this issue [1]. The best subset of features from a given collection of all features is chosen via feature selection, a preprocessing approach used in data mining. It aims to make the classification model more straightforward, lessen undesired characteristics (redundant and superfluous information), and raise the machine learning model's accuracy of prediction [2].

In addition to lowering the classifier learning time and increasing classification accuracy, FS can lessen the dimensionality (space complexity) of features. Two goals must be met by feature selection: optimizing classification accuracy rate and decreasing number of features picked [1]. Feature selection is a significant step in data mining, and its role is to extract the most significant features, having an impact on classification model performance in a considerable way [3]. Feature selection (FS) has become one of the most often mentioned subjects in the fields of data mining and machine learning [4]. In general, it might be difficult to identify the ideal subset of traits. According to the previous literatures, the optimization algorithms have shown to be highly successful in solving a variety of feature selection problems [5]. Many heuristic search techniques for feature selection have been effectively evaluated in order to solve several feature selection challenges [6, 7]. Among these optimization algorithms are Grey Wolf Optimization (GWO), Particle Swarm Optimization (PSO), Genetic Algorithm (GA), Whale Optimization Algorithm (WOA), and so on [8, 9].

In general terms, it may be assumed that thorough verification of meta-heuristic algorithms in relation to overall FS models is still necessary. In fact, the optimization strategies that were first described frequently have performance issues when applied to the largest datasets. In general, optimization algorithms can become more efficient through hybridization and modification. In this field, integration of search policies with several optimizations approaches are presented [10] such as, whale optimization algorithm (WOA) with GWO (grey wolf optimization), simulation annealing (SA) with HHO, ring theory based evolutionary optimization algorithm, with harmony search (HAS) thermal exchange optimization (TEO) with seagull optimization algorithm, opposition-based learning (OBL) with social spider, tabu search (TS) with chemical reaction optimization (CRO) [4].

GA is also one of the algorithms that is hybridized with other algorithms. A heuristic technique to search and optimization mimics the natural evolution of humans. Genetic algorithms' foundation relies on reproducing new generations from older ones through the fundamental processes of inheritance, mating, and mutation [10]. Crossover factors depend on the data within populations to generate new solutions. However, the mutation is in charge of creating new information by altering a portion of the existing information.

A brief review of related works shows that GA has successfully addressed difficult optimization problems across various fields. Its performance generally surpasses that of traditional and contemporary alternatives, highlighting the effectiveness of its global search capability in providing competitive solutions for optimization challenges [11]. The Artificial Hummingbirds Algorithm (AHA) is an optimization technique inspired by the behaviours of hummingbirds. It tackles optimization problems by mimicking the unique flying abilities, strong memory, and effective foraging methods of hummingbirds [12].

In the suggested research, we introduced a hybrid approach named Hummingbird Search with a Genetic Algorithm (AHA-GA). This method merges the hummingbird search algorithm (AHA) with the genetic algorithm (GA) to solve FS issues. The AHA-GA algorithm combines the efficiency of AHA with the exploration capabilities of GA. Search results show that AHA-GA outperforms both traditional algorithms and recent alternatives. This algorithm achieves a balance between fast convergence, effective exploitation and exploration, and superior search performance.

In summary, the following are the planned work's primary contributions:

1. The proposed work leverages the significant benefits of the FSA algorithm by integrating it with GA to create a hybrid version called AHA-GA.
2. The effectiveness of AHA-GA is assessed on the Python platform using several metrics across 18 diverse UCI datasets.
3. The performance of AHA-GA is compared with different versions of popular population-based optimization algorithms such as BWOAHHO, WOA-CM.
4. The results demonstrate the superiority of the proposed AHA-GA in handling complex optimization problems, making it suitable for various real-world applications.

Other parts of the paper are sequenced chronologically. The second basin deeply explores the latest texts of diverse writers. Section 3 deals with Artificial Hummingbird (AHA) and Genetic Algorithms (GA). In Section 4, the method suggested is defined. Section 5 encompasses the discussion of experimental findings concerning the tested model. A brief presentation of the results and some new ideas for further research are available in the last part of Section 6.

2. Related Work

Based on a survey of literature, metaheuristics for features selection have improved numerous optimization techniques. For instance, on study [13] introduced a multimodal fusion and discriminant feature selection method utilizing a quantum-inspired genetic algorithm. Quantum-Inspired Genetic Algorithms (QIGAs) utilize the principles of quantum computing, such as qubits and superposition. QIGAs improve gene diversity by using qubits to represent solutions, enabling a linear superposition of states. Quantum rotation computing (QRC) is utilized to address challenges like computing complexity and early convergence in Genetic Algorithms (GA). QIGAs combines mutation for divergence and crossover for convergence to expanding diversity and improving population selection. Genetic algorithms (GA) use quantum rotation computing (QRC) to address problems including early convergence and computational complexity. By combining crossover to improve convergence and mutation to increase divergence, it maximizes population selection and increases variety.

In [14], The Binary Sparrow Search Algorithm with a Random Repositioning of Roaming Agents is proposed in a quest for both maximization of exploration and exploitation. As in SSA, feature selection can be facilitated with BSSA, but its performance is sensitive to the selection of a transfer function. Computational costs increase with the incorporation of a local search, and its use can extend only to classification problems. In [15] the Nonlinear Binary Grasshopper Whale

Optimization method (NL-BGWOA) is a hybrid method developed to address the FS problem. This technique has a unique position update approach that combines the whales' location changes with the grasshopper population to increase the variety of searches in the target domain. It works with medical datasets and show how (NL-BGWOA) solving real-world FS issues.

A particle swarm optimization algorithm (PLTVACIW-PSO) with parallelized linear time-variant acceleration coefficients and inertial weight is proposed in [16]. Its design has integrated the advantages of parallel computing with the combined strength of inertial weight (IW) and time-variant acceleration coefficients (TVAC). In [17] proposed the utilization of the BSMO binary optimizer method. The binary starling murmuration optimizer (SMO) is employed to identify an optimal subset of characteristics, addressing a wide range of complex engineering issues. BSMO employed two distinct methods to search medical datasets for the most significant characteristics. The first method involves creating binary versions of BSMO using various V-shaped and S-shaped transfer functions. The second method translates each dimension of a continuous solution provided by SMO to 0 or 1 using a configurable threshold.

In [18], they proposed using a binary moth-flame optimization (B-MFO) technique; the best features were chosen from medical datasets of various sizes. To enhance three different forms of B-MFO, U-shaped, and S-shaped transfer functions were applied, converting the canonical MFO from the continuous domain to the binary domain. The performance of these B-MFO variations was estimated using seven medical datasets and compared against four popular binary metaheuristic optimization algorithms: BPSO, BGWO, BDA, and BSSA.

