Neural Computing and Applications (2021) 33:13601–13618
https://doi.org/10.1007/s00521-021-05991-y

ORIGINAL ARTICLE

Enhanced Harris hawks optimization with genetic operators for selection chemical descriptors and compounds activities

Essam H. Houssein[1] · Nabil Neggaz[2] · Mosa E. Hosney[3] · Waleed M. Mohamed[1] · M. Hassaballah[4]

Received: 23 November 2019 / Accepted: 31 March 2021 / Published online: 20 April 2021

Abstract

This paper presents modified versions of a recent swarm intelligence algorithm called Harris hawks optimization (HHO) via incorporating genetic operators (crossover and mutation CM) boosted by two strategies of (opposition-based learning and random opposition-based learning) to provide perfect balance between intensification and diversification and to explore efficiently the search space in order to jump out local optima. Three modified versions of HHO termed as HHOCM, OBLHHOCM and ROBLHHOCM enhance the exploitation ability of solutions and improve the diversity of the population. The core exploratory and exploitative processes of the modified versions are adapted for selecting the most important molecular descriptors ensuring high classification accuracy. The Wilcoxon rank sum test is conducted to assess the performance of the HHOCM and ROBLHHOCM algorithms. Two common datasets of chemical information are used in the evaluation process of HHOCM variants, namely Monoamine Oxidase and QSAR Biodegradation datasets. Experimental results revealed that the three modified algorithms provide competitive and superior performance in terms of finding optimal subset of molecular descriptors and maximizing classification accuracy compared to several well-established swarm intelligence algorithms including the original HHO, grey wolf optimizer, salp swarm algorithm, dragonfly algorithm, ant lion optimizer, grasshopper optimization algorithm and whale optimization algorithm.

Keywords Cheminformatics · Genetic operators · Harris hawks optimization · Feature selection · QSAR · Swarm intelligence

1 Introduction

Cheminformatics has many strategies that can be used in drug design and discovery. A lot of efforts with large numbers of chemical compounds are being used to evaluate specific molecular properties [1]. The prediction process of molecular properties is close to drug virtual screening of a chemical library. Commonly, the drug is an organic molecule that inhibits the function of proteins as bimolecular interactions [2]. Drug design is often referred to as a rational design and inventive process of finding new drugs based on biological target knowledge [3]. Drug design and discovery consist of lead structures optimization, quantitative structure–activity relationships (QSAR) [4] and docking of ligand into a receptor [5]. Recently, machine learning (ML) techniques are applied in chemoinformatics filed to predict the chemical descriptor selection, compound activities, molecular properties [6] and for drugs design and discovery [7]. The storage space

✉ Essam H. Houssein
essam.halim@mu.edu.eg

Nabil Neggaz
nabil.neggaz@univ-usto.dz

Mosa E. Hosney
software.eng94@yahoo.com

Waleed M. Mohamed
waleedmakram@mu.edu.eg

M. Hassaballah
m.hassaballah@svu.edu.eg

[1] Faculty of Computers and Information, Minia University, Minia 61519, Egypt

[2] Université des Sciences et de la Technologie d'Oran Mohamed Boudiaf, USTO-MB, BP 1505, EL M'naouer, 3100 Oran, Algeria

[3] Faculty of Computers and Information, Luxor University, Luxor, Egypt

[4] Department of Computer Science, Faculty of Computers and Information, South Valley University, Qena, Egypt

size increases exponentially with respect to the number of features available in the data set.

Feature selection (FS) is used in many critical fields such as classification, data mining and object recognition, where it is useful in eliminating obsolete and redundant features from datasets [8, 9]. It represents a real challenge and computational processing, especially when working with datasets of high dimensions in classification problems [10, 11]. The aim of FS is to minimize the number of features which increasing the search space and allowing ML techniques to use only the most significant features that affecting the classification accuracy [12]. Swarm intelligence (SI) algorithms are the most common methods used to solve FS-based problems [13]. The SI algorithms reflect computational intelligence methods made up of artificial agent population and inspired by social behavior of animals from the real world [14].

Heidari et al. [15] have proposed the HHO algorithm that mimics the Harris hawks cooperative hunting behavior. The original HHO maintains some important limitations, such as: (1) the exploration and exploitation are smooth but unbalanced and hence the global search and local search process becomes difficult to manipulate, (2) it has premature convergence when the problems are highly multi-modal and (3) the exploitation strategy in HHO is insufficient and the search agents may find local solutions. In this study, to overcome the limitations of HHO, a wrapper feature selection method termed HHOCM, which hybridizes HHO with crossover and mutation for chemical descriptors selection and chemical activities. The role of mutation and crossover is to generate new offspring that helps to find solutions simulating the nature laws of origin and adaptation to the environment.

Experimental results revealed that the three versions of proposed HHOCM algorithms are efficient alternatives for solving the FS problems. Several comparisons are performed considering seven well-established SI algorithms, namely the grey wolf optimizer (GWO) [16], original HHO [15], whale optimization algorithm (WOA) [17], salp swarm algorithm (SSA) [18], ant line optimizer (ALO) [19], grasshopper optimization algorithm (GOA) [20] and dragonfly algorithm (DA) [21] using k-nearest neighbors (k-NN) as a classifier. Further, the Wilcoxon test is used to evaluate the performance of the proposed algorithms. It is noted that the ROBHHOCM and HHOCM algorithms achieve the best results compared with the competitor algorithms in most statistical and graphical measurements (e.g., average and standard deviation of fitness, accuracy, number of selected features, and the convergence curves and boxplot curves). To sum up, the major contributions of this work are:

1. Crossover and mutation evolutionary operators are used to enhance the performance of the HHO algorithm.
2. Integration of opposition-based learning operator to HHO.
3. Three versions based on HHO called HHOCM, OBLHHOCM and ROBLHHOCM algorithms are proposed to select the chemical descriptors and chemical compound activities.
4. Modeling a wrapper FS paradigm using the three versions is conducted.
5. A series of experiments are carried out to prove superiority of these versions in choosing the best molecular descriptor subset with high classification accuracy.

The rest of this paper is organized as follows. Section 2 summarizes related works in the literature. Section 3 presents briefly basics of the HHO algorithm, the k-nearest neighbor (k-NN) algorithm and genetic operators. Section 4 explains the proposed modifications over the HHO algorithm. Experimental results are reported and discussed in Sect. 5. Finally, Sect. 6 concludes the paper.

2 Related work

Generally, for creating a medical molecule, it is necessary to use protein bank databases in order to determine crystal structure of the protein [22]. Computer-aided drug design (CADD) is an efficient tool for chemical compounds identification of drug design and discovery [2]. CADD methods are used to extract the compounds such as Pubchem by the pharmacophore modeling tools. These methods depend on the important concept of docking large libraries for small molecule [23]. Also, CADD methods are used to obtain protein and ligand. A good drug depends on good ligands selected by the PyMOL software [24] to separate the ligand from protein, and the energy is calculated by AutoDock software.

In the same context, QSAR is applied to describe the correlation between structures from a set of molecule and the response for targets. Thus, it can be considered as an alternative tool for the CADD methods [25]. In general, QSAR consists of an active and inactive molecule, which requires good molecular descriptors representing molecular features responsible for the relevant molecular activity. Drug design and discovery is one of the main aspects in cheminformatics including two phases: encoding phase to represent the molecular graph (or connection table) to extract vector of features through calculating the descriptors as three-dimensional information for the molecular structure and mapping phase to build various models for

ML techniques in cheminformatics. Mapping between different feature vectors and property is the major role of ML techniques in cheminformatics to discover different functions, which can be done by ML techniques [12].

Several efforts were developed for selecting proper features in datasets. Considering this fact, three categories of FS are found in the literature [26]: filter-based [27], embedded-based [28] and wrapper-based [29]. FS-based SI includes several algorithms, such as improved salp swarm algorithm using crossover [30], whale optimization algorithm [31] and binary dragonfly optimization [32]. In [33], a filter-based FS method is introduced for the QSAR Biodegradation and other medical benchmarks. It combined relief-*f* with differential evolution for selecting the most relevant features. It achieved 85.4% classification accuracy with keeping only 16 relevant molecular descriptors from 41 features. Wrapper-based methods have been attracted more attention due to the involvement of learning algorithms in the FS process. Thus, selection of significant features effected by the performance of learning algorithms (e.g., rate of correct classification accuracy) [12]. A swarm-based algorithm is introduced in [34] using wrapper FS for predicting chemical compound activities. SSA is applied for selecting the best subset of molecular descriptors of the Monoamine Oxidase (MAO) dataset. It is important to note that SSA with *k*-NN classifier obtained the highest accuracy of 87.35% and kept only 783 molecular descriptors. Houssein et al. [35] proposed two classification approaches called HHO-SVM and HHO-kNN for drug design and discovery prediction. In [36], the HHO is combined with cuckoo search for drug design and discovery in chemoinformatics.

Another branch of multi-objective optimization algorithms-based FS has developed recently for selection of molecular descriptors in QSAR by introducing the molecular descriptors subsets selection software (*MoDeSuS*) for QSAR Biodegradation [37]. Two scenarios are proposed for selecting relevant molecular descriptors known as aggregation and Pareto based FS. In the first one, a binary vector is generated which contains *m* molecular descriptors, where ones bits indicate that the molecular descriptors are selected and zeros bits indicate the molecular descriptors are ignored. In addition, the selected subset should be evaluated using aggregation function that combines the accuracy with the selection ratio. The second scenario (Pareto-based methods) employs two algorithms (i.e., non-dominated sorting genetic algorithm (NSGAII) and strength Pareto evolutionary algorithm (SPEA2)) to optimize the accuracy and the selection ratio, separately. The *MoDeSuS* achieved high performance on the QSAR Biodegradation dataset with an accuracy rate of 84% and 37% of selection ratio.

In [38], a method based on biclustering is proposed to reduce the number of molecular descriptors for predicting Biodegradation of chemical compounds. The task of Biodegradation is evaluated using three classifiers: random committee, neural network and random forest. The experimental results have shown that the best classifier was RF, which achieved 88.81% of accuracy with only 19 molecular descriptors (MD) on the QSAR Biodegradation dataset. Recently, artificial intelligence knows a remarkable progress that allows to develop several horizons based on ML and deep learning for QSAR modeling [39, 40]. Also, Putra et al. [41] combined artificial neural network and support vector machine for QSAR modeling and principal component analysis (PCA) is utilized for reducing the dimensionality of data. The performance is assessed on the QSAR Biodegradation dataset and a classification rate of 82% is achieved.

Hierarchical stochastic graphlet embedding (HSGE) [42] is introduced using different hierarchical configurations for treating molecular graph dataset. The approach achieved 95.71% of accuracy on the MAO dataset. In the same context, the work of [43] presented a fusion between old neuronal architecture (multi-layer perceptron) and recent architecture based on deep learning called CNN-MLP for predicting chemical activities. In the CNN-MLP method, two models DeepBioD+ and DeepBioD are proposed for the QSAR Biodegradation dataset based on domain-specific features engineering and learned representations from pattern samples, which achieved 90% and 87.5% of accuracy, respectively. A similar work based on a pre-trained model called ChemNet was presented in [44] for the prediction of chemical activities using deep learning model achieved 86.7% of accuracy on the QSAR Biodegradation dataset. In [45], diffusion-convolutional neural network (DCNN) was produced for graph-structured data using a representation of deep learning architecture called diffusion CNN. The experiments showed that an accuracy rate of 75.14% can be achieved on the MAO dataset.

3 Preliminaries

This section introduces necessary basics of the HHO algorithm, *k*-NN and the genetic operators.

3.1 Harris hawks optimization

The HHO [15] as a new SI algorithm is inspired from the cooperative behaviors of Harris hawks in hunting and escaping preys. Harris hawks demonstrate a variety of chasing styles dependent on the dynamic nature of circumstances and escaping patterns of a prey. In this intelligent strategy, several Harris hawks try to cooperatively attack from different directions and simultaneously converge on a detected escaping rabbit outside the cover showing different hunt strategies. The candidate solutions are the Harris hawks, and the intended prey is the best candidate solution (nearly the optimum) in each step. The three phases of the HHO algorithm can be highlighted as: exploration phase, transition from exploration to exploitation phase and exploitation phase. The hunting is modeled as:

$$x^i_{t+1} = \begin{cases} x_{rand} - \tau_1 \left| x_{rand} - 2\tau_2 x^i_t \right| & if\ \tau_5 \geq 0.5 \\ (x_{Rabbit} - \overline{x_t}) - \tau_3 \left| lb^j + \tau_4 (ub^j - lb^j) \right| & else \end{cases}$$
$$t \in [1 \cdots T], i \in [1 \cdots N], \tag{1}$$

where the current location of *ith* hawk and its new location in iteration $t+1$ are represented by x^i_t and x^i_{t+1}, whereas x_{rand} and x_{Rabbit} are randomly selected hawk location and the best solution (target:rabbit). Lower and upper bounds of *jth* dimension are defined by lb^j and ub^j, while τ_1 to τ_5 represent random numbers which belong to the interval [0, 1]. The average hawk position $\overline{x_t}$ is defined as:

$$\overline{x_t} = \frac{1}{N} \sum_{i=1}^{N} x_t(i). \tag{2}$$

In Eq. (1), the first scenario ($\tau_5 \geq 0.5$) grants a chance to the hawks to hunt randomly spread in the planned space, while the second scenario explains context when the Hawks hunt beside family members close to a target. In the exploration to exploitation transformation phase, the prey attempt to escape from the capture, so the escaping energy E_n level of the prey decreases gradually. The energy is defined by

$$E_n = 2 * E_{n0} * \left(1 - \frac{t}{T} \right), \tag{3}$$

where the initial energy (E_{n0}) is defined by $E_{n0} = 2 * rand - 1$, randomly changed inside $(-1, 1)$, and T is the maximum number of iterations. HHO keep explorative as long as $|E_n| \geq 1$ and hawks remain on exploring global regions, while it swaps into exploitative mode when $|E_n| < 1$. R refers to escaping probability of the target. The exploitation phase aims to avoid fall into local optima.

The first task-surrounding soft The surrounding soft can be formulated mathematically when $R \geq \frac{1}{2}$ and the level of energy is greater than $\frac{1}{2}$ (i.e., $|E_n| \geq \frac{1}{2}$) as:

$$\begin{aligned} & x^i_{t+1} = \Delta x^i_t - E_n \left| J x_{Rabbit} - x^i_t \right| \\ & \Delta x^i_t = x_{Rabbit} - x^i_t,\ J = 2(1 - \tau_6), \end{aligned} \tag{4}$$

where Δx^i_t is the difference between the best agent (i.e., a rabbit) and the current position of *ith* hawk. J indicates random strength jump of the prey, and τ_6 is a random number which belongs to the interval [0, 1].

The second task-surrounding hard When the level of energy is less than $\frac{1}{2}$ ($|E_n| < \frac{1}{2}$) & $R \geq \frac{1}{2}$, the rabbit becomes exhausted and the possibility of escaping low (or escaping becomes hard) because the level of energy is decreased. This behavior can be modeled by

$$x^i_{t+1} = x_{Rabbit} - E_n \left| \Delta x^i_t \right|. \tag{5}$$

The third task-surrounding soft beside advanced rapid dives This task is applicable when the level of energy is greater than $\frac{1}{2}$ ($|E_n| > \frac{1}{2}$) & $R < \frac{1}{2}$, where the rabbit still has sufficient force to run away. Hence, the hawk tries progressive dives in order to take the best position for catching the prey. This behavior is modeled by integrating the Lévy flight function [46].

The position of *ith* hawk should be modified using:

$$\begin{aligned} & x^i_{t+1} = \begin{cases} y & if\ fit(y) < fit\left(x^i_t\right) \\ z & if\ fit(z) < fit\left(x^i_t\right), \end{cases} \\ & y = x_{rabbit} - E_n \left| J x_{rabbit} - x^i_t \right|, \\ & \quad z = y + r_v \times Lv(D), \end{aligned} \tag{6}$$

where

$$Lv(D) = 0.01 \times \frac{rand(1, D) \times \sigma}{|rand(1, D)|^{\frac{1}{\beta}}}, \tag{7}$$

$$\sigma = \left(\frac{\Gamma(1 + \beta) \times \sin\left(\frac{\pi\beta}{2}\right)}{\Gamma\left(\frac{1+\beta}{2}\right) \times \beta \times 2^{\left(\frac{\beta - 1}{2}\right)}} \right)^{\frac{1}{\beta}}, \tag{8}$$

where D is the dimensionality space, r_v contains D components generated randomly inside (0,1), Lv represents the Lévy flight function, β is a constant with default $\beta = 1.5$ and *fit* indicates the fitness function computed by Eq. (16).

The fourth task-surrounding hard beside advanced rapid dives In this task, it is assumed that $R < \frac{1}{2}$ and the level of energy is less than $\frac{1}{2}$ ($|E_n| < \frac{1}{2}$), the prey has a lower level of energy to evade, and Hawks are close to realize a successive dives for catching. This process can be described by

$$\begin{aligned} & x^i_{t+1} = \begin{cases} y & if\ fit(y) < fit\left(x^i_t\right) \\ z & if\ fit(z) < fit\left(x^i_t\right) \end{cases}; \\ & y = x_{Rabbit} - E_n \left| J x_{Rabbit} - \overline{x} \right|; \\ & z = y + r_v \times Lv(D). \end{aligned} \tag{9}$$

For illustration, the general flowchart of the HHO algorithm is shown in Fig. 1.

3.2 *k*-Nearest neighbors algorithm (*k*-NN)

The *k*-NN classifier belongs to the supervised machine for identifying a new pattern based on statistical metric. It is considered as a lazy model of learning, which can be performed for prediction tasks and classification problems [47]. This algorithm presents a certain advantage that is an easy interpreter of the output. It provides less computing cost and efficiency. The classification process depends only on computing the Euclidean distance between the current test example and the query examples of training data. After that, the first k minimal distances are selected to determine the label of the current test vector. The *k*-NN is composed of different steps given in Algorithm 1.

Algorithm 1 The *k*-NN algorithm

```
Inputs: Import the training and test datasets.
Initialize the number of neighbors $k$.
for (each example in the test dataset) do
    Evaluate the metric between the current (test sample) and the training samples.
    Save the calculated distances and sort it in a list.
    Select the first $k$ samples.
    Determine the output label of the current example.
end for
Return Accuracy ($Acc$) and classification error ($Er = 1 - Acc$) .
```

3.3 Genetic operators

The use of evolutionary operators is widely exploited in several algorithms. Primarily, two basic algorithms have explored deeply genetic operators such as differential evolution and genetic algorithms. Here, we give a quick overview of the genetic operators (i.e., mutation, crossover and selection).

Mutation The results of the tasks numbers three and four of HHO and the target solution (x_{Rabbit}) are utilized for producing the mutation operation. For each component, a number between 0 and 1 is randomly generated. In a case the value is superior to the mutation rate (ζ), the element of the target agent (x_{Rabbit}) is considered. If this value is less than to the mutation rate (ζ), the old vector is replaced by the component of y or z vectors. The mutation operator is determined using:

$$\begin{aligned} y_{Mut} &= \begin{cases} x_{Rabbit} & if\ \ rand_1 \geq \zeta \\ y & else \end{cases} \quad and \\ z_{Mut} &= \begin{cases} x_{Rabbit} & if\ \ rand_2 \geq \zeta \\ z & else \end{cases} \\ Where:& \begin{cases} \zeta = \dfrac{t}{T}; \\ y = x_{rabbit} - E_n \left| J x_{rabbit} - x_t^i \right|; \\ z = y + r_v \times Lv(D) \end{cases} \end{aligned} \tag{10}$$

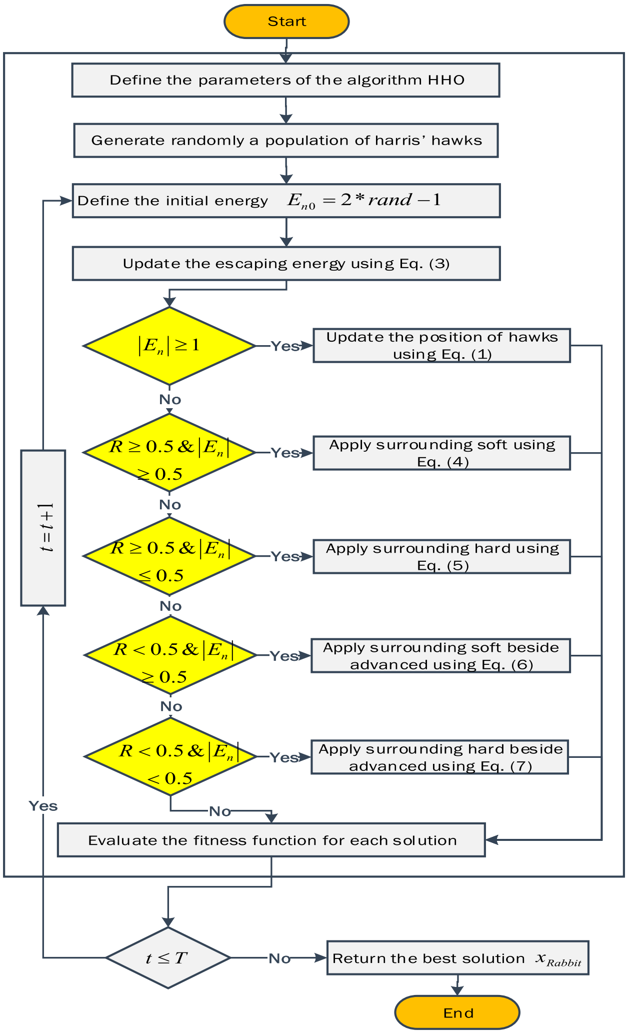

Start
Define the parameters of the algorithm HHO
Generate randomly a population of harris' hawks
Define the initial energy $E_{n0} = 2 * rand - 1$
Update the escaping energy using Eq. (3)
$|E_n| \geq 1$
Yes
Update the position of hawks using Eq. (1)
No
$R \geq 0.5 \& |E_n| \geq 0.5$
Yes
Apply surrounding soft using Eq. (4)
No
$R \geq 0.5 \& |E_n| \leq 0.5$
Yes
Apply surrounding hard using Eq. (5)
No
$R < 0.5 \& |E_n| \geq 0.5$
Yes
Apply surrounding soft beside advanced using Eq. (6)
No
$R < 0.5 \& |E_n| < 0.5$
Yes
Apply surrounding hard beside advanced using Eq. (7)
No
$t = t + 1$
Yes
Evaluate the fitness function for each solution
$t \leq T$
No
Return the best solution x_{Rabbit}
End

Fig. 1 Flowchart of the HHO algorithm

$$\begin{aligned} y_{Mut} &= \begin{cases} x_{Rabbit} & if\ \ \rho_1 \geq \zeta \\ y & else \end{cases} \quad and \\ z_{Mut} &= \begin{cases} x_{Rabbit} & if\ \ \rho_2 \geq \zeta \\ z & else \end{cases} \\ Where:& \begin{cases} \zeta = \dfrac{t}{T}; \\ y = x_{rabbit} - E_n \left| J x_{rabbit} - \overline{x} \right|; \\ z = y + r_v \times Lv(D). \end{cases} \end{aligned} \tag{11}$$

Crossover In order to produce more diversity, the crossover involves recombination of two individuals. An intermediate crossover with a random number τ is used to generate a new offspring w_{Cross} as

$$w_{Cross} = y_{Mut} + (\tau) * (z_{Mut} - y_{Mut}). \tag{12}$$

This type of operator allows children to inherit more information from parents compared to other type as linear recombination.

Selection The type of selection used in HHO is a greedy selection inspired from differential evolution. The offspring produced after evolutionary functions (mutation & crossover) are accessed. Then, the performance of the child and parent is compared to select the best one. Finally, the parent has a chance to remain in the population if their performance is high. The greedy selection is defined by the following rule:

$$x_{t+1}^{i} = \begin{cases} y_{Mut} & if\ fit(y_{Mut}) < fit\left(x_t^i\right) \\ z_{Mut} & if\ fit(z_{Mut}) < fit\left(x_t^i\right) \\ w_{Cross} & if\ fit(w_{Cross}) < fit\left(x_t^i\right) \end{cases} . \tag{13}$$

4 The proposed HHOCM algorithm

In this section, we give an alternative method for FS that combines the HHO with genetic operators. Like other meta-heuristic algorithms, HHO tends to be trapped in low diversity, local optima and unbalanced exploitation ability [48, 49]. Although HHO has the characteristics of acceptable convergence speed and a simple structure, it may fail to maintain the balance between exploration and exploitation and fall into a local optimum in some complex optimization problems [50].

Thus, the main contribution of the proposed HHOCM algorithm focuses on integrating genetic operators (mutation, crossover and selection) for solving the problem of exploitation in the HHO algorithm. To this end, the proposed HHOCM tries to ensure more diversity considering two main phases: initialization phase and updating phase. The framework of the proposed HHOCM algorithm for FS is given in Fig. 2.

4.1 Initialization phase

In this step, HHOCM generates N swarm agents in the first population, where each individual represents a portion of molecular descriptors (features) to be selected for evaluation. This step has a significant effect on the convergence and aptitude of the optimal solution. The population X is generated randomly as:

$$x_i^j = lb^j + \lambda^j \times (ub^j - lb^j),\ \ i = 1, 2, ., N; j = 1, 2..D. \tag{14}$$

The lower and upper bounds lb^j and ub^j for each candidate solution i are in the range of [0, 1]. The λ^j is a random number $\in [0, 1]$. To select a subset of molecular descriptors, an intermediate binary conversion step is necessary before fitness evaluation. Thus, each solution x^i undergoes a binary conversion (x_{bin}^i) using:

$$x_{bin}^i = \begin{cases} 1 & \text{if } x^i > 0.5 \\ 0 & \text{otherwise.} \end{cases} \tag{15}$$

The solution x^i with ten molecular descriptors, where $x^i = [0.6, 0.2, 0.9, 0.33, 0.15, 0.8, 0.2, 0.75, 0.1, 0.9]$, is considered. The operation of conversion is applied using Eq. (15) to generate a binary vector x_{bin}^i, where ones imply that the molecular descriptors are selected and otherwise are not selected. This means that first, third, sixth, eighth and the last molecular descriptors in original datasets are relevant ones and must be selected, while the others are irrelevant features and must be eliminated. After determining the subset of selected molecular descriptors, the fitness function is calculated for each agent x_{bin}^i to determine the quality of these features. The fitness of the *ith* solution is defined by

$$fit_i = \upsilon_1 \times Er_i + \upsilon_2 \times \frac{d_i}{D}, \tag{16}$$

where $\upsilon_1 = 0.99$ and $(\upsilon_2 = 1 - \upsilon_1)$. The weight υ_1 represents the equalizer parameter employed to ensure a relationship between the error rate of classification $(Er_i = 1 - Acc_i)$ and the size of selected molecular descriptors (d_i). In Eq. (16), D is the total size of Molecular Descriptors (MD) in the original dataset. The k-NN is utilized as a classifier in the FS cycle. As a strategy of classification, the hold-out is utilized, which assigns 80% as a training set and the rest of data as testing samples. The Er_i refers to the error rate of test datasets computed by k-NN (Algorithm 1). The lower value of fitness through all agents is assigned to the best prey (x_{Rabbit}).

4.2 Updating phase

The process of updating solutions consists of applying firstly the exploration step, which aims to apply a global search when the energy is greater than one. After that, the transition from exploration to exploitation is applied. Then, the exploitation phase is employed, which contains four tasks: surrounding soft, surrounding hard, surrounding soft beside advanced rapid dives and surrounding hard beside advanced rapid dives boosted by genetic operators. For improving the local search capability, HHOCM integrates the mutation operator in task three (surrounding soft beside advanced rapid dives) and task four (surrounding hard beside advanced rapid dives) of HHO using Eqs. (10) and (11), respectively. For more diversity, another genetic operator is introduced called crossover. This operator tries to combine both mutant vectors y_{Mut} and z_{Mut} for producing a new child w_{Cross} as described in Eq. (12). The fitness values of all offspring: y_{Mut}, z_{Mut} and w_{Cross} based on selection operator Eq. (13) are compared to identify the best prey x_{Rabbit}. The process is reproduced, while the

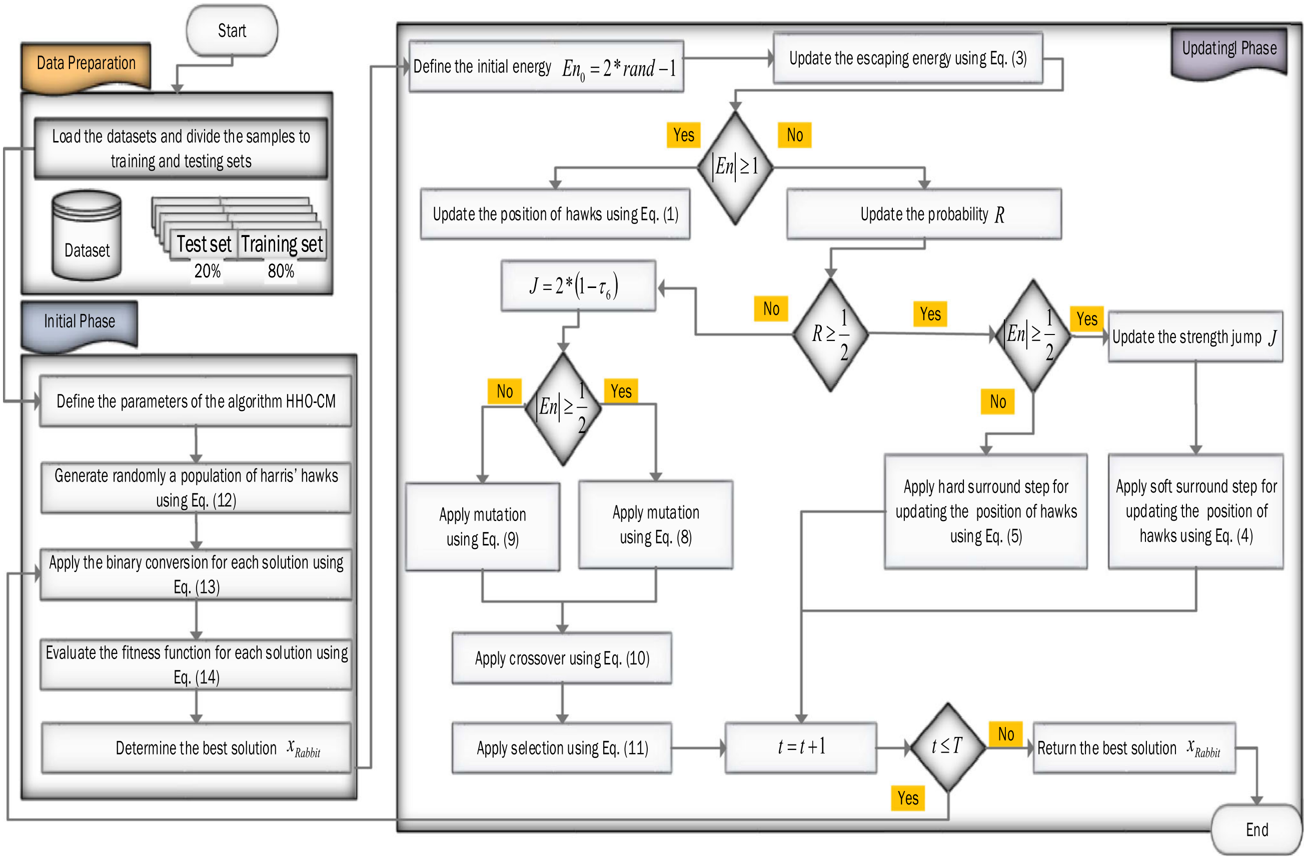

Start
Data Preparation
Load the datasets and divide the samples to training and testing sets
Dataset
Test set
Training set
20%
80%
Initial Phase
Define the parameters of the algorithm HHO-CM
Generate randomly a population of harris' hawks using Eq. (12)
Apply the binary conversion for each solution using Eq. (13)
Evaluate the fitness function for each solution using Eq. (14)
Determine the best solution x_{Rabbit}
Updating Phase
Define the initial energy $En_0 = 2*rand - 1$
Update the escaping energy using Eq. (3)
$|En| \geq 1$
Yes
No
Update the position of hawks using Eq. (1)
Update the probability R
$J = 2*(1-\tau_6)$
No
$R \geq \frac{1}{2}$
Yes
$|En| \geq \frac{1}{2}$
Yes
Update the strength jump J
No
No
$|En| \geq \frac{1}{2}$
Yes
Apply mutation using Eq. (9)
Apply mutation using Eq. (8)
Apply hard surround step for updating the position of hawks using Eq. (5)
Apply soft surround step for updating the position of hawks using Eq. (4)
Apply crossover using Eq. (10)
Apply selection using Eq. (11)
$t = t+1$
$t \leq T$
No
Yes
Return the best solution x_{Rabbit}
End

Fig. 2 A general framework for the proposed HHOCM algorithm

termination condition is met. The stop criterion corresponds to the maximum amount of iterations that allows to evaluate the performance of the HHOCM algorithm. Then, the best solution x_{Rabbit} is returned and converted to determine the number of relevant features. In this regard, experiments are carried out 30 times independently for achieving accurate and precise results.

5 Experimental results and discussion

For validating the effectiveness of the proposed HHOCM algorithm, a number of experiments are conducted using two common datasets widely used in the field of chemoinformatics: QSAR Biodegradation and MAO. The first experiment is assigned to study the impact of swarm size (N) and the maximum number of iterations (T) on the accuracy and number of selected features. The second experiment aims to determine the best value of control parameter β used in Lévy flight function. The third experiment is conducted to compare the performance of the HHOCM algorithm and seven recent SI algorithms (i.e., HHO, GWO, WOA, SSA, DA, GOA and ALO) based on the optimal control parameters obtained by previous two experiments between. Each algorithm is executed 30 times with keeping the same optimal values of N, T and β. Other experiments are conducted using statistical Wilcoxon test, which is used as assessment measure in order to verify significance of the accuracy achieved by HHOCM and ROBLHHOCM against the other competitor algorithms. The last experiment is conducted to compare between the three versions, HHOCM, OBLHHOCM and ROBLHHOCM, and other works from the literature using same parameters configuration on both of the QSAR Biodegradation and MAO datasets. The running is established on a PC with Intel Core i7-5500 CPU@2.40 GHz 2.40 GHz, 8 GB RAM, Windows 10 and Matlab 2016a.

5.1 Parameter settings

Parameters settings of the DA, ALO, GWO, WOA, SSA, GOA and HHO algorithms as well as the proposed versions of HHO are listed in Table 1.

5.2 Performance measures

The performance of the proposed

HHOCM algorithm is assessed based on several criteria including, average and standard deviations of fitness, accuracy, number of selected features, sensitivity, specificity and CPU time. Table 2 shows the confusion matrix that allows to produce some performance metrics as accuracy (Acc), sensitivity (Sn) and specificity (Sp).

- Average accuracy (AVG_{Acc}): (Acc) represents the correct number of correspondences between the label of sample data and the output of classifier and is computed using:

$$Acc = \frac{Tp + Tn}{Tp + Fn + Fp + Tn}. \tag{17}$$

The number of runs is fixed $N_r = 30$, so the average accuracy AVG_{Acc} is calculated as:

$$AVG_{Acc} = \frac{1}{N_r} \sum_{k=1}^{N_r} Acc_{best}^{(k)}. \tag{18}$$

- Average sensitivity (AVG_{Sn}): The sensitivity (Sn) accesses the rate of prognosticating positive samples as:

$$Sn = \frac{Tp}{Tp + Fn}. \tag{19}$$

The AVG_{Sn} is calculated from the best prey (x_{Rabbit}) using:

$$AVG_{Sn} = \frac{1}{N_r} \sum_{k=1}^{N_r} Sn_{best}^{(k)}. \tag{20}$$

Table 1 Parameters settings of the SI algorithms

Algorithms	Parameters setting
DA	Dragonfly number $N = 10$
	$T = 100$
	D indicates to the number of features
ALO	Ants number $N = 10$
	$T = 100$
	D indicates the dimensional of features
GWO	a ∈ [2; 0]
	Wolves number ($N = 10$)
	$T = 100$
	D indicates to features number
WOA	a ∈ [2; 0]
	b = 1
	Whales number$N = 10$
	$T = 100$
	D indicates to the size of features
SSA	c_1 and c_2 are randomly distributed
	Salps number $N = 10$
	$T = 100$
	D indicates to features domain
GOA	$C_m ax = 1$ and$C_m in = 0.00004$
	number of agents ($N = 10$)
	$T = 100$
	D indicates to features number
HHO	$N = 10$
	$T = 100$
	D indicates to features number
	$\beta = 1.5$ used in Lévy flight function [46]
HHOCM	The size of swarm ($N = 10$, $N = 20$ and $N = 30$)
OBLHHOCM	Maximum number of iterations ($T = 50$,$T = 100$ and $T = 150$)
ROBLHHOCM	$\beta \in [0.5; 2]$ used in Lévy flight function
	The dimensionality (D) represents to the number of features
	$\upsilon_1 = 0.99$ and $\upsilon_2 = 0.01$ in fitness function Eq. (16)

Table 2 Confusion matrix

Actual	Predicted 1	Predicted 0
1	Tp	Fn
0	Fp	Tn

- Average precision (AVG_{Pr}): The precision (Pr) indicates the rate of true predicted samples as:

$$Pr = \frac{Tp}{Fp + Tp}. \tag{21}$$

The AVG_{Pr} is determined via the following equation:

$$AVG_{Pr} = \frac{1}{N_r}\sum_{k=1}^{N_r} Pr_{best}^{(k)}. \tag{22}$$

- Average fitness value (AVG_{fit}): The objective value estimates the quality of algorithms that study the correlation between the selection ratio of FS and the error rate of classifier as in Eq. (16). Its average is computed by

$$AVG_{fit} = \frac{1}{N_r}\sum_{k=1}^{N_r} fit_{best}^{(k)}. \tag{23}$$

- Average size of selected features (AVG_{size}): This metric implies the size of relevant features. It is computed as:

$$AVG_{size} = \frac{1}{N_r}\sum_{k=1}^{N_r} d_{best}^{(k)}, \tag{24}$$

where $d_{best}^{(k)}$ is the cardinally of the selected features of the best agent for kth execution.

- Average CPU time (AVG_{Time}): It is the average of computation time for each method, that is:

$$AVG_{Time} = \frac{1}{N_r}\sum_{k=1}^{N_r} T_{best}^{(k)}. \tag{25}$$

- Standard deviation (Std): It is the quality of each algorithm and analysis of the obtained results over different executions and metrics. It is calculated for all measures described previously.

5.3 Datasets and preprocessing details

Monoamine Oxidase (MAO) dataset is represented by an enzyme that is dispersed in largest tissues. It can catalyze the inactivation and oxidation of monoamine neurotransmitters. The information used in this data is taken from the publicly available GREYC's chemistry dataset.[1] Thus, it is transferred from MOA to simplified molecular-input line entry system (SMILES) styles via open Babel software [51]. Then, the molecular descriptors are determined using E-Dragon [4]. It contains 1665 features (MD) with 68 compounds divided into two classes.

QSAR Biodegradation dataset has 41 attributes (molecular descriptors) for classifying 1055 chemicals compounds. This data is explored in the field of discrimination between two chemical classes including 356 of readily biodegradation samples and 699 of not readily biodegradation patterns. In addition, this data can be useful for QSAR development in order to determine the correlation molecular biodegradation and chemical design. It is available on the web page of UCI.[2]

The preprocessing stage of the datasets follows three steps as follows:

1. Information of proteins is converted to isomeric SMILES using the open Babel software [51]. Information of proteins is stored in MOA chemical format that must be converted to isomeric SMILES using Babel software. Features represent attributes that have values for making instances.
2. Descriptors are calculated by E-Dragon; in chemistry, the features are performed for implementation of different 2*D* and 3*D* data in QSAR model and calculate descriptor by E-Dragon software [4]. The descriptors are categorized into types as structural or physicochemical (weight and volume of molecule, rotary links, the distance inter atoms, the type of atom, account of molecular walking, electro-negativity, atom distribution, aromatic and thawed characteristics).
3. The correlation between chemical design and biological activeness is expressed mathematically using QSAR. Also, the features can identify the instances. QSAR is used for seeking main characteristics of chemical compounds as shown in Fig. 3. On the other side, several techniques of ML are exploited to structure–activity correlation analysis for predicting the similarity of the compounds in the presence of a given malady. The compounds of complex molecule contain several features like topological factors [52].

5.4 Sensitivity analysis

This initial test is conducted to determine sensitivity and to understand the main impact of some HHOCM parameters, such as the size of swarm (N), number of iterations (T) and the β parameter. The sensitivity is assessed in three stages. First, we treat the effect of swarm size and maximum number of iterations according to accuracy (AVG_{Acc}) and

[1] https://brunl01.users.greyc.fr/CHEMISTRY/.

[2] https://archive.ics.uci.edu/ml/datasets/QSAR+biodeg.

number of selected features (AVG_{size}) obtained by HHO and HHOCM. Second, we study the influence of the β parameter used in Lévy flight function according to accuracy (AVG_{Acc}) and number of selected features (AVG_{size}) obtained using HHO and HHOCM on the QSAR Biodegradation and MAO datasets. Third, we analyze the influence of initialization using opposition-based learning (OBL) and random OBL (ROBL) according to accuracy (AVG_{Acc}) and number of selected features (AVG_{size}) obtained by HHO and HHOCM on the two datasets (QSAR Biodegradation and MAO datasets). A short description for OBL [53, 54] and ROBL [55] is described as follows:

- OBL can produce opposite solutions that enhance the convergence and jump out local optima. So, this operator can be modeled mathematically using

$$x_i^{j*} = lb^j + ub^j - x_i^j, \; i = 1, 2, \ldots, N; j = 1, 2 \ldots, D. \tag{26}$$

- ROBL as a new operator allows to explore the search space with more diversity. It can be formulated by

$$x_i^{j*} = lb^j + ub^j - r * x_i^j. \tag{27}$$

  r is a random number $\in [0, 1]$. According to the best fitness of current solution (x_i) and their opposite (x_i^*), the initial population is created.

By inspecting the results of Tables 3, 4, 5 and 6, it can be seen that the optimal values of accuracy (AVG_{Acc}) and the number of selected features (AVG_{size}) are obtained when the swarm size N is 10 and the maximum number of iterations T is 100 for both datasets using the basic HHO and HHOCM. The second stage is conducted to treat the impact of β by varying their value from 0.5 to 2 and fixing the swarm size (N) and the maximum number of iterations (T) to the best values obtained in the first stage, which are 10 and 100, respectively. From Tables 7, 8, 9 and 10, it can be observed that the best values of accuracy and selected features are reached when the value of β in Lévy flight function is equal to 1.5 for both datasets using HHO and HHOCM. From Tables 11 and 12, we can highlight the impact of OBL and ROBL in initialization step for basic HHO and the proposed method HHOCM over both datasets (QSAR Biodegradation and MAO datasets). It can be seen clearly that ROBL enhanced the performance of HHO and HHOCM for both datasets. Additionally, ROBLHHOCM provides high performance in terms of average accuracy and size of selected features compared to HHO, OBLHHO, ROBLHHO, HHOCM and OBLHHOCM. The best obtained values of these control parameters are used for the rest of experiments. The initialization-based OBL allows to give another angle of view that help exploring a new variants of HHOCM called OBLHHOCM and OBLHHOCM.

5.5 Comparison of HHOCM with other SI algorithms

- In terms of the average and standard deviations of fitness: Table 13 reports the mean fitness values obtained by the proposed algorithms HHOCM, OBLHHOCM and ROBLHHOCM and recent SI algorithms. It can be deduced clearly that ROBLHHOCM outperforms all other competitor algorithms on both datasets. The performance can be interpreted by two reasons: The first reason is justified by the use of genetic operators in HHO, which are based on evolutionary CM operators, while the second one is illustrated by the use of OBL operator, specially Random OBL that enhances the exploration and avoids the convergence to local optima. Also, the OBLHHO algorithm takes the second rank in terms of average fitness for the QSAR Biodegradation dataset. However, HHOCM is ranked thirdly which can be interpreted by generating more diverse solutions by the use of CM operators. For the MAO dataset, the convergence curves between the three variants of HHOCM are shown in Fig. 4. The proposed ROBLHHOCM algorithm highlights more stability for both datasets because the value of Std is close to zero, which presents the key reason of good balance between exploration and exploitation.

- In terms of the average and standard deviations of accuracy and selected features: the performance of three variants of HHOCM (ROBLHHOCM, OBLHHOCM and HHOCM), three variants of HHO (ROBLHHO, OBLHHO and HHO) and other swarm competitor algorithms in terms of accuracy and number of selected features are illustrated in Tables 14 and 15. It can be seen that ROBLHHOCM finds the most informative features that provide high accuracy for both datasets. It is important to highlight that ROBLHHOCM achieves high classification accuracy of 100% with keeping only four features from 1665 in the case of MAO dataset that represents high-dimensional low-instance data. Also, it can be observed that the three variants of HHOCM outperform the variants of HHO in terms of average correct classification rate and average size of selected features for both datasets. For MAO dataset, the second rank is shared between OBLHHOCM and HHOCM in terms of average accuracy, while the second best optimizer for the QSAR Biodegradation dataset is OBLHHOCM. In this regard, the three variants of HHOCM achieve high

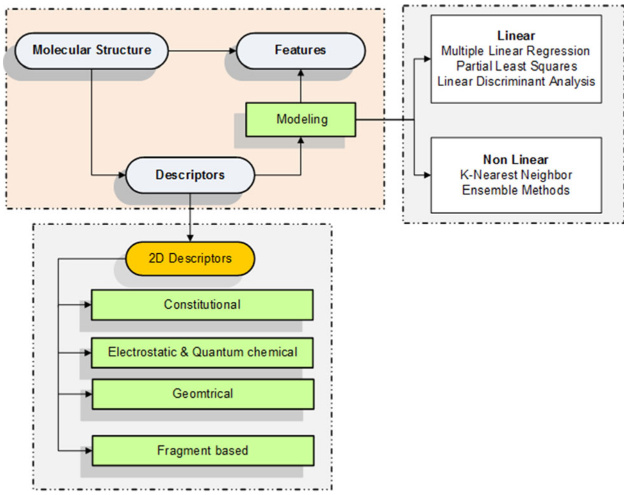


Fig. 3 Flowchart of the QSAR model

Table 3 Impact of iterations number and swarm size on the accuracy and number of selected features for the MAO dataset using basic HHO

N	*T*	The MAO dataset	
		AVG_{Acc}	AVG_{Size}
HHO			
10	50	0.8786	92.4000
20	50	0.8833	82.6333
30	50	0.9333	37.3333
10	100	**0.9476**	**15.6333**
20	100	0.9286	21.6333
30	100	0.9190	26.9333
10	150	0.9381	18.9667
20	150	0.9381	25.7000
30	150	0.9333	20.3000

Table 4 Impact of iterations number and swarm size on the accuracy and number of selected features for the MAO dataset using HHOCM

N	*T*	The MAO dataset	
		AVG_{Acc}	AVG_{Size}
HHOCM			
10	50	0.9714	34.6667
20	50	0.9952	54.7333
30	50	0.9976	9.0667
10	100	**1.0000**	**4.5667**
20	100	**1.0000**	4.8667
30	100	**1.0000**	5.4333
10	150	**1.0000**	5.8333
20	150	**1.0000**	5.6000
30	150	**1.0000**	5.6667

performance in terms of average accuracy and average size of selected features.

– In terms of the average and standard deviations of sensitivity and precision metrics: Comparison the

Table 5 Impact of iterations number and swarm size on the accuracy and number of selected features for QSAR Biodegradation dataset using basic HHO

N	*T*	QSAR Biodegradation dataset	
		AVG_{Acc}	AVG_{size}
HHO			
10	50	0.8627	20.7667
20	50	0.8373	21.3333
30	50	0.8578	20.7667
10	100	**0.8798**	**19.9667**
20	100	0.8537	20.4667
30	100	0.8669	20.3667
10	150	0.8667	20.8333
20	150	0.8491	21.9333
30	150	0.8403	21.6667

Table 6 Impact of iterations number and swarm size on the accuracy and number of selected features for the QSAR Biodegradation dataset using HHOCM

N	*T*	QSAR Biodegradation dataset	
		AVG_{Acc}	AVG_{size}
HHOCM			
10	50	0.8825	16.0000
20	50	0.8795	16.1667
30	50	0.8844	15.4667
10	100	**0.9079**	**14.8667**
20	100	0.8863	14.9667
30	100	0.8946	14.8967
10	150	0.8978	14.9333
20	150	0.8803	15.6333
30	150	0.9013	15.3333

Table 7 Impact of the β parameter on Lévy function for the MAO dataset using basic HHO

β	The MAO dataset	
	AVG_{Acc}	AVG_{Size}
HHO		
0.5	0.9305	37.3000
1	0.9405	31.5000
1.5	**0.9476**	**15.6333**
2	0.8643	22.8667

performance of sensitivity and precision of three variants of HHOCM, three variants of HHO and six SI algorithms are illustrated in Tables 16 and 17. The performance of ROBLHHOCM in terms of sensitivity

Table 8 Impact of the β parameter on Lévy function for the MAO dataset using HHOCM

β	The MAO dataset	
	AVG_{Acc}	AVG_{Size}
HHOCM		
0.5	0.9667	26.2333
1	**1.0000**	18.0000
1.5	**1.0000**	**4.5667**
2	**1.0000**	18.7

Table 9 Impact of the β parameter on Lévy function for the QSAR Biodegradation dataset using basic HHO

β	QSAR Biodegradation dataset	
	AVG_{Acc}	AVG_{Size}
HHO		
0.5	0.8674	21.7667
1	0.8642	20.5000
1.5	**0.8798**	**19.9667**
2	0.8469	21.8333

Table 10 Impact of the β parameter on Lévy function for the QSAR Biodegradation dataset using basic HHOCM

β	QSAR Biodegradation dataset	
	AVG_{Acc}	AVG_{Size}
HHOCM		
0.5	0.8705	14.7000
1	0.8912	14.9000
1.5	**0.9079**	**14.8667**
2	0.8954	14.8733

Table 11 Impact of initialization strategies for the MAO dataset using basic HHO and HHOCM

Algorithms	The MAO dataset	
	AVG_{Acc}	AVG_{Size}
HHO	0.9476	15.6333
OBLHHO	0.9738	10.1333
ROBLHHO	0.9786	9.9333
HHOCM	**1.0000**	4.5667
OBLHHOCM	**1.0000**	4.4667
ROBLHHOCM	**1.0000**	**4.3333**

and precision is still much better than all other competitor algorithms. In terms of precision, we can observe a clear advantage obtained by the three variants of HHOCM, specially for MAO dataset.

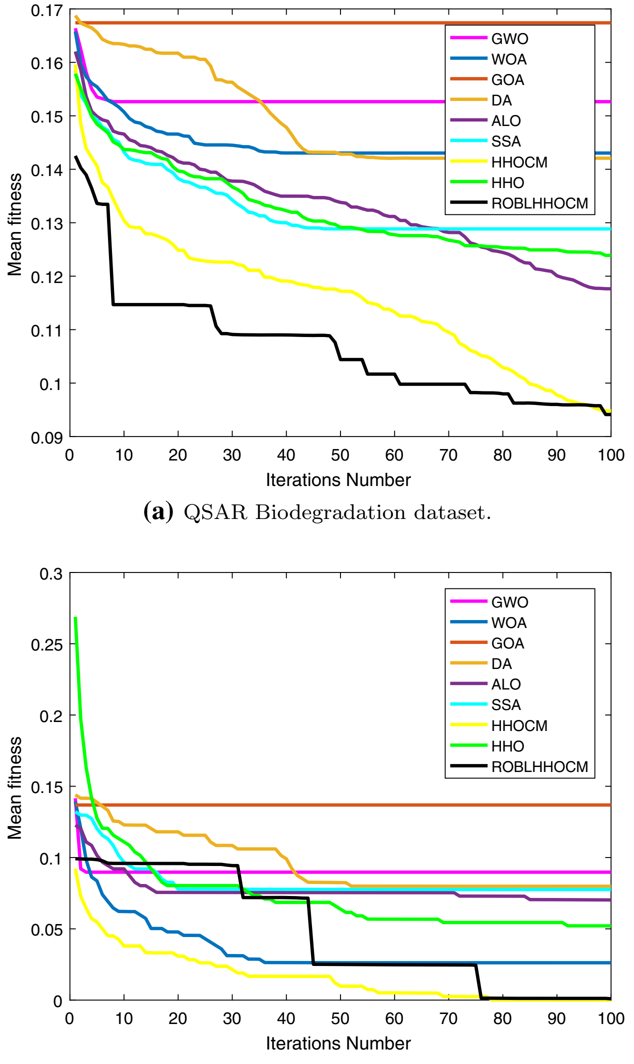
Table 12 Impact of initialization strategies for the Biodegradation dataset using basic HHO and HHOCM

Algorithms	QSAR Biodegradation dataset	
	AVG_{Acc}	AVG_{Size}
HHO	0.8798	19.9667
OBLHHO	0.8826	18.8667
ROBLHHO	0.9039	18.6667
HHOCM	0.9079	14.8667
OBLHHOCM	0.9081	14.0000
ROBLHHOCM	**0.9084**	**13.6667**

Table 13 The average fitness values of all competing optimizers

Algorithms	QSAR Biodegradation dataset		The MAO dataset	
	AVG_{fit}	Std_{fit}	AVG_{fit}	Std_{fit}
DA	0.1421	0.0089	0.0799	0.0391
WOA	0.1430	0.0078	0.0262	0.0346
GOA	0.1674	0.0090	0.1369	0.0244
ALO	0.1176	0.0059	0.0703	0.0179
GWO	0.1526	0.0100	0.0897	0.0354
SSA	0.1288	0.0104	0.0776	0.0129
HHO	0.1239	0.0081	0.0521	0.0366
OBLHHO	0.1208	0.0088	0.0260	0.0346
ROBLHHO	0.0997	0.0071	0.0212	0.0329
HHOCM	0.0948	0.0081	0.0000	**0.0001**
OBLHHOCM	0.0944	0.0065	0.0000	**0.0001**
ROBLHHOCM	**0.0941**	**0.0010**	**0.0000**	**0.0001**

- In terms of the average and standard deviations of CPU time: The CPU time consumed by the three variants of HHOCM/HHO and the other algorithms is given in Table 18. From the listed results, it can be observed that the WOA is very fast, specially for the MAO dataset when the number of patterns is small, while the three variants of HHOCM require more time when the number of samples increases exponentially. This behavior can be interpreted by adding two genetic operators: CM and the use of OBL operator. For the QSAR dataset, SSA provides the lowest time due to the use of simple updating operator.
- Wilcoxon rank-sum test: The significance of the obtained results using different algorithms requires to realize a statistical test in order to access the efficiency of the proposed ROBLHHOCM algorithm against HHOCM and other SI algorithms including DA, WOA, GOA, ALO, GWO, SSA and HHO. Table 19 shows the p-values of Wilcoxon rank-sum test based on



(a) QSAR Biodegradation dataset.

(b) The MAO dataset.

Fig. 4 Convergence curves of the HHOCM and ROBLHHOCM algorithms against other SI algorithms

accuracy metric. It can be concluded that the proposed ROBHHOCM provides a clear superiority compared to the other SI algorithms. For both datasets, the ROBLHHOCM obtained lower values of p-values, which are less than 1% compared to all SI except HHOCM. Thus, the proposed algorithms HHOCM and ROBLHHOCM are statistically significant compared to all optimizer tested in this study. In addition, we can see that the HHOCM and ROBLHHOCM provide same performance on QSAR, while on the MAO dataset, HHOCM is statistically significant compared to ROBLHHOCM.

- Graphical analysis: Fig. 4 illustrates convergence curves of the ROBLHHOCM, HHOCM algorithms against all other SI algorithms HHO, GWO, GOA, WOA, SSA, DA and ALO, which are implemented and assessed under same conditions (i.e., same number of agents $(N = 10)$ and same number of iterations $(T = 100)$). It is clear that the HHOCM and ROBLHHOCM algorithm present fast convergence on both datasets. Also, the convergence behavior of HHOCM is more accelerated than that of ROBLHHOCM algorithm for large dataset in the case of MAO dataset. Additionally, for QSAR Biodegradation dataset, the convergence behavior of ROBLHHOCM is faster than that of WOA, ALO, DA, GWO, GOA, SSA and HHOCM. Moreover, the convergence of the HHOCM and ROBLHHOCM algorithms show that the optimal values of fitness coincide perfectly with the optimal value of accuracy. This phenomenon can be explained by the effective trade-off balance between exploration and exploitation due to the integration of genetic operators and the use of random OBL operator. Figure 5 shows box plots of the accuracy for all datasets achieved by the competitor algorithms and the proposed variants of HHOCM. From this representation, one can determine first quartile (Q_1), third quartile (Q_3), maximum and minimum values. The red line inside the box indicates the median value. It is important to emphasize that each box is obtained after 30 runs of each algorithms. Looking closely to Fig. 5, it can be concluded that the HHOCM and ROBLHHOCM algorithms have higher box plots for both datasets than the other SI algorithms. The third place is taken by ALO on the QSAR Biodegradation dataset, and WOA shows the third high box on the MAO dataset.

Table 14 The average classification accuracy of all competing optimizers

Algorithms	QSAR Biodegradation dataset		The MAO dataset	
	AVG_{Acc}	Std_{Acc}	AVG_{Acc}	Std_{Acc}
DA	0.8613	0.0087	0.9214	0.0391
WOA	0.8605	0.0079	0.9738	0.0350
GOA	0.8360	0.0091	0.8667	0.0247
ALO	0.8852	0.0058	0.9333	0.0181
GWO	0.8507	0.0101	0.9119	0.0360
SSA	0.8744	0.0104	0.9262	0.0130
HHO	0.8798	0.0082	0.9476	0.0372
OBLHHO	0.8826	0.0089	0.9738	0.0350
ROBLHHO	0.9039	0.0096	0.9786	0.0300
HHOCM	0.9079	0.0080	**1.0000**	0.0000
OBLHHOCM	0.9081	0.0079	**1.0000**	0.0000
ROBLHHOCM	**0.9084**	**0.0078**	**1.0000**	0.0000

Table 15 The average size of selected features of all competing optimizers

Algorithms	QSAR Biodegradation dataset		The MAO dataset	
	AVG_{size}	Std_{size}	AVG_{size}	Std_{size}
DA	19.5000	4.0065	348.6667	233.8507
WOA	20.2333	7.1036	45.2333	77.8055
GOA	20.7333	2.9935	811.4667	16.7038
ALO	16.1667	2.4925	712.3000	11.5345
GWO	19.9000	5.5916	419.5333	69.7581
SSA	18.5000	2.9798	754.1333	16.4981
HHO	19.9667	4.4527	15.6333	5.1561
OBLHHO	18.8667	3.9596	10.1333	3.8889
ROBLHHO	18.6667	3.3333	9.9333	2.6667
HHOCM	14.8667	3.1703	4.5667	1.9270
OBLHHOCM	14.0000	3.0568	4.4667	1.8667
ROBLHHOCM	**13.6667**	**2.3333**	**4.3333**	**1.3333**

Table 16 The average sensitivity of the optimizers

Algorithms	QSAR Biodegradation dataset		The MAO dataset	
	AVG_{Sn}	Std_{Sn}	AVG_{Sn}	Std_{Sn}
DA	0.8430	0.0161	0.9374	0.0338
WOA	0.8433	0.0170	0.9752	0.0353
GOA	0.8181	0.0175	0.8963	0.0192
ALO	0.8758	0.0159	0.9481	0.0141
GWO	0.8365	0.0173	0.9315	0.0280
SSA	0.8599	0.0182	0.9426	0.0101
HHO	0.8774	0.0121	0.9533	0.0503
OBLHHO	0.8611	0.0168	0.9592	0.0574
ROBLHHO	0.8800	0.0330	0.9666	0.0462
HHOCM	0.8805	0.0111	**1.0000**	**0.0000**
OBLHHOCM	0.8900	0.0079	**1.0000**	**0.0000**
ROBLHHOCM	**0.8974**	**0.0062**	**1.0000**	**0.0000**

Table 17 The average precision of the optimizers

Algorithms	QSAR Biodegradation dataset		The MAO dataset	
	AVG_{Pr}	Std_{Pr}	AVG_{Pr}	Std_{Pr}
DA	0.8150	0.0112	0.9127	0.0386
WOA	0.8143	0.0100	0.9728	0.0375
GOA	0.7845	0.0110	0.8651	0.0206
ALO	0.8431	0.0070	0.9222	0.0211
GWO	0.8018	0.0121	0.9036	0.0322
SSA	0.8303	0.0124	0.9147	0.0109
HHO	0.8609	0.0086	0.9342	0.0459
OBLHHO	0.8545	0.0104	0.9797	0.0301
ROBLHHO	0.9633	0.0396	0.9800	0.0266
HHOCM	0.9763	0.0426	**1.0000**	**0.0000**
OBLHHOCM	0.9745	0.0398	**1.0000**	**0.0000**
ROBLHHOCM	**0.9797**	**0.0030**	**1.0000**	**0.0000**

5.6 Comparison of HHOCM variants with the existing algorithms

For proving the efficiency of HHOCM,OBLHHOCM and ROBLHHOCM, some results from literature on the same datasets are reported in Table 20. This task becomes difficult because researchers use different parameters configuration, specially in terms of population size (N) and maximum number of iterations (T) in swarm algorithms. For explaining this situation, the work of [34] used eight solutions as population size ($N = 8$) and and maximum number of iterations ($T = 200$). Also, a recent work is developed by [35], which employed other values of parameters (i.e., population size ($N = 30$) and and maximum number of iterations (T = 100, 500 and 1000) and the type of classifiers (k-NN and SVM)). To solve this issue, other experiments are added using same parameters configurations for realizing a fair comparison between HHOCM, OBLHHOCM, ROBLHHOCM and other swarm optimizers from the literature.

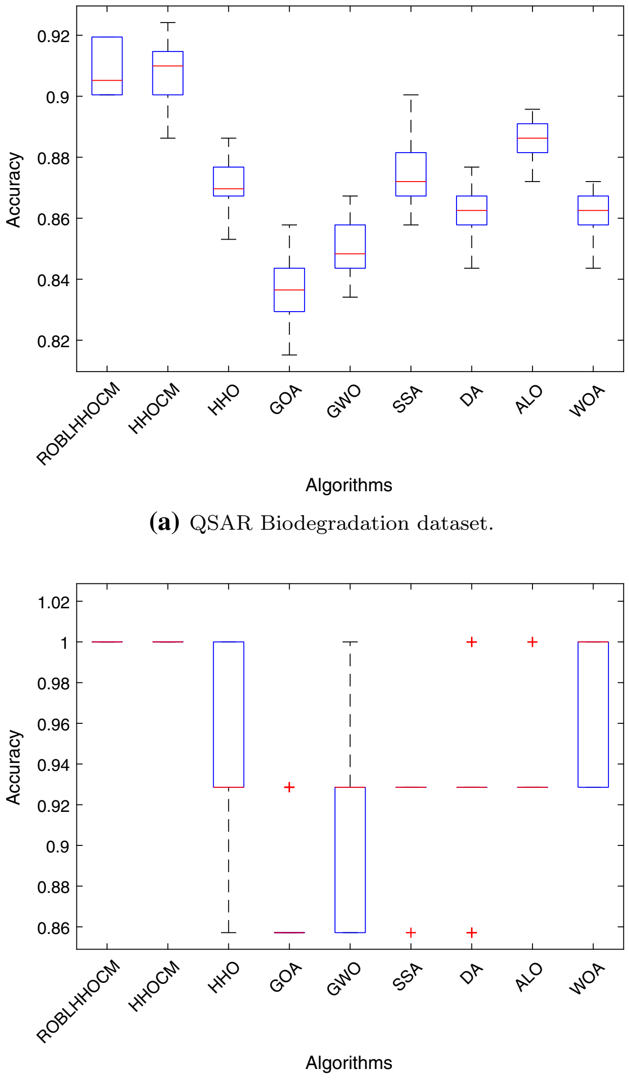
MAO dataset Table 20 presents results of the three versions of the HHOCM algorithm and other competitor algorithms including SSA, MFO, PSO, GOA, SCA, DCNN and HSGE. According to these results, it is obvious that the competitor algorithms are not so good as ROBLHHOCM or HHOCM, where the SSA optimizer achieved 87.35% with keeping 783.55 molecular descriptors, while ROBLHHOCM obtained higher rate of classification with 100% and keeping only 4.3333 MD from 1665 features in the case of ($N = 10$ and $T = 100$). Additionally, ROBLHHOCM showed 100% as accuracy and 4.6667

Table 18 The average CPU time of the optimizers

Algorithms	QSAR Biodegradation dataset		The MAO dataset	
	AVG_{Time}	Std_{Time}	AVG_{Time}	Std_{Time}
DA	13.9586	0.5285	22.6599	0.2870
WOA	13.1186	0.6489	**10.5675**	0.5239
GOA	13.4102	0.3528	12.0696	**0.2393**
ALO	13.1269	**0.2352**	11.6625	0.1440
GWO	13.5632	0.3726	12.6311	0.3561
SSA	**13.0468**	0.2983	11.6254	0.3799
HHO	27.8865	1.2023	21.3808	0.8292
OBLHHO	33.6254	2.2675	25.2581	1.6709
ROBLHHO	34.6751	1.2491	26.2148	1.3641
HHOCM	36.4030	3.6827	27.2690	1.2057
OBLHHOCM	37.3333	1.2654	28.6941	1.3619
ROBLHHOCM	37.6714	2.9699	29.3689	1.5891

molecular descriptors in the case of ($N = 30$ and $T = 100$). In [35], the HHO-k-NN with same conditions achieved 96.9% as accuracy in the case of ($N = 30$ and $T = 100$), while DCNN as classifier is ranked in third position with 75.14% of accuracy.

QSAR Biodegradation dataset The obtained results listed in Table 20 on the QSAR Biodegradation dataset prove that the best optimizer is ROBLHHOCM, which achieved 90.84% of accuracy in the case of ($N = 10$ and $T = 100$) followed by OBLHHOCM approach which achieved 90.81% of accuracy. Also, the approaches which used deep learning methods show that DeepBioD+ presented a good behavior of performance because it achieved 90%. In the case of ($N = 30$ and $T = 100$), an accuracy of 85.9% was achieved using HHO-k-NN in [35]. For approaches, which use ML methods such as the ANN-SVM, only 82% of compounds activity was recognized. It is important to highlight that the lower number of molecular descriptors for the QSAR Biodegradation dataset is obtained by ROBLHHOCM, which equals to 13.6667 in the case of ($N = 10$ and $T = 100$). Also, HHOCM, *MoDeSuS* and OBLHHOCM determined a low number of features around 15 MD. Thus, the proposed ROBLHHOCM shows a



(a) QSAR Biodegradation dataset.

(b) The MAO dataset.

Fig. 5 Boxplot of the HHOCM and ROBLHHOCM algorithms against other SI algorithms

Table 19 Wilcoxon rank-sum test

ROBLHHOCM versus	QSAR Biodegradation dataset p value	The MAO dataset p value
DA	**2.00E−11**	**1.71E−11**
WOA	**1.89E−11**	**2.85E−04**
GOA	**2.03E−11**	**8.64E−14**
ALO	**1.86E−11**	**7.15E−13**
GWO	**2.07E−11**	**1.19E−12**
SSA	**2.06E−11**	**2.71E−14**
HHO	**2.01E−11**	**2.10E−08**
HHOCM	1	NA

Table 20 Comparison with some existing algorithms

Algorithms	MAO dataset		Algorithms	QSAR dataset	
	AVG_{Acc}	AVG_{size}		AVG_{Acc}	AVG_{size}
SSA [34]	87.35%	783.55	*MoDeSuS* [37]	84%	15
MFO [34]	86.87%	266	Biclustering [38]	88.81%	19
PSO [34]	84.93%	250	DeepBioD [43]	87.50%	*
SCA [34]	79.41%	1066	DeepBioD+ [43]	90%	*
GOA [34]	85.14%	691.64	Relief-f DE [33]	85.40%	16
DCNN [45]	75.14%	*	ANN-SVM [41]	82%	*
HSGE [42]	95.71%	*	ChemNet [44]	86.7	*
HHO-*k*-NN [35] ($N = 30$ and $T = 100$)	96.9%	*	HHO-*k*-NN [35] ($N = 30$ and $T = 100$)	85.9%	*
HHOCM ($N = 10$ and $T = 100$)	100%	4.5667	HHOCM ($N = 10$ and $T = 100$)	90.79%	14.8667
OBLHHOCM ($N = 10$ and $T = 100$)	100%	4.4667	OBLHHOCM ($N = 10$ and $T = 100$)	90.81%	14
ROBLHHOCM ($N = 10$ and $T = 100$)	100%	4.3333	ROBLHHOCM ($N = 10$ and $T = 100$)	90.84%	13.6667
ROBLHHOCM ($N = 8$ and $T = 200$)	100%	4.6667	ROBLHHOCM ($N = 8$ and $T = 200$)	90.05%	13.8333
ROBLHHOCM ($N = 30$ and $T = 100$)	100%	4.6667	ROBLHHOCM ($N = 30$ and $T = 100$)	90.76%	16

powerful efficiency on both datasets because the performance is more efficient using both metrics (accuracy and average number of features) which has high accuracy with low number of molecular descriptors compared to competitor algorithms as reported in Table 20. This behavior can be interpreted by the incorporation of genetic operators to HHO and the random OBL operator which allows to enhance clearly the diversity of the population and the exploitation step. However, the proposed HHOCM, OBLHHOCM and ROBLHHOCM algorithms suffer from certain drawbacks as the computation time and the subset of selected molecular descriptors change according to the execution that may cause confusion for users.

6 Conclusion

In fields of cheminformatics research, QSAR is an important model that predicts the biological activities and physiochemical properties of chemical compounds. QSAR presents a real challenge problem because the representation of chemical compounds requires several features (i.e., high dimensionality problem is provoked). FS based on SI algorithms has become an efficient solution for keeping the prominent features and removing irrelevant data. To tackle with the previous challenges, this paper has proposed a three hybrid wrapper FS algorithms called HHOCM, OBