Vol. 12-2 (2025) 81-94
International Journal of Computers and Information (IJCI)
Available online at *https://ijci.journals.ekb.eg/*

Hybrid Beta-Henry Optimization: A Novel Feature Selection Approach

Shadia Elgazzar, Hatem Abdelkader, Asmaa Elsayed

Dept. of Information Systems, College of Computers and Information - Menoufia University, Shibin Al Kawm, Menoufia 32511, Egypt
shadia_elgazzar@hotmail.com, hatem.abdelkader@ci.menofia.edu.eg, asmaa.elsayed@ci.menofia.edu.eg

Abstract

Inspired by nature, the Henry Gas Solubility Optimization (HGSO) algorithm is a novel approach for solving global optimization problems by simulating Henry's Law of gas solubility. However, premature convergence and an imbalanced exploration-exploitation ratio remain significant challenges. HGSO's simple search strategy limits its ability to exploit optimal solutions effectively, hindering its performance on complex optimization problems. To address these limitations, we propose a beta hill climbing local search to enhance HGSO's performance. This novel beta operator improves upon traditional hill climbing by carefully balancing exploration and exploitation. By incorporating this operator, the resulting Enhanced HGSO (EHGSO) can more efficiently traverse the solution space and identify optimal solutions. We evaluated EHGSO on nine benchmark datasets and a real-world dataset. For the real-world dataset, we employed a Random Forest model, while for the benchmark datasets, we used a KNN model. EHGSO achieved an accuracy of 0.9467 on the real-world dataset and consistently outperformed other meta-heuristics like GOA, WOA, DA, GWO, and SSA on the benchmark datasets. These results demonstrate the superior optimization capabilities of EHGSO in tackling complex optimization problems.

Keywords: Henry Gas Solubility Optimization; Machine learning; Local Search; Optimization Algorithms

1. Introduction

Predictive models are developed using machine learning [1]. We should reduce input variables by doing this we can improve the computationally expensive modeling and model performance. Here lies the feature selection method. Feature selection (FS) [2] is one of the most important tasks in machine learning to enhance prediction accuracy and reduce overfitting by selecting effective features.

Meta-heuristic algorithms, a subclass of heuristic algorithms [3], inspired mainly from natural processes, have been introduced as effective remedies for these defects. In this context, these algorithms are considered a problem-solving and feature selection method for complex problems, such that they often outperform traditional methods both in terms of computational expenses and complexity barriers. We organized these algorithms into several groups, the evolutionary algorithms that are inspired by natural selection. as Genetic Algorithm (GA) [4], Evolutionary Strategies (ES) [5], and Differential Evolution (DE) [6]. Swarm Intelligence Algorithms, inspired by the collective behavior of animals. Such as Particle Swarm Optimization (PSO) [7], Ant Colony Optimization (ACO) [8], and Artificial Bee Colony (ABC) [9]. Physics-inspired algorithms inspired by physical phenomena. Such as Simulated Annealing (SA) [10], Gravitational Search Algorithm (GSA) [11], and Henry Gas Solubility Optimization (HGSO) [12].

Exploration and exploitation are two key concepts used to evaluate the optimization ability of a heuristic algorithm. Exploration, this refers to the algorithm's ability to search new, unexplored regions of the solution space. It's essential for discovering potentially better solutions. Exploitation, this refers to the algorithm's ability to refine promising solutions found during the exploration phase. It's important for converging to high-quality solutions. A well-balanced heuristic algorithm should effectively balance exploration and exploitation.

If an algorithm explores too much, it may fail to converge to good solutions. Conversely, if it exploits too much, it may get stuck in local optima.

Henry Gas Solubility Optimization (HGSO), inspired by physics phenomena, has provided some of the promising solutions to optimize various problems. In feature selection, HGSO performs well for high-dimensional problems based on Henry's law. While useful in terms of simplicity, it has the downside that can slow down convergence and have the potential for overuse. This paper's main contribution is to propose a new exploitation strategy in order to handle the limitations of the Henry gas optimization algorithm, which significantly enhances HGSO's capability of finding the best solutions.

Beta Hill Climbing (BHC) [13] is known for its effective exploitation capabilities, allowing it to precisely locate optimal solutions within local search spaces. This paper proposes a hybrid algorithm that integrates BHC's exploitation strategy into the Henry Gas Solubility Optimization (HGSO) algorithm.

The paper is organized as follows: Section 2 reviews related work, Section 3 presents the proposed hybrid algorithm, Section 4 analyses experimental results, Section 5 outlines future work, and Section 6 draws conclusions.

2. Related work

Motivated by the ideas of gas solubility, the Henry Gas Solubility Optimization (HGSO) algorithm is employed to address complex optimization problems. Nevertheless, HGSO can be hindered by an imbalance between exploration and exploitation, as well as premature convergence.

2.1 Original Henry Gas Solubility Optimization Algorithm

Henry's law, which is famous physics law states that the partial pressure of a gas is directly proportional to its solubility in a liquid at constant temperature, is the basis for the HGSO. The following stages make up the mathematical model that was created to explain the HGSO algorithm based on this idea.

Initialization phase: a population of N potential solutions—represented as gas particles—is randomly generated inside the specified search space using Equation (1), [12]:

$$x_i^0 = lb_i + \ rand_i \times (ub_i + lb_i) \tag{1}$$

The initial position of each gas particle, denoted as x_i^0 (i = 1, 2, ..., N), is bounded by the lower bounds lb_{i} and upper bounds ub_{i}, of the hyperspace, respectively. Symbolizes a real number generated at random in the interval [0, 1]. Other attributes of each gas particle are also initialized using Equation (2), [12]:

$$H_j^0 = l_1 + \ rand_1, P_{i.j}^0 = \ l_2 \times rand_2, C_j^0 \ = \ l_3 \times rand_3 \tag{2}$$

For instance C_j^0 indicates the initial constant value of type j, $p_{i.j}^0$ indicates the initial partial pressure of gas i in cluster j, and H_j^0 indicates the starting value of Henry's constant for type. The constants l_1,l_2, and l_3 are contained in equation (2) and have values of 5E-02, 100, and 1E-02, respectively.

Clustering phase: The gas particles are divided into k clusters on the basis of their gas type. Henry's constant (Hj). is assigned a unique value for each cluster.

Evaluation phase: The fitness of each gas particle within its respective cluster is evaluated to assign the best cluster $X_{j.best}$. All candidate solutions are ranked based on their fitness values. Every potential solution is scored based on its fitness values, and the population's best solution is identified based on its fitness.

Updating Henry's Coefficient phase: Since the pressure on gas particles changes with each repeat, it is essential to update Henry's coefficient for each cluster j. This update is calculated using equation (3), [12]:

$$H_j^{t+1} = H_j^t \times exp\left[-c_j \times \left(\frac{1}{T^t} - \frac{1}{T^\theta}\right)\right]. T^t = \exp\left(\frac{-t}{t_{max}}\right) \quad (3)$$

Where H_j^t represents Henry's coefficient for cluster j in iteration t, T^t denotes the temperature at iteration t, T^θ is a constant with a value of 298.15, and is the maximum number of iterationst_{max}.

Updating Gas Particle Solubility phase: during the *t*th iteration, the solubility S_{ij}^t of the *i*th gas particle in the *j*th cluster must be updated using Equation (4), [12]:

$$s_{i.j}^t = \mathrm{K} \times H_j^{t+1} \times P_{i.j}^t \quad (4)$$

Where *K* is a constant, and $P_{i.j}^t$ is the partial pressure on *i*th gas particle in *j*th cluster.

Update Gas Particle Position phase: the position of the *i*th gas particle in the *j*th cluster for iteration t = 1 + 1 is updated using Equation (5) ,[12]:

$$\begin{gathered} \mathrm{x}_{\mathrm{i.j}}^{\mathrm{t+1}} = \mathrm{x}_{\mathrm{i.j}}^{\mathrm{t}} + \mathrm{F} \times \mathrm{rand}_1 \times \gamma \times \left(\mathrm{x}_{\mathrm{j.best}} - \mathrm{x}_{\mathrm{i.j}}^{\mathrm{t}}\right) + \mathrm{F} \times \mathrm{rand}_2 \\ \times \alpha \times \left(\mathrm{S}_{\mathrm{i.j}}^{\mathrm{t}} \times \mathrm{X}_{\mathrm{best}} - \mathrm{X}_{\mathrm{i.j}}^{\mathrm{t}}\right). \\ \text{Where } \gamma = \beta \times \exp\left(-\frac{\mathrm{F}_{\mathrm{best}+\epsilon}^{\mathrm{t}}}{\mathrm{F}_{\mathrm{i.j}+\epsilon}^{\mathrm{t}}}\right). \varepsilon = 0.05 \end{gathered} \quad (5)$$

Where F is used to control search direction by flagging, γ is the ability of a gas particle with respect to its cluster, and α is the influence of other gas particles on *ith* particle. In Equation (5), $rand_1$ and $rand_2$ are two different randomly generated real values between [0, 1], and ε is a small value to avoid the divide by zero error.

Escape from local optimum: HGSO selects the worst solutions, as in Equation (6) , [12], for re-initialization in order to implement the strategy of avoiding local optima problems:

$$\mathrm{N_w} = \mathrm{N} \times [\mathrm{rand} \times (\mathrm{C_2} - \mathrm{C_1}) + \mathrm{C_1}]. \mathrm{C_1} = 0.1 \text{ and } \mathrm{C_2} = 0.2 \quad (6)$$

Where *N* denotes the population size and *rand* denotes a random number between [0, 1]. Equation (1) is used to re-initialize the position of the worst solutions chosen in this step.

3. Proposed Work

The proposed enhanced HGSO algorithm, which incorporates more efficient Beta Hill Climbing local search techniques to address the challenges of feature selection and optimization in high-dimensional datasets, was evaluated using a comprehensive framework as illustrated in Fig. 1. The steps involved in the evaluation process are outlined below:

3.1 Dataset and parameter settings

The first step in evaluating our model is choosing the dataset. Reputable benchmark datasets were used. These datasets are materials that are publicly accessible or may be requested; they have been deliberately published as datasets. It is meant for assessment purposes and includes well-defined evaluation methodologies. The datasets represent the complexities and challenges of real-world scenarios. They usually involve complicated problems or large datasets that push the boundaries of models and algorithms, leading to advancements in domains such as machine learning, computer vision, and natural language processing. Table 1 gives a detailed description of the specific datasets used in this study. From the UCI Machine Learning Repository, the datasets are available for download. These datasets are readily available and can be downloaded from (https://archive.ics.uci.edu/ml/index.php). Table 2 summarizes the parameter settings for all algorithms used in this study.

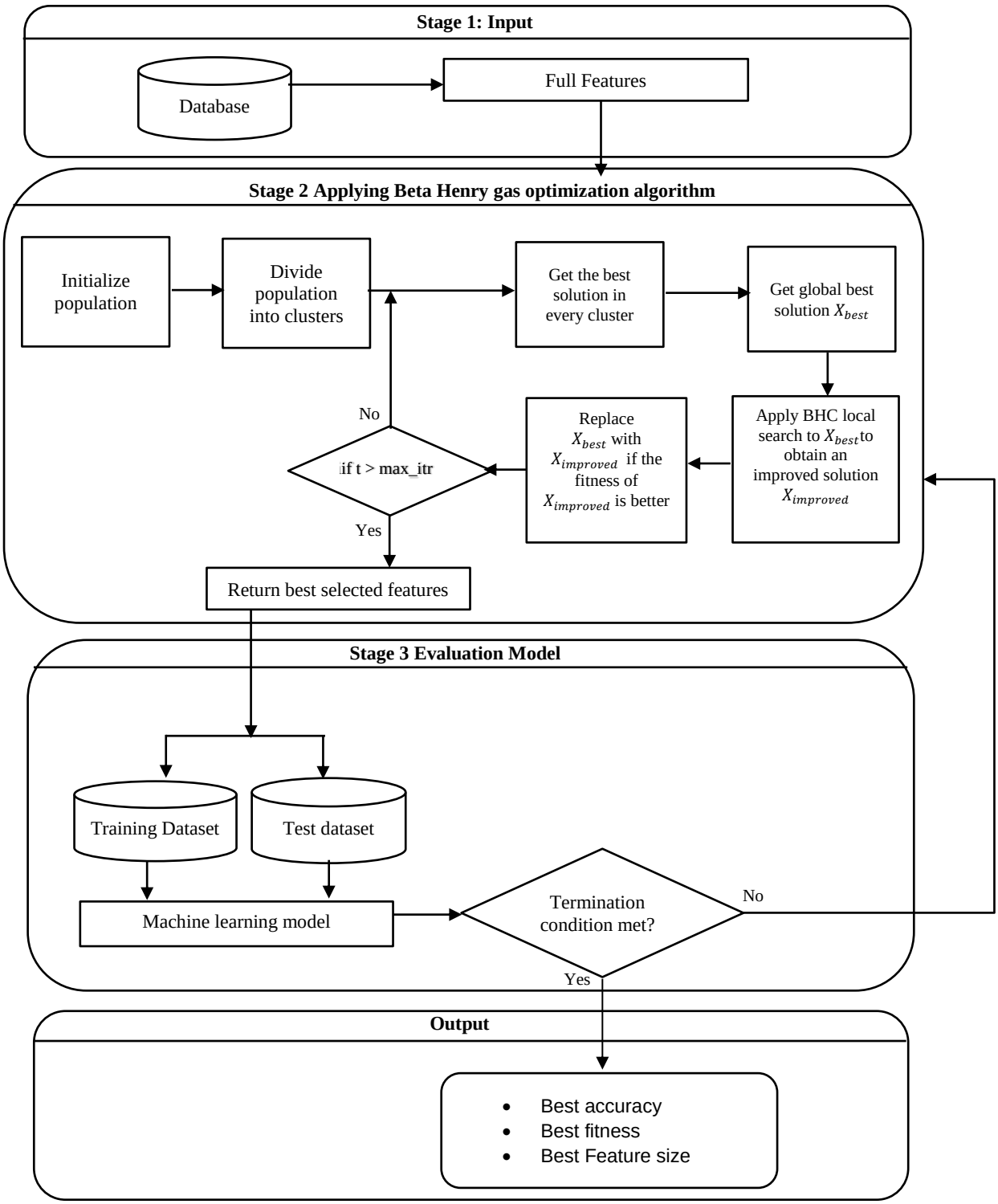

Fig 1. Proposed framework

Table1 *Datasets Description*

No.	Dataset	No.of features	No.of instances	No.of classes
1	CongressEW	16	435	2
2	IonosphereEW	34	351	2
3	HeartEW	13	270	2
4	BreastEW	30	569	2
5	Lymphography	18	148	4
6	Exactly2	13	1000	2
7	SonarEW	60	205	2
8	SpectEW	22	267	2
9	Spambase	58	4601	2

Table 2 *Algorithms Parameters Setting*

algorithm	parameters
β-HGSO	Gases Number (N = 10) Maximum number of iterations (100) Cluster number = 2 M1 and M2 = 0.1 and 0.2 β = 1, α = 1 and K = 1 l1, l2, l3 are constants with values equal 5E−03, 100, 1E−02 l1, l2, l3 are constants with values equal 1, 10, 1 Beta = 0.5,1.0 bw = 0.5,0.05,0.005
HGSO	The same as β-HGSO
WAO	a decreases linearly from 2 to 0 [t] b = 1 Maximum number of iterations (100) Dimension corresponds to the number of features
GWO	a decreases linearly from 2 to 0 Maximum number of iterations (100) Dimension corresponds to the number of features
GOA	C max = 1 & C min = 0.00004 Maximum number of iterations (100) Dimension corresponds to the number of features
DA	Maximum number of iterations (100) [t] Dimension corresponds to the number of features
SSA	c 1 and c 2 are randomly distributed Maximum number of iterations (100) Dimension corresponds to the number of features

3.2 **Improved Henry Gas Algorithm for Solubility Optimization**

The ability of the Henry Gas Solubility Optimization (HGSO) algorithm to refine the global best solution can be greatly improved by incorporating Beta Hill Climbing (BHC) [13] as a local search method. The β-hill climbing algorithm (BHC) is a novel and a very competent explorative local search algorithm. The algorithm proceeds with an arbitrary solution to a specified issue $x = (x_1\ x_2\ \ldots\ldots x_N)$. Neighbourhood navigation or the N operator and β-operator are the two operators that BHC incorporates iteratively to produce new random solutions $x' = (x_1'\ x_2'\ \ldots\ldots x_N')$.

During every iteration of the BHC algorithm, the N operator uses the improve (N(x)) function along with 'random walk' acceptance rule for the determination of a neighboring solution of x, using. Equation (7), [13]:

$$x_1' = x_i \pm \mathrm{U}\ (0,1) \times \mathrm{bw}\ \ \exists! \in [1, \mathrm{N}] \tag{7}$$

Note that i is randomly selected within the dimensionality range of the solution, $i \in [1,2, \ldots. N]$. Furthermore, for the selected dimension i, a parameter known as bw (bandwidth) establishes the range permitted for the difference between the current value and the new value.

The β-operator plays a crucial role by guiding the exploration of the search space. During this stage, the values for the new solution's variables are determined based on a parameter β (beta) that lies between 0 and 1. Or depending on the values in the current solution. According to Equation (8), [13]:

$$x_i' \leftarrow \begin{cases} x_r & rnd \leq \beta \\ x_i & otherwise \end{cases} \tag{8}$$

Where *rnd* generates a uniform random number between 0 and 1, and $x_r \in x_i$ is the decision variable

Convergence towards the best solution in the β-Hill Climbing algorithm depends on two basic operators: the β-operator, which accelerates the optimization process by altering specific components of the current solution. And the N-operator that randomly selects a neighbouring solution with a higher objective value.

In this process, the β-operator serves as the source of exploration, while the N-operator is primarily responsible for exploitation.

This hybrid approach involves HGSO identifying a promising candidate solution that appears to be close to the optimum solution. Then BHC is applied to this candidate solution to refine it further within its neighbourhood.

BHC operates by iteratively perturbing the solution using a mutation operator. This mutation rate, controlled by a beta parameter, gradually decreases over time. Initially, a higher mutation rate allows for wider exploration of the solution space. As the search progresses, the mutation rate decreases, focusing on more targeted exploration of promising regions. In each iteration, BHC generates a new solution by slightly modifying the current best solution. If this new solution yields a better objective function value, it replaces the current best solution. This process continues until a stopping criterion is met, such as reaching a maximum number of iterations or a satisfactory solution quality.

Algorithm 1 Beta-HGSO pseudo-code

Initialize HGSO and BHC parameters:
Divide the population into clusters based on gas types with the same Henry's constant value (Hj)
Calculate the fitness value of each search agent in the population
Identify the best gas $X_{j,best}$ in each cluster and the overall best search agent $X_{j,best}$ according to fitness values
Initialize iteration counter t = 0
while t < maximum number of iterations do:
    Update the position of all agents using Eq. (5)
    Update Henry's coefficient for each gas type using Eq. (3)
    Update the solubility of each gas type using Eq. (4)
    Rank all agents based on their fitness and select the worst-performing agents using Eq. (6)
    Update the position of the worst-performing agents using Eq. (1)
    Apply BHC local search to X_{best} to obtain an improved solution $X_{improved}$ using Eq. (7)
    If the fitness of $X_{improved}$ is better than X_{best}
        Replace X_{best} with $X_{improved}$
    t = t + 1
end while
Return X_{best} as the final optimized solution

3.3 **Classification Model**

Once the improved HGSO process is finished, it returns an optimal solution. It keeps only the best features in the original dataset. To evaluate classification performance, we employ a hold-out strategy, randomly partitioning the dataset into 80% training data and 20% testing data. To assess the accuracy, we employ two classification algorithms: The k-NN [14] algorithm, with a fixed value of k = 5, is used to evaluate the performance, which is used to evaluate benchmark datasets. And a Random Forest classifier [15] is trained on the training set, and its predictions are compared against data in the testing set, which is used for the real-world dataset. To obtain statistically significant results, each experiment is repeated ten times independently.

4. Results Analysis and Discussion

Each experiment utilizing Beta-HGSO was carried out separately ten times to guarantee accurate experimental results. Every algorithm was configured to execute for 100 iterations. Table 2 provides a summary of the parameters used for both the original algorithm and Beta-HGSO. The following is a summary of our experiment results.

4.1 Comparison of β-HGSO with other metaheuristics for benchmark datasets

We thoroughly compared our improved HGSO algorithm against six swarm intelligence algorithms: original Henry Gas Optimization (HGSO), Grasshopper Optimization Algorithm (GOA) [16], Dragonfly Algorithm (DA) [17], Whale Optimization Algorithm (WOA) [18], Grey Wolf Optimizer (GWO) [19], and Salp Swarm Algorithm (SSA) [20]. To assess its efficacy, we employed a K-Nearest Neighbours (KNN) machine learning model. The results of mean accuracy, mean fitness, and mean selected features for these older algorithms on the first six datasets can be found in detail in Tables 6, 8, and 10 of paper "An efficient henry gas solubility optimization for feature selection "[12] (pages 9, 10, and 11, respectively). Tables 3, 4, and 5 in this paper present the comparative results of β-HGSO against these older algorithms.

Table 3 *Table Accuracy Comparison of βeta HGSO and Other Algorithms (k -NN classifier)*

Datasets	Metric	β-HGSO	HGSO	DA	WAO	GAO	GWO	SSA
HeartEW	AVG	**0.9065**	0,8778	0,8043	0,8111	0,7691	0,7840	0,8167
	STD	0.013	**0,0104**	0,0186	0,0197	0,0314	0,0250	0,0132
BreastEW	AVG	**0.96**	0,9532	0,9444	0,9509	0,9257	0,9386	0,9564
	STD	**0.0058**	0,0074	0,0120	0,0145	0,0132	0,0132	0,0091
Exactly2	AVG	0.7845	**0,8152**	0,7123	0,7160	0,6792	0,7015	0,7270
	STD	0.0129	**0,0009**	0,0139	0,0126	0,0206	0,0196	0,0114
CongressEW	AVG	**0.991**	0,9839	0,9732	0,9797	0,9556	0,9556	0,9556
	STD	**0.0073**	0,0094	0,0114	0,0089	0,0153	0,0123	0,0071
Lymphography	AVG	0.9133	0,9114	0,8956	0,9122	0,8122	0,8811	**0,9300**
	STD	**0.0163**	0,0330	0,0358	0,0406	0,0450	0,0388	0,0332
IonosphereEW	AVG	0.9157	**0,9737**	0,9169	0,9390	0,8892	0,9155	0,9385
	STD	**0.0126**	0,0151	0,0150	0,0154	0,0137	0,0161	0,0136
SonarEW	AVG	**0.9586**	0.9286	------	------	------	------	------
	STD	**0.0124**	0.0151	------	------	------	------	------
SpectEW	AVG	**0.8196**	0.784	------	------	------	------	------
	STD	0.028	**0.0196**	------	------	------	------	------
Spambase	AVG	**0.9204**	0.903	------	------	------	------	------
	STD	0.0059	**0.0039**	------	------	------	------	------
Over All Accuracy		**0.9077**	0.8719					

As shown in Table 3, β-HGSO, combined with a k-NN classifier, consistently demonstrates superior average accuracy on six out of nine datasets (HeartEW, BreastEW, CongressEW, SonarEW, SpectEW, and Spambase) under identical experimental conditions. HGSO and SSA exhibit the best performance on two (IonosphereEW, Exactly2) and one (Lymphography) dataset, respectively. Notably, β-HGSO achieves the highest overall accuracy of 90.77%, significantly outperforming HGSO's 87.19%. Figures 1 and 2 provide additional visual insights.

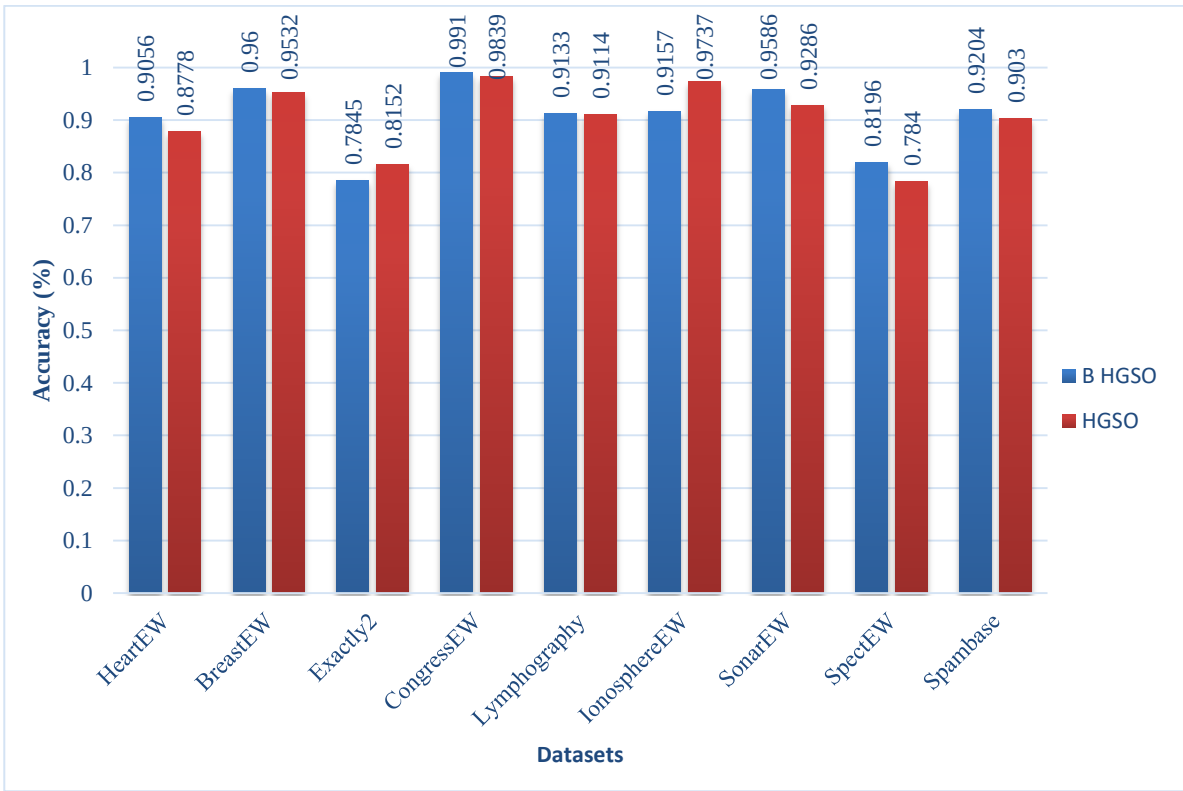

Fig 2. Average accuracy Comparison of βeta HGSO and HGSO based on KNN classifier

,

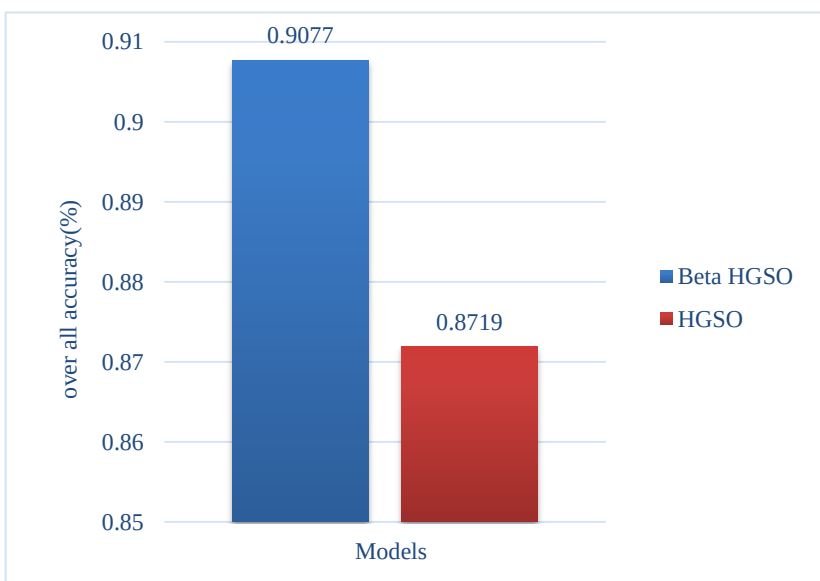

Fig 3. The overall accuracy of Beta HGSO and HGSO based on KNN classifier

Table 4. *Fitness Comparison of βeta HGSO and Other Algorithms (k -NN classifier)*

Datasets	Metric	B HGSO	HGSO	DA	WAO	GAO	GWO	SSA
HeartEW	AVG	**0.0983**	0.1263	0.1994	0.1926	0.2339	0.2184	0.1861
	STD	0.0122	**0.0103**	0.0179	0.0189	0.0309	0.0245	0.0126
BreastEW	AVG	**0.044**	0.0488	0.0591	0.0516	0.0786	0.0642	0,0468
	STD	**0.0054**	0.0069	0.0121	0.0146	0.0133	0.0130	0.0089
Exactly2	AVG	0.218	**0.1838**	0.2906	0.2871	0.3230	0.3014	0.2764
	STD	0.0122	**0.0005**	0.0134	0.0122	0.0207	0.0184	0.0106
CongressEW	AVG	**0.0119**	0.0171	0.0305	0.0232	0.0488	0.0342	0.0191
	STD	**0.0055**	0.0090	0.0119	0.0092	0.0156	0.0124	0.0077
Lymphography	AVG	0.0905	0.0916	0.1076	0.0908	0.1908	0.1218	**0.0731**
	STD	**0.0158**	0.0331	0.0361	0.0408	0.0445	0.0386	0.0332
IonosphereEW	AVG	0.087	**0.0275**	0.0861	0.0627	0.1144	0.0865	0.0642
	STD	**0.0122**	0,0149	0.0150	0.0153	0.0137	0.0161	0.0139
SonarEW	AVG	**0.0458**	0.0755	------	------	------	------	------
	STD	**0.0121**	0.0148	------	------	------	------	------
SpectEW	AVG	**0.1825**	0.2182	------	------	------	------	------
	STD	0.0276	0.0192	------	------	------	------	------
spam	AVG	**0.084**	0.1008	------	------	------	------	------
	STD	0.0056	**0.0038**	------	------	------	------	------
Over All Fitness		**0.0958**	0.0988	------	------	------	------	------

Table 4 shows that β-HGSO with a k-NN classifier achieves lower average fitness values on five out of nine datasets. HGSO, the second-best optimizer, only outperforms it slightly on two datasets: Exactly2 and IonosphereEW, with margins of 0.1838 and 0.0275, respectively. β-HGSO achieves the lowest overall fitness of 0.0958, significantly outperforming HGSO's 0.0988. Figures 4 and 5 provide additional visual insights.

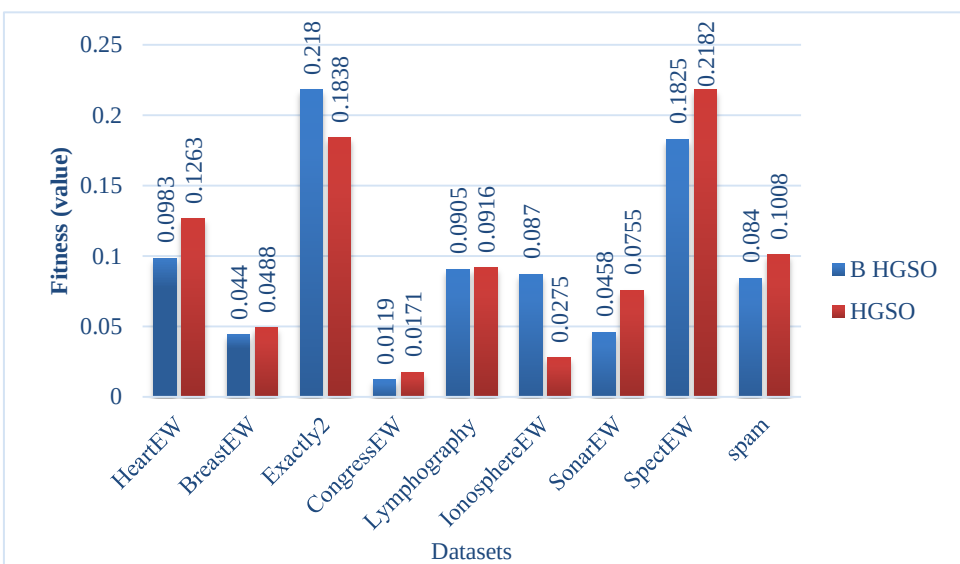

Fig 4. Average fitness comparison of βeta HGSO and HGSO based on KNN classifier

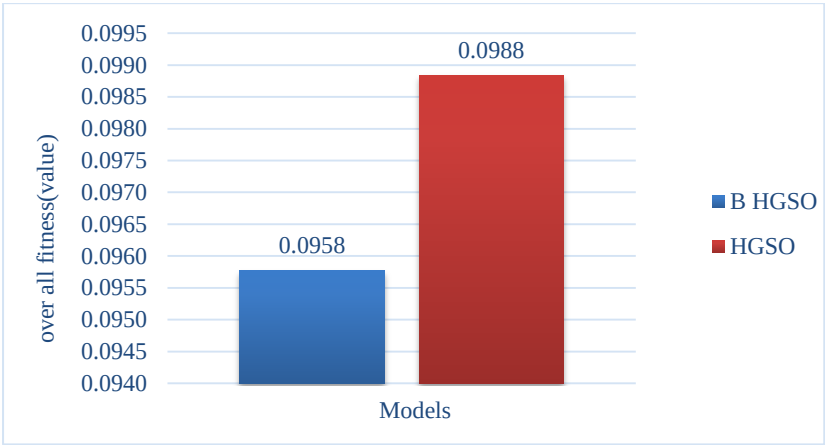

Fig 5. The overall fitness of Beta HGSO and HGSO based on KNN classifier

Table 5.*Average number of Features Selected: Comparison of βeta HGSO with Other Algorithms*

Datasets	Metric	B-HGSO	HGSO	DA	WAO	GAO	GWO	SSA
HeartEW	AVG	**5.8**	6,9333	7,4000	7,3333	6,9667	5,8667	6
	STD	**1.249**	1,5742	1,9405	2,8446	1,4499	2,0965	2,0172
BreastEW	AVG	13.05	**7,3667**	12,2333	8,9667	15,1000	10,2333	11,1333
	STD	2.0609	2,9418	2,2389	3,2641	2,4403	3,5202	**1,8333**
Exactly2	AVG	6.1	**1,1000**	7,5000	7,7000	7,0000	7,6333	7,9667
	STD	1.5133	**0,5477**	1,7956	2,8545	2,1173	2,6061	1,8843
CongressEW	AVG	4.9	**1,8667**	6,2667	5,0333	7,7000	6,8000	5,1000
	STD	2.7731	**0,5074**	2,1961	1,9205	1,6846	1,7301	1,6682
Lymphography	AVG	8.4	**6,9667**	7,5333	7,0000	8,8000	7,4667	6,7667
	STD	**1.1136**	2,0424	2,5425	3,1184	1,9547	2,8736	1,8511
IonosphereEW	AVG	12.1	**4,8667**	13,0667	7,6333	15,9333	9,6333	11,4000
	STD	2.9138	**1,5477**	2,8640	2,3116	2,6901	2,6061	2,9781
SonarEW	AVG	**28.74**	28.8	------	------	------	------	------
	STD	**4.0832**	5.9296	------	------	------	------	------
SpectEW	AVG	**8.5**	9.5	------	------	------	------	------
	STD	**1.8303**	2.8018	------	------	------	------	------
Spambase	AVG	**29.4**	32.4	------	------	------	------	------
	STD	**3.2619**	4.0447	------	------	------	------	------

Table 5 presents the average number of features selected by β-HGSO and other competitor algorithms using the k-NN classifier on various UCI datasets. By analysing the results, we conclude that the average number of selected features obtained by β-HGSO is the best for four out of nine datasets, while HGSO is the best for five datasets. Note that the best-selected features by β-HGSO achieved the best accuracy and fitness in most datasets employed in this study.

4.2 Evaluation of performance on real world dataset

To emphasize the efficiency of the proposed model, we employed a real-world dataset to identify the variables that affect individual earnings. We used our model to predict relevant data from this dataset. Specifically, we utilized an American Community Survey (ACS) dataset obtained from census.gov, spanning the period from 2015 to 2022. Additionally, we employed a random forest classifier as the machine learning model for classification. The outputs are shown below.

Table 6. *Dataset Description*

No.of features	No.of instances	No.of classes
25	90000	2

Table 7 *ACS dataset Accuracy results (RFC classifier)*

Model	Best	Mean	Worst
Beta -HGSO	0.9467	0.9417	0.9362
HGSO	0.9128	0.9022	0.8633

Table 7 presents the accuracy of the Beta-HGSO and HGSO models, evaluated using three metrics: best, mean, and worst. These metrics assess the models' classification performance on the ACS dataset. Beta-HGSO outperformed HGSO, achieving a best accuracy of 0.9467, a worst accuracy of 0.9362, and a mean accuracy of 0.9417.

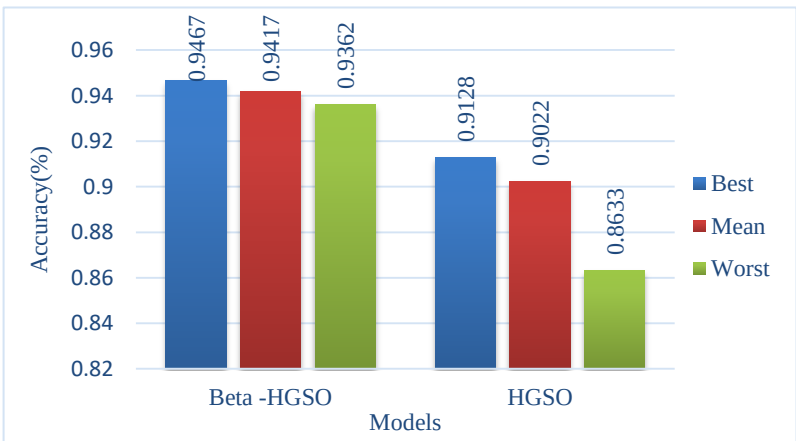

Fig.6. ACS dataset Accuracy results of Beta HGSO and HGSO based on RFC classifier

Table 8 ACS *dataset Fitness results*

Model	Best	Mean	Worst
Beta -HGSO	0.0616	0.0628	0.0643
HGSO	0.0882	0.0912	0.0946

Table 8 presents the performance of two models, Beta-HGSO and HGSO, on the ACS dataset, evaluated across three fitness metrics: Best, Mean, and Worst. These metrics indicate the fitness value achieved by each model. The Beta-HGSO model achieved the following fitness values: 0.0616 (Best), 0.0628 (Mean), and 0.0643 (Worst).

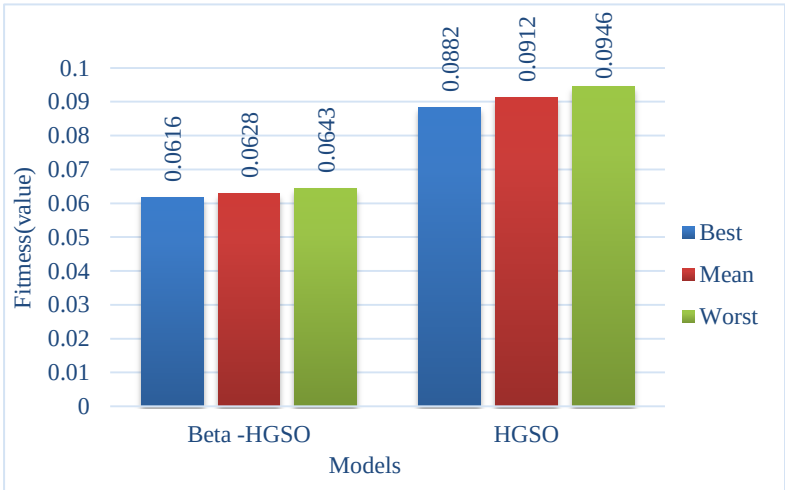

Fig.7. ACS dataset fitness results of Beta HGSO and HGSO based on RFC classifier

Table 9. *ACS dataset number of selected Features results*

Model	Best	Mean	Worst
Beta -HGSO	12	13.6	12
HGSO	10	12.3	12

Table 9 describes the number of features selected by the two models across three metrics: Best, Mean, and Worst.

5. Future work

The proposed future work involves investigating adaptive parameter tuning techniques, a commendable step toward enhancing the algorithm's robustness and efficiency. Additionally, testing the algorithm on a broader range of datasets, including more complex and diverse real-world data, will help evaluate its generalizability and effectiveness.

6. Conclusion

In this work we provide a new optimization method called Beta HGSO. In order to address the shortcomings of the traditional HGSO method, Beta HGSO addresses the problem of insufficient exploitation associated with HGSO's single location update technique by enhancing its exploitation capabilities through the use of the Beta-Hill Climbing (BHC) strategy.

Beta HGSO is a hybrid algorithm that dynamically employs BHC. Our hybrid algorithm establishes a balance between exploration and exploitation across various optimization problems, resulting in dependable and consistent performance.

A real-world dataset and nine benchmark datasets were used to emphasize the efficiency of the improved algorithm. Their results were compared to the results obtained using the original HGSO method. Through considerable experimentation, Beta HGSO was shown to have satisfactory robustness and optimization capacity, successfully discovering global optima for a range of functions.

References

[1] K. Sharifani, M. Amini," Machine Learning and Deep Learning: A Review of Methods and Applications" *World Information Technology and Engineering Journal*, Vol. 10, pp. 3897-3904, 2023.

[2] Li. J. Cheng, K. Wang, S. Morstatter, F. Trevino, R. P. Tang, J. and H. Liu, "Feature selection: A data perspective". *ACM Computing Surveys (CSUR)*, vol. 50, no.6, pp. 1-45, 2017.

[3] S. Rahim, N. Javaid, A. Ahmad, S A. Khan, Z. A. Khan, N. Alrajeh, and U. Qasim. "Exploiting heuristic algorithms to efficiently utilize energy management controllers with renewable energy sources." *Energy and Buildings*, vol. 129, pp. 452-470, 2016.

[4] A. Lambora, K. Gupta and K. Chopra, "Genetic Algorithm- A Literature Review," International *Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COMITCon)*, Faridabad, India, pp. 380-384, 2019.

[5] A. Slowik, H. Kwasnicka," Evolutionary algorithms and their applications to engineering problems" Natural computing, Vol. 32, pp. 12363–12379, 2020.

[6] M.F. Ahmad, N.A.M. Isa, W.H. Lim, and K.M. Ang," Differential evolution: A recent review based on state-of-the-art works" *Alexandria Engineering Journal*,Vol. 61, pp. 3351-4158, 2022.

[7] T.M. Shami, A. A. El-Saleh, M. Alswaitti, Q. Al-Tashi, M. A. Summakieh, and S. Mirjalili, "Particle swarm optimization: A comprehensive survey." *Ieee Access*, vol. 10, pp. 10031-10061, 2022.

[8] A. Akhtar, "Evolution of Ant Colony Optimization Algorithm--A Brief Literature Review." *arXiv preprint arXiv: 1908.08007*, 2019.

[9] A. Sharma, A. Sharma, S. Choudhary, R. K. Pachauri, A. Shrivastava, and D. Kumar. "A review on artificial bee colony and it's engineering applications." *Journal of Critical Reviews*, vol. 7, no. 11, pp. 4097-4107, 2020.

[10] T. Guilmeau, E. Chouzenoux, and V. Elvira, "Simulated annealing: A review and a new scheme" *IEEE statistical signal processing workshop (SSP)*, pp. 101-105, 2021.

[11] H. Mittal, A. Tripathi, A.C. Pandey, and R. Pal" Gravitational search algorithm: a comprehensive analysis of recent variants," *Multimedia Tools and Applications*, Vol. 80, pp. 7581–7608, 2021.

[12] N. Neggaz, E.H. Houssein, and K. Hussain, "An efficient henry gas solubility optimization for feature selection." *Expert Systems with Applications*, vol.152, p.113364, 2020.

[13] A., M. Azmi, "β-hill climbing: an exploratory local search." *Neural Computing and Applications*, vol. 28, no. 1, pp. 153-168, 2021.

[14] S. Uddin, I. Haque, H. Lu, M.A. Moni, and E. Gide, "Comparative performance analysis of K-nearest neighbour (KNN) algorithm and its different variants for disease prediction." *Scientific Reports*, vol. 12, no. 1 2022.

[15] A Parmar, R Katariya, and V Patel," A Review on Random Forest: An Ensemble Classifier" *International Conference on Intelligent Data Communication Technologies and Internet of Things (ICICI)*, pp 758–763, 2019.

[16] Y. Meraihi, A.B. Gabis, and S. Mirjalili," Grasshopper Optimization Algorithm: Theory, Variants, and Applications" *IEEE Access*, vol. 9, pp.50001-50024, 2021

[17] M. Alshinwan, L. Abualigah, and M. Shehab," Dragonfly algorithm: a comprehensive survey of its results, variants, and applications" *Multimedia Tools and Applications*, Vol. 80, pp. 14979–15016, 2021.

[18] N. Rana, M.S.A. Latiff, and S.M. Abdulhamid," Whale optimization algorithm: a systematic review of contemporary applications, modifications and developments"*Neural Computing and Applications* , Vol. 32, pp. 16245–16277, 2020.

[19] S.M. Almufti, H.B. Ahmad, and R.B. Marqas,"Grey wolf optimizer: Overview, modifications and applications." *International Research Journal of Science, Technology, Education, and Management*, Vol. 1, August 2021.

[20] L. Abualigah, M. Shehab, and M. Alshinwan," Salp swarm algorithm: a comprehensive survey" *Neural Computing and Applications*, Vol. 32, pp. 11195–11215, 2020.

المجلد (12) العدد (2) (2025)

المجلة الدولية للحاسبات والمعلومات

متاح على الإنترنت على الرابط : https://ijci.journals.ekb.eg/

طريقة التحسين الهجينة بيتا–هنرى: نهج جديد لاختيار الميزات

شادية الجزار – أ.د حاتم عبد القادر– أ.د اسماء السيد

نظم المعلومات – الحاسبات والمعلومات –المنوفيه -shadia_elgazzar@hotmail.com,

نظم المعلومات – الحاسبات والمعلومات –المنوفيه -hatem.abdelkader@ci.menofia.edu.eg,

نظم المعلومات – الحاسبات والمعلومات –المنوفيه -asmaa.elsayed@ci.menofia.edu.eg,

---

ملخص البحث:

تعد خوارزمية تحسين هنري لذوبان الغازات (HGSO) المستوحاة من الطبيعة نهجًا جديدًا لحل مشكلات التحسين الشامل من خلال محاكاة قانون هنري لذوبان الغازات. ومع ذلك، تظل مشكلة التقارب المبكر والنسبة غير المتوازنة بين الاستكشاف والاستغلال تمثل تحديات كبيرة. كما أن استراتيجية البحث البسيطة لـ HGSO تحد من قدرتها على استغلال الحلول المثلى بشكل فعال، مما يعيق أدائها في مشكلات التحسين المعقدة. لمعالجة هذه القيود، نقترح استخدام البحث المحلي Beta hill climbing لتعزيز أداء الخوارزمية. يعمل المُشغّل الجديد Beta على تحسين البحث المحلي التقليدي hill climbing من خلال تحقيق توازن دقيق بين الاستكشاف والاستغلال. من خلال دمج هذا المُشغّل، تتمكن النسخة المحسّنة من HGSO والمعروفة بـ (EHGSO) من التنقل بكفاءة أكبر في مساحة البحث وتحديد الحلول المثلى. قمنا بتقييم خوارزمية EHGSO باستخدام تسعة مجموعات بيانات معيارية وبيانات من العالم الواقعي. بالنسبة للبيانات الواقعية، استخدمنا نموذج الغابات العشوائية(Random Forest) ، بينما استخدمنا نموذج KNN مع مجموعات البيانات المعيارية. حققت EHGSO دقة بلغت 0.9467 في مجموعة البيانات الواقعية وتفوقت على الخوارزميات الأخرى مثل GOA و WOAو DAو GWOو SSA في مجموعات البيانات المعيارية. تُظهر هذه النتائج القدرات المتفوقة لـ EHGSO في معالجة مشكلات التحسين المعقدة.