In [19] the paper explores the constraints of the Moth-Flame Optimization (MFO) algorithm, a meta-heuristic method for addressing optimization challenges. The MFO algorithm, inspired by moths' transverse orientation navigation method, generates solutions for these problems. Despite its utility, MFO's performance is influenced by the flame production and spiral search elements, suggesting potential improvements in the many types of flames and the moths' solution-finding capabilities. The authors introduce an enhanced version named GMSMFO, which incorporates a Gaussian mutation mechanism and shrinks MFO to balance exploration and exploitation and increase population variety. The study assesses GMSMFO's effectiveness using 20 datasets and the CEC 2017 benchmark, including a dataset from a high-dimensional intrusion detection system. GMSMFO is examined using statistical tests like the Friedman and Wilcoxon rank-sum to compare its performance to other sophisticated metaheuristics. In the data, GMSMFO ranks the top while other algorithms not running at the same speed to it. Therefore, GMSMFO is very strong in finding the optimal feature subset that will increase classification accuracy and use less features.
The main contribution of this research paper includes the improvement of the exploration/exploitation balance and the expansion of the local search.

3. Preliminaries

3.1 Genetic algorithm

In recent years, metaheuristic algorithms have gained significant popularity for finding optimal solutions in various fields, including data mining. Data mining involves multiple steps, such as feature extraction and selection, to extract the best results. Consequently, Genetic algorithms are vital for distinguishing essential factors, which in turn help decision-makers to form a set of classifier examples. [20].

The genetic algorithm (GA) is inspired by biological evolution and is used for optimization. It involves key components: chromosome representation, fitness selection, and biologically inspired operators. Each solution is a chromosome, and each parameter is a gene. The fitness function evaluates each candidate's suitability. The roulette wheel mechanism randomly selects the best solutions to enhance weaker ones, avoiding local optima. Crossover and mutation processes are essential for improvement [21]. The main steps of GA are initializing the population, selecting the best solutions, performing crossover, and inducing mutation. [22].

The initial population is created using the Gaussian random distribution to increase variety. Each chromosome represents a potential solution, offering many options. The goal is to evenly disperse individuals across the search region, increasing overall population variability for more effective solutions [23]. Selection is a key part of genetic algorithms because it decides which individuals take part in the process of making new generations. Just like in nature, where the strongest individuals are more likely to get food and find a mate, genetic algorithms use a roulette wheel method to give probabilities to individuals based on how fit they are. This makes sure that the strongest individuals have a bigger role in making the next generation, just like how natural selection works [20].

Crossover operators create new solutions by merging two selected solutions from the previous phase, resulting in two fresh solutions. Single-point and double-point crossovers are the most popular approaches used in crossover operations. Moreover, the uniform crossover often yields superior results [23].

Mutation comes after the crossover process in genetic algorithms. Its purpose is to prevent all population solutions from converging to a local optimum of the current issue. A low mutation rate is maintained to avoid turning genetic algorithms into a basic random search. This step ensures that there are several alternatives and reduces the likelihood of local optima becoming the best ones [20].

3.2 Artificial-hummingbird-algorithm (AHA)

The AHA algorithm is inspired by the natural behaviours of hummingbirds. It tackles optimization challenges by emulating these behaviours to find the best solution among several options. This algorithm replicates the hummingbirds' flight abilities guided, territorial, and migratory foraging as well as their adeptness at locating food. By utilizing a visited table that mimics the memory of a hummingbird, the algorithm employs three distinct search strategies: omnidirectional, diagonal, and axial. Here are the fundamental concepts of AHA explained [12]:

3.2.1 Guided foraging

In AHA, the optimal food source is selected based on the highest nectar-refilling rate when the visit value in the visit table is identical. This mirrors the behaviour of hummingbirds, which prefer flowers with rapid nectar replenishment. Similarly, AHA search agents reposition themselves according to this chosen food source. Hummingbirds employ three distinct flight techniques to navigate towards their target, a concept that can be extended to a d-dimensional space.

These skills are described as follows [12]:

Axis flight skill: in this skill hummingbird fly in one axis coordinate.

$$D^i = \begin{cases} 1, & if\ i = randi([1,d]) \\ 0, & otherwise \end{cases} \quad i = 1..d \quad (1)$$

Where the function $randi([1, d])$ generates a random integer between 1 and d.

Diagonal flight skill: In this skill, hummingbirds fly from one corner to opposite corner of rectangle.

$$D^i = \begin{cases} 1, & if\ i = P(j), j \in [1,k] \\ 0, & otherwise \end{cases} \quad i = 1..d \quad (2)$$

In this ability, hummingbirds travel from one corner to the opposite corner of a rectangle.

where $k \in [2, r1.(d - 2)] + 1$ and P is determined by the following equation:

$$P = randperm(k) \qquad (3)$$

$randperm(k)$ generates a random permutation of numbers between 1 and k, and $r1$ is a randomly selected number within the range of 0 to 1.

Omnidirectional flight skill: In this skill, hummingbirds fly to any direction.

$$D^i = 1 \quad i = 1,..,d \qquad (4)$$

The hummingbird’s guided foraging behaviour is simulated using the following equation:

$$Px_i(t+1) = X_{i,tar}(t) + a.D.(X_i(t) - X_{i,tar}(t) \quad (5)$$

$$a \sim N(0,1)$$

Where $x\ i$ and $xtar$ represent the positions of the ith hummingbird and the target edible source at time t, respectively. The factor a follows a normal distribution N (0, 1) with a standard deviation of 1 and a mean of 0. The variable px_i (t+ 1) is used to store provisional values for the ith hummingbird at time t + 1.

$$X_i(t+1) = \begin{cases} X_i(t), & fit\ (X_i(t) \leq fit\ (px_i(t+1)) \\ px_i(t+1), & fit\ (X_i(t) > fit\ (px_i(t+1)) \end{cases} \quad (6)$$

where x_i (t + 1) and fit(.) indicate the position of ith hummingbird in time t+1 and fitness function respectively. Visit table is update after guided foraging.

3.2.2 Territorial foraging

Hummingbirds can relocate to a nearby food source after finishing their current one. AHA uses the following equations to simulate this behaviour [12].

$$px_i(t+1) = x_i(t) + b.D.x_i(t) \quad (7)$$

$$b \sim N(0,1) \quad (8)$$

where b is a distribution factor N (0, 1), characterized by a standard deviation of 1 and a mean of 0.

$$x_i(t+1) = \begin{cases} X_i(t), & fit\ (X_i(t) \leq fit\ (px_i(t+1)) \\ px_i(t+1), & fit\ (X_i(t) > fit\ (px_i(t+1)) \end{cases} \quad (8)$$

where x_i (t + 1) and fit(.) indicate the position of ith hummingbird in time t+1 and fitness function respectively. Visit table is update after traditional foraging.

3.2.3 Migration foraging

Hummingbirds will leave their current area when there is a shortage of food sources. AHA uses the following equations to simulate this behavior.

$$x_{wor}(t+1) = LB + r.(UP - LB) \quad (9)$$

where $x\ wor$ corresponds to the worst nectar-refilling rate among all of the edible sources present in the population.

The Adaptive Hummingbird Algorithm (AHA) begins with a random state for the visit table and hummingbird positions, then performs guided, territorial, or migration foraging as outlined in pseudocode 1 [12]. To improve the efficiency of the AHA algorithm in feature selection tasks, a novel method called the AHA-GA is presented in the next section. AHA-GA not only matches the complexity of AHA but also delivers better results.

3.3 Transfer Function

A transfer function represents discrete search regions by converting continuous values—like search areas—into a binary representation. In meta-heuristic-based feature selection (FS) techniques, this role is essential. It converts the answer into binary form by creating a probability value based on a position or solution's velocity or. In general, it converts continuous numbers into a range of [0, 1][24]. The transfer function used in this work is the sigmoid function.

$$Segmoid(x) = \frac{1}{1+e^{-x}} \quad (10)$$

where e is Euler's number. The following equation is used to put the values in two states, zero or one

$$x_d^{t+1} = \begin{cases} 1 & if\ sigmoid(x) > rand \\ 0 & otherwise \end{cases} \quad (11)$$

where $rand$ is random number in [0, 1]. The binary values are fed to the fitness function to be evaluated.

4. Proposed method

The feature selection problem is a binary optimization challenge, making binary representation the ideal choice. Here, a value of one indicates a feature that has been chosen, while a value of zero indicates a feature that has not been selected. The AHA-GA method is introduced to identify the optimal features that achieve the highest classification accuracy with the fewest features. AHA-GA merges the strengths of both AHA and GA. AHA-GA is the combination of AHA and GA.

Previous studies have indicated that the AHA algorithm is versatile and yields outstanding results across various applications. It has also been utilized for feature selection from a set of characteristics. However, the existing approach often falls short in finding the optimal solution when directly applied, particularly in multi-objective scenarios such as identifying the most important features with minimal numbers while upholding high classification accuracy. To overcome this limitation, a combined approach with other algorithms, notably the genetic algorithm, has been recommended.

This method combines several strategies to overcome the local minima. The creation and selection of the best solution sets, which the AHA then chooses, is greatly based on genetic algorithms. The roulette wheel is used to select parent hummingbirds for crossover. The newly created solutions are mutated for variety to increase the chance of finding the perfect global answer. The AHA not only expands the search space by leveraging the best solutions from the previous generation but also speeds up the discovery of the global solution by guiding the algorithm through optimal paths, rather than depending solely on random searches. The proposed phases and procedures of the algorithm are detailed in the following section.

4.1 Initial population

The population is generated randomly, where the Gaussian random distribution was used to generate random numbers because it increases the chance of diversity in generations. The population of hummingbirds contains multiple solutions, each of which represents the individual's chromosomes. Features are represented in each chromosome as binary vector (0,1) where 0 denotes an unselected feature and 1 means a chosen feature. The aim of initialization is to maximize the number of possibilities to find the optimal solution, which is made feasible by the population's vast variety and uniform distribution over the solution area.

4.2 Fitness function

All chromosomes in the solution space have been evaluated using the fitness function. Each chromosome in the solution space is assigned a fitness score based on the fitness evaluation. The fitness score determines whether a chromosome can be selected for reproduction in the next phase. The FS problem is also a bi-objective problem in the proposed model. The first objective of the feature selection process is to improve the accuracy of classification by determining the importance of features. The second goal is to reduce the number of features selected. The fitness function emerged as a result of these two opposing objectives, and it is expressed as follows [25]:

$$FF(Y) = X\ \delta_T(Y) + \mu\ \frac{|Y|}{|T|} \qquad (12)$$

where, FF(Y) denotes the fitness function, while δT(Y) represents the classification error rate. The term |Y| refers to the number of selected features in the solution Y, and |P| indicates the total number of features in the dataset. The constants χ and μ are parameters with values ranging from 0 to 1. A relationship between χ and μ is maintained, where μ equals $(1-\chi)$.

4.3 Parent Selection

At this stage, parent hummingbirds are selected for crossover. The hummingbirds are ranked by their efficiency, which is depicted by the size of their segment on a virtual roulette wheel. The roulette wheel selection method is used for this process, where the probability of selection corresponds to the varying sizes of the regions on the wheel. Subsequently, the top candidates are readied for the crossover operation, which is the following step.

4.4 Parent Selection

Crossover occurs between parents to produce offspring that possess new traits. the uniform crossover [23] outperforms better than single point crossover, so it is used in the suggested model. The crossover take place between the parents based on two masks. The first is generated randomly, while the second is taken from the inverse of the first mask. The length of the mask is equal to the length of the parents. The offspring are created with the first based on the first mask and the second based on the second mask. For each position in the offspring, if the corresponding bit in the binary crossover mask is 0, the bit is replicated from the second parent (or the first parent). If the corresponding bit in the binary crossover mask is 1, the bit is replicated from the first parent. Figure 1 show the crossover between parents.

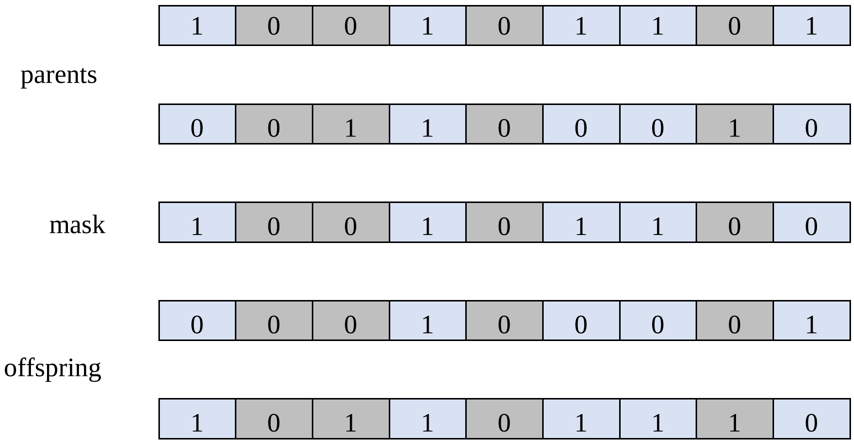


Figure 1. Crossover between parents.

4.5 Artificial Hummingbird Algorithm-Genetic Algorithm (AHA-GA)

The position of each hummingbird in the AHA algorithm suggests a potential feature selection method. The challenge's number of unique features is commensurate with the size of the optimization issue. After crossing produces child chromosomes, a local search is performed by looking at nearby regions in the solution space to identify better solutions within the AHA framework. As previously indicated, the hummingbird algorithm contains three tactics for searching using various flying ways. Hummingbirds use territorial, directed and migration foraging techniques to keep their location updated. The most crucial elements may be found and prioritized by using guided and territorial foraging, where the choice of characteristics has a major influence on the performance of the result, improving accuracy and efficiency.

The Artificial Hummingbird Algorithm (AHA) models the hummingbird's territorial and guided foraging using equations 555 and 666, respectively. Lastly, migration foraging is the third tactic. The emphasis is on adaptability and the necessity of modifying the feature set to preserve forecast accuracy and relevance over time. The final location is chosen by the hummingbirds' least advantageous position of migration to other areas. Hummingbird migratory foraging behaviour is modelled by the Artificial Hummingbird Algorithm (AHA) using equation (00 99). In AHA-GA, the last step completes the process, whereas the previous four steps are repeatedly executed to get the desired results. When the best option is determined using the input provided, the process is over. Figure 2 shows the flowchart and pseudocode steps of the proposed AHA-GA are illustrated in figures (1, 2).

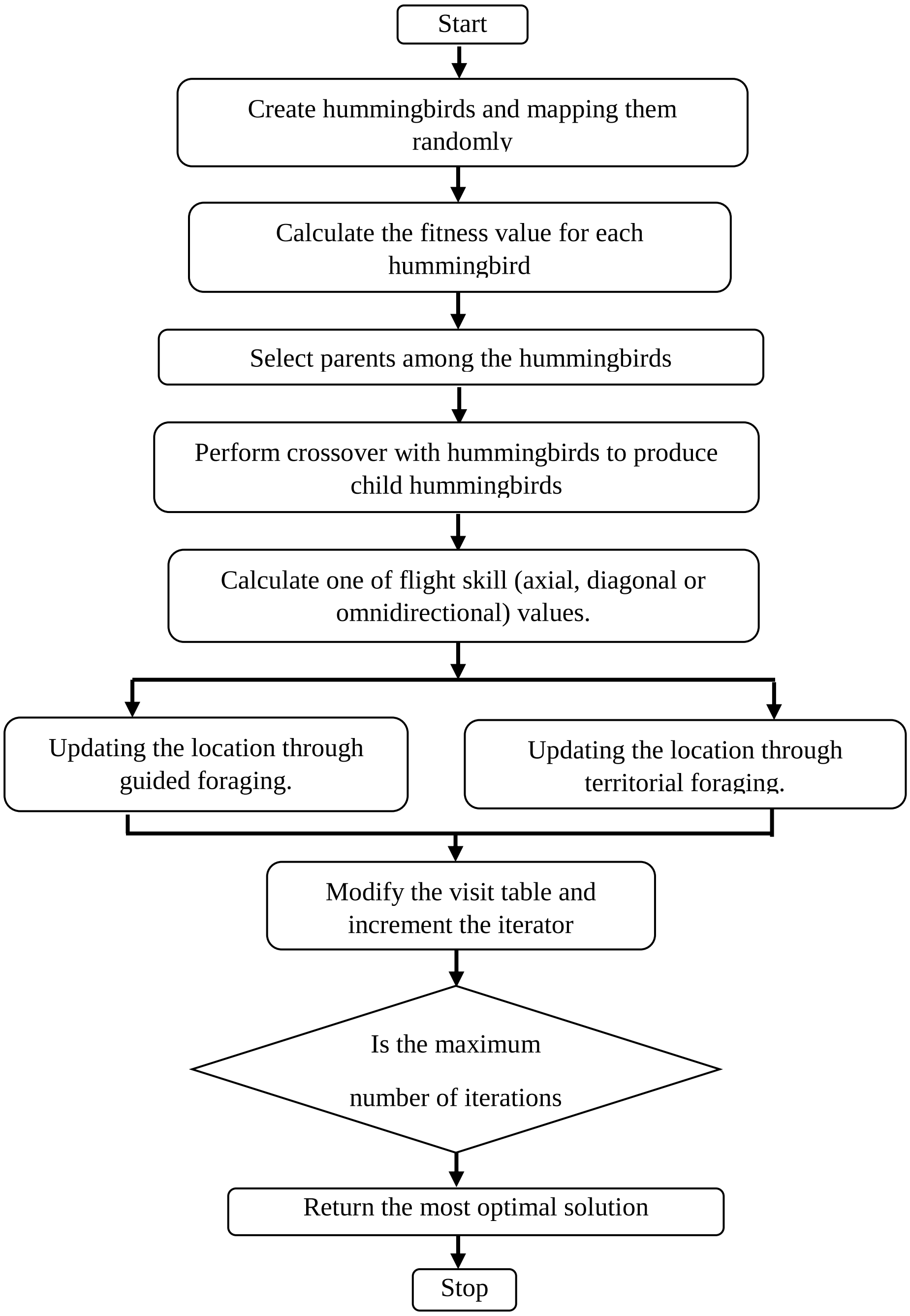


Figure 2. The suggested system steps

Input: algorithmic parameters, Features, size of the population, maximum iterations (MX)

Output: Best Selected features

Phase 1: Initialization

Calculate the fitness for each solution

Phase 2: Iterations N ← 1

For M → 1 to MX do

Achieve random selection of solutions and calculate the fitness

Do the crossover operator in order to obtain a new set of solutions.

Update the new list with the updated solution set.

Set the new solution set as the starting population in the new list for AHA.

Determine the current best option and rank the fitness value.

Update the hummingbirds' foraging location (guided, territorial and migration).

Check for hummingbirds that are out of boundaries

Calculate the fitness value for each hummingbird.

N←N+1

Return the best solution for feature selection

Stop

Figure 3. Pseudocode of the proposed AHA-GA for feature selection

5. Experimental outcomes and discussions

5.1 Experimental framework

This section provides a detailed explanation of the experimental results obtained from the proposed AHA-GA for FS. The performance of AHA-GA is assessed using the Python platform and various metrics across several UCI datasets. Table 1 shows the UCI dataset with number of attribute and objects for each one.

Table 1: UCI dataset with number of attribute and objects for each one.

Index	Datasets	No. of Objects	No. of Attributes
1	PenglungEW	73	325
2	Sonar	208	60
3	WaveformEW	5000	40
4	KrVsKpEW	3196	36
5	Ionosphere	351	34
6	BreastEW	569	30
7	SpectEW	267	22
8	Lymphography	148	18
9	CongressEW	435	16

10	Vote	300	16
11	Zoo	62	16
12	Exactly	1000	13
13	Exactly2	1000	13
14	M-of-n	1000	13
15	HeartEW	270	13
16	Wine	178	13
17	Tic-tac-toe	958	9
18	BreastCancer	699	9

The study examined suggested approaches utilizing the UCI datasets. The UCI dataset is divided into two sections, the first section for training (80%) and the second part for testing (20%). The K-nearest neighbour (KNN) classifier is used to determine the classification accuracy. This accuracy value is subsequently integrated into the fitness function (labelled as 12) to determine the fitness value. The fitness value obtained is then put back into the AHA-GA, which modifies the population's location according to the value acquired.

The proposed AHA-GA is compared with five algorithms BWOAHHO, HSGW, WOA-CM, BDA-SA and ASGW. The parameter settings of each one of these methods are given in Table 2.

Table 2: parameters setting for AHA-GA and others algorithms.

Algorithm	Parameter Settings
ASGW	β=0.5 , A$\in$[−2,2] , C$\in$[0,2]
HSGW	α = 0.99 , β= 0.01 ,a= [2 0]
WOA-CM	a = 2 to 0, a2 = -1 to -2
BWOAHHO	a = 0.1, d = 0.1, w = 0.005
BDA-SA	$\propto$ =0.99
AHA-GA	a = 0.1, d = 0.1, w = 0.005, r $\in$ [0, 1]

The datasets used for all five algorithms remain the same each time. However, the test and training datasets are randomly divided, resulting in 10 different executions for the split testing and training sets. The AHA-GA implementation is carried out on a system with a Core i7 CPU at 2.4 GHz and 8 GB of RAM.

5.2. Metrics of evaluation

The comparison of the algorithms is based on the following criteria:

*__Classification accuracy__: The first priority is to increase the classification accuracy, which is measured by KNN algorithm. This is done by removing irrelevant and redundant features.

* **Number of Selected Features** (**NSF**): The second priority is to reduce the number of selected features.

5.2.1 average classification accuracy

This section highlights the better performance of AHA-GA by comparing it with five state-of-the-art algorithms. The BWOAHHO, HSGW, WOA-CM, BDA-SA, and ASGW algorithms are among those that are compared. Classification accuracy is utilized to evaluate the effectiveness of the proposed method in each run of the algorithm, comparing it with state-of-the-art optimization algorithms. This metric includes applying the KNN classifier to the chosen feature set from the given dataset. Throughout the experiments, a population size of 25 and a maximum repetition count of 50 were chosen, with the experiment being repeated 10 times. Table 3 presents a comparison of AHA-GA with five state of-the-art algorithms based on their average classification accuracy (The bold items indicate the highest average classification accuracy achieved). The results presented in the table highlight the importance and capability of each algorithm in identifying and extracting the best features that lead to the highest classification accuracy.

Table 3: Average classification accuracy

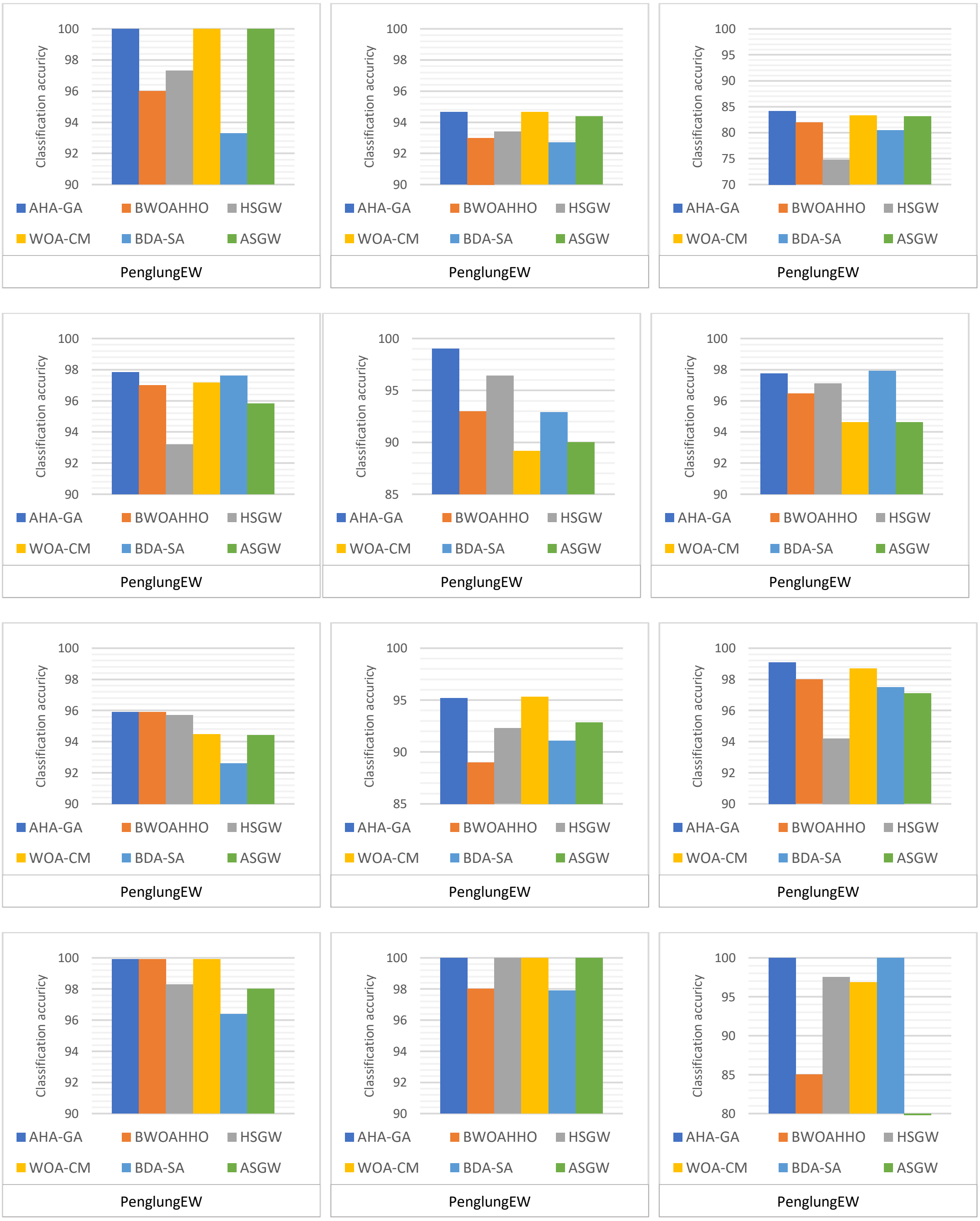
	Database name	**AHA-GA**	**BWOAHHO**	**HSGW**	**WOA-CM**	**BDA-SA**	**ASGW**
1	PenglungEW	**100**	96	97.3	**100**	93.3	**100**
2	Sonar	**94.65**	93	93.4	**94.65**	92.7	94.39
3	WaveFormEW	**84.2**	82	74.8	83.32	80.5	83.15
4	KrVsKpEW	**97.84**	97	93.2	97.18	97.6	95.82
5	Ionosphere	**99**	93	96.4	89.18	92.9	90
6	BreastEW	97.75	96.45	97.1	94.61	**97.92**	94.61
7	SpectEW	**95.9**	**95.9**	95.72	94.48	92.6	94.44
8	Lymphography	95.2	89	92.3	**95.33**	91.1	92.86
9	CongressEW	**99.1**	98	94.2	98.7	97.5	97.1
10	Vote	**99.9**	**99.9**	98.3	99.9	96.4	98
11	Zoo	**100**	98	**100**	**100**	97.9	**100**
12	Exactly	**100**	85	97.5	96.86	**100**	74.5
13	Exactly2	**81**	76	78.2	78.71	75.9	77
14	M-of-n	**99.8**	94	94.4	99.64	96.8	93.43
15	HeartEW	93.3	**94.8**	81.5	93.92	89.5	89.95
16	Wine	**100**	**100**	**100**	99.6	99.9	93.27
17	Tic-tac-toe	91.145	**91.6**	82.8	84.9	81.8	83.33
18	BreastCancer	98.9	**98.93**	98.6	98.57	98.8	98.27

AHA-GA outperforms BWOAHHO in 12 datasets based on average classification accuracy. In 3 other datasets, both methods yield equal results, while BWOAHHO performs better in 3 datasets. Consequently, BAHA demonstrates superior average classification accuracy over benchmark datasets compared to BWOAHHO. In two datasets, HSGW produced results that were identical, while in the other datasets, AHA-GA performed better. As such, AHA-GA outperforms HSGW in terms of average classification accuracy across all datasets.

AHA-GA outperforms WOA-CM in terms of average classification accuracy across ten datasets. In four additional datasets, both methods achieve equal results, while WOA-CM surpasses AHA-GA in only two datasets. Consequently, AHA-GA demonstrates superior average classification accuracy compared to WOA-CM.

AHA-GA gave the same results as BDA-SA in one dataset and exceeded BDA-SA in one other dataset. In the remaining 16 datasets, AHA-GA consistently delivered higher accuracy. ASGW has never outperformed AHA-GA in any dataset. AHA-GA shows better results in 14 datasets, with equal results in the remaining ones. Consequently, AHA-GA beats both BDA-SA and ASGW in average classification accuracy across all datasets. Overall, AHA-GA exhibits superior average classification accuracy compared to the five state-of-the-art algorithms evaluated.

The graphs of average classification accuracy according to the number of iterations for all datasets is given in figure 4, these graphs show the comparative performance of all six algorithms in the same conditions.



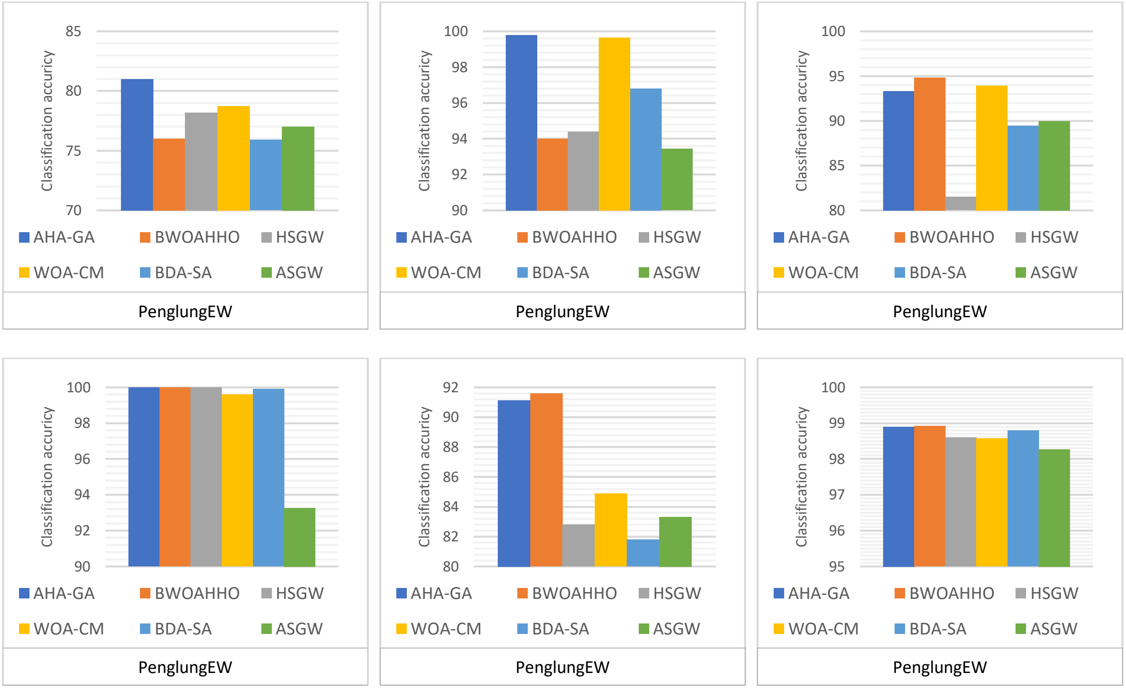


Figure 4. The average classification comparative results of all six algorithms in the same conditions.

Figure 4 explains how performance varies in various algorithms. The fruits of the AHA-GA algorithm's exceptional performance can be seen in its average score of 95.98, and it excels noticeably with a perfect prediction value for the PenglungEW dataset as well as the Zoo and Wine datasets. BWOAHHO is marginally worse on a dataset-by-dataset basis, with an average of 93.25 and works almost uniformly well across datasets but has shown variability, particularly seen in exactly, where it got a score of just ~85%. This is a good result that the HSGW algorithm achieves with an average of 92.54; however, there are datasets that have decreased numbers like WaveFormEW and Tic-tac-toe. Similarly, to DDU-CM, WOA-CM also exhibits strong performance and surpasses 92.02 on average. Other datasets, such as Zoo or Vote, are the best for this model. The averages of BDA-SA and ASGW are around 92.95 and 91.67 respectively; they demonstrate more competitive results but with inferior work on datasets like Exactly2 and HeartEW. This can be concluded that for all types of datasets, AHA-GA and WOA-CM are the algorithms, which maintain high consistency.

5.2.2 Selected features comparison

An important statistic to consider when assessing a feature selection algorithm is its ability to reduce the number of selected features, thereby eliminating redundant ones. The number of features tested affects the accuracy of classification. Therefore, it is necessary to conduct this test and show the results after reducing the number of features to increase the accuracy of classification results. In 18 datasets, Table 4 shows the average features selection achieved by AHA-GA and others algorithm.

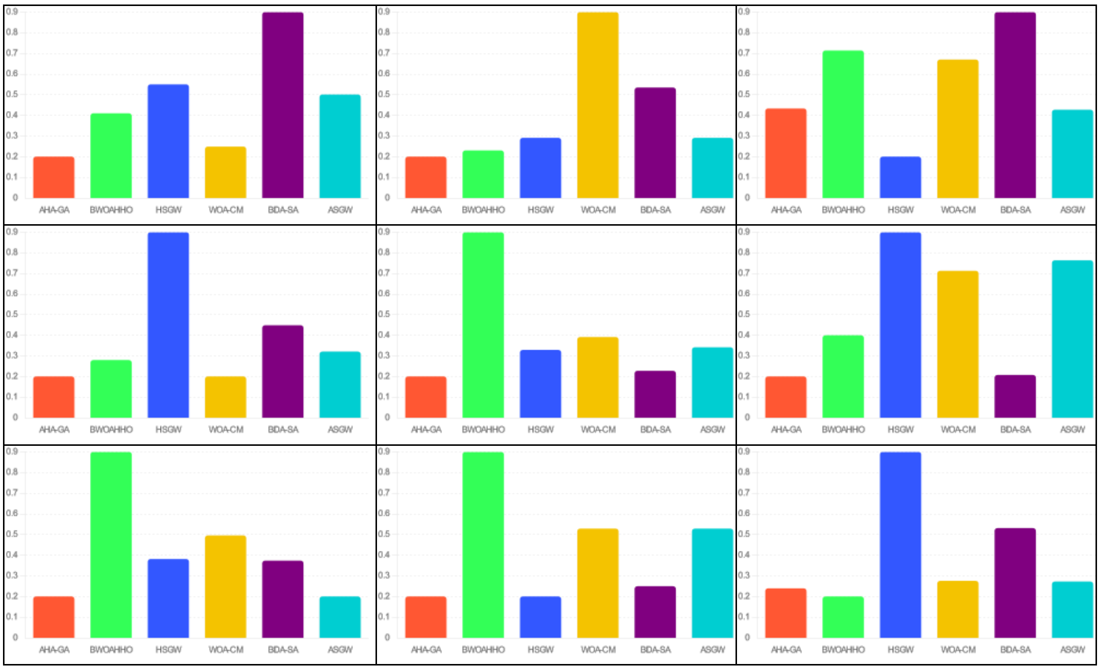
Table 4: Average features selection achieved by AHA-GA and others algorithm

	Database name	**AHA-GA**	**BWOAHHO**	**HSGW**	**WOA-CM**	**BDA-SA**	**ASGW**
1	PenglungEW	**123.4**	**123.4**	165.53	123.5	141.2	170.2
2	Sonar	**19.48**	25.00	24.30	23.57	23.90	25.30
3	WaveFormEW	**20.50**	22.30	21.93	22.70	21.10	23.83
4	KrVsKpEW	**19.10**	24.20	24.80	23.10	19.40	21.50
5	Ionosphere	**8.89**	9.70	11.17	11.43	10.40	12.30
6	BreastEW	**9.60**	**9.42**	11.67	14.43	11.45	15.83
7	SpectEW	9.30	9.70	10.23	**9.30**	**8.85**	10.17

8	Lymphography	**9.50**	9.73	10.56	11.86	12.50	11.200
9	CongressEW	**5.12**	5.30	8.87	7.29	5.15	8.83
10	Vote	**3.74**	3.51	7.57	3.95	5.43	3.93
11	Zoo	**5.30**	10.2	**5.30**	7.60	5.65	7.60
12	Exactly	**6.00**	8.70	6.70	7.14	6.67	**6.00**
13	Exactly2	**3.39**	5.00	9.03	7.52	3.45	7.93
14	M-of-n	**6.10**	9.90	6.8	7.14	6.25	6.87
15	HeartEW	5.87	6.20	8.77	**5.87**	6.90	6.37
16	Wine	5.97	7.70	4.53	7.43	8.85	**5.933**
17	Tic-tac-toe	**6.70**	6.80	7.00	9.00	7.80	7.00
18	BreastCancer	**4.00**	4.60	5.00	4.14	6.00	4.86

The outcomes from AHA-GA outperformed BWOAHHO by 16 datasets. However, in just one dataset, BWOAHHO defeated AHA-GA, and in one dataset, both performed equally. As a result, AHA-GA performs better than BWOAHHO in the average count of selected features across datasets.

All datasets accept one yielded equal result, with AHA-GA outperforming HSGW. For the average count of chosen features across datasets, AHA-GA outperforms HSGW therefore. AHA-GA beats WOA-CM in 16 datasets based on average NSF and in the others two dataset, they give equal results. WOA-CM has not had better results AHA-GA in any dataset. As a result, AHA-GA performs better than WOA-CM in average count of selected features across datasets. Based on the average NSF, AHA-GA outperforms BDA-SA in 17 datasets whereas BDA-SA obtains just one. Consequently, AHA-GA outperforms BDA-SA in terms of the average count of features selected across datasets. The figure 5 shows the NFS for different databases.



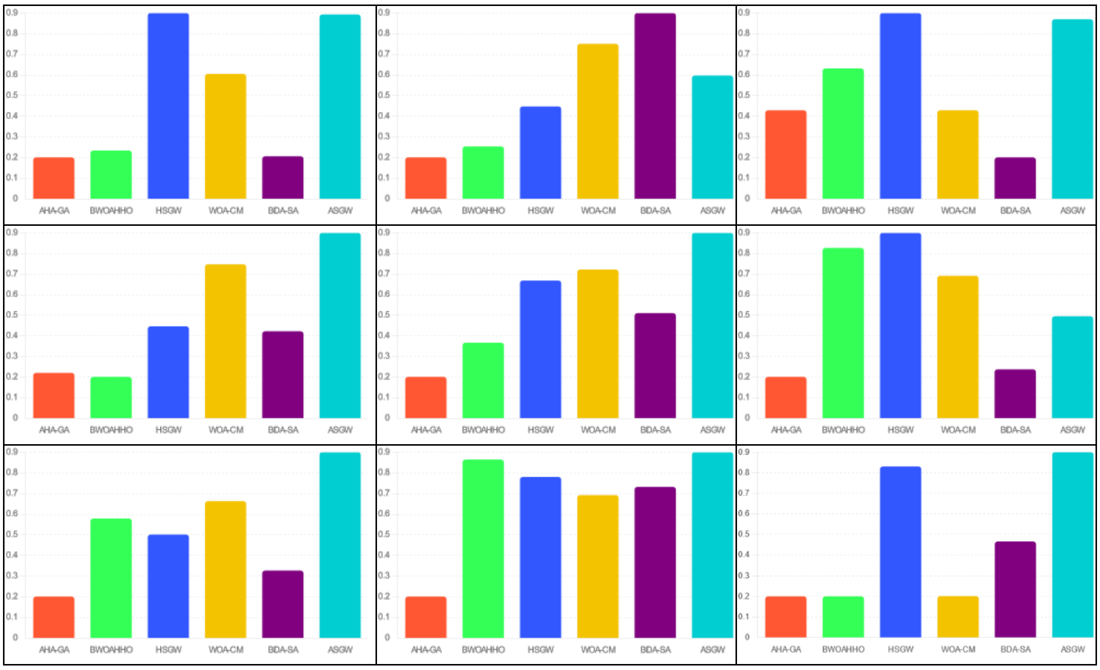


Figure 5. the average features selection achieved by AHA-GA and others algorithm.

It is clear from the figure that the proposed algorithm delivers the best results using the minimum possible number of features overall. Additionally, each database has features that differ in number and type from other databases, and this diversity contributes to the accuracy of the test. Therefore, we observe variations in the ranking of the best algorithms according to the database used. In general, some algorithms show good results despite the differences in databases, and among these algorithms, the proposed algorithm shows the best results compared to the other algorithms.

6. Conclusion

In this study, we employed a genetic algorithm to enhance the original AHA algorithm for solving the feature selection (FS) problem. The hybridized version (AHA-GA) helps prevent particles from becoming trapped in local optima and improves the convergence rate by balancing exploration and exploitation. Eighteen datasets from the UCI repository were used to validate the performance of the proposed method. Our results indicate that the AHA-GA outperformed other algorithms on most datasets. Among its competitors, the AHA-GA effectively identified a subset of highly discriminative features that accurately described the target concepts. Consequently, the AHA-GA often achieves the highest accuracy in feature selection tasks.

Future research could apply different classifiers to the features selected by the AHA-GA, such as support vector machines (SVM). Additionally, the proposed AHA-GA algorithm could be adapted for many problems, including engineering applications, industrial applications, image processing, spam email detection, and health data analysis.

References

[1] M. Abdel-Basset, D. El-Shahat, I. El-Henawy, V. H. C. De Albuquerque, and S. Mirjalili, "A new fusion of grey wolf optimizer algorithm with a two-phase mutation for feature selection," *Expert Syst. Appl.*, vol. 139, p. 112824, 2020.

[2] E. Hancer, B. Xue, D. Karaboga, and M. Zhang, "A binary ABC algorithm based on advanced similarity scheme for feature selection," *Appl. Soft Comput.*, vol. 36, pp. 334–348, 2015.

[3] C. F. Tsai, K. L. Sue, Y. H. Hu, and A. Chiu, "Combining feature selection, instance selection, and ensemble classification techniques for improved financial distress prediction," *J. Bus. Res.*, vol. 130, pp. 200–209, 2021.

[4] H. Chantar, M. Tubishat, M. Essgaer, and S. Mirjalili, "Hybrid binary dragonfly algorithm with simulated annealing for feature selection," *SN Comput. Sci.*, vol. 2, no. 4, p. 295, 2021.

[5] E.-G. Talbi, *Metaheuristics: From design to implementation*. John Wiley & Sons, 2009.

[6] H. K. Chantar and D. W. Corne, "Feature subset selection for Arabic document categorization using BPSO-KNN," in *Proc. 3rd World Congr. Nature Biologically Inspired Comput*, 2011, pp. 546–551.

[7] L.-Y. Chuang, C.-H. Yang, and J.-C. Li, "Chaotic maps based on binary particle swarm optimization for feature selection," *Appl. Soft Comput.*, vol. 11, no. 1, pp. 239–248, 2011.

[8] E. Emary, H. M. Zawbaa, and A. E. Hassanien, "Binary grey wolf optimization approaches for feature selection," *Neurocomputing*, vol. 172, pp. 371–381, 2016.

[9] H. Huang, H.-B. Xie, J.-Y. Guo, and H.-J. Chen, "Ant colony optimization-based feature selection method for surface electromyography signals classification," *Comput. Biol. Med.*, vol. 42, no. 1, pp. 30–38, 2012.

[10] R. K. Eluri and N. Devarakonda, "Feature selection with a binary flamingo search algorithm and a genetic algorithm," *Multimedia Tools Appl.*, vol. 82, no. 17, pp. 26679–26730, 2023.

[11] Y. Song, "Research on the application of computer graphic advertisement design based on a genetic algorithm and TRIZ theory," 2022.

[12] W. Zhao, L. Wang, and S. Mirjalili, "Artificial hummingbird algorithm: A new bio-inspired optimizer with its engineering applications," *Comput. Methods Appl. Mech. Eng.*, vol. 388, p. 114194, 2022.

[13] A. A. Abdulhussien, M. F. Nasrudin, S. M. Darwish, and Z. A. A. Alyasseri, "Feature selection method based on quantum inspired genetic algorithm for Arabic signature verification," *J. King Saud Univ. Comput. Inf. Sci.*, vol. 35, no. 3, pp. 141–156, 2023.

[14] A. G. Gad, K. M. Sallam, R. K. Chakrabortty, M. J. Ryan, and A. A. Abohany, "An improved binary sparrow search algorithm for feature selection in data classification," *Neural Comput. Appl.*, vol. 34, no. 18, pp. 15705–15752, 2022.

[15] L. Fang and X. Liang, "A novel method based on nonlinear binary grasshopper whale optimization algorithm for feature selection," *J. Bionic Eng.*, vol. 20, no. 1, pp. 237–252, 2023.

[16] A. T. Azar, Z. I. Khan, S. U. Amin, and K. M. Fouad, "Hybrid global optimization algorithm for feature selection," *Comput. Mater. Continua*, vol. 74, no. 1, pp. 2021–2037, 2023.

[17] M. H. Nadimi-Shahraki, Z. Asghari Varzaneh, H. Zamani, and S. Mirjalili, "Binary starling murmuration optimizer algorithm to select effective features from medical data," *Appl. Sci.*, vol. 13, no. 1, p. 564, 2022.

[18] M. H. Nadimi-Shahraki, M. Banaie-Dezfouli, H. Zamani, S. Taghian, and S. Mirjalili, "B-MFO: A binary moth-flame optimization for feature selection from medical datasets," *Comput.*, vol. 10, no. 11, p. 136, 2021.

[19] N. K. Hussein, S. Damas, M. A. Elaziz, A. Kamil, and A. Darwish, "Enhancing feature selection with GMSMFO: A global optimization algorithm for machine learning with application to intrusion detection," *J. Comput. Des. Eng.*, vol. 10, no. 4, pp. 1363–1389, 2023.

[20] L. Abualigah and A. J. Dulaimi, "A novel feature selection method for data mining tasks using hybrid sine cosine algorithm and genetic algorithm," *Cluster Comput.*, vol. 24, pp. 2161–2176, 2021.

[21] Z. Michalewicz and M. Schoenauer, "Evolutionary algorithms for constrained parameter optimization problems," *Evol. Comput.*, vol. 4, no. 1, pp. 1–32, 1996.

[22] A. Lambora, K. Gupta, and K. Chopra, "Genetic algorithm—A literature review," in *Proc. Int. Conf. Mach. Learn., Big Data, Cloud Parallel Comput. (COMITCon)*, 2019, pp. 380–384.

[23] T. Alam, S. Qamar, A. Dixit, and M. Benaida, "Genetic algorithm: Reviews, implementations, and applications," *arXiv preprint arXiv: 2007.12673*, 2020.

[24] Q. Al-Tashi, S. J. A. Kadir, H. M. Rais, S. Mirjalili, and H. Alhussian, "Binary optimization using hybrid grey wolf optimization for feature selection," *IEEE Access*, vol. 7, pp. 39496–39508, 2019.

[25] A. Hamdipour, A. Basiri, M. Zaare, and S. Mirjalili, "Baha: Binary artificial hummingbird algorithm for feature selection," *SSRN*, p. 4519771, 2023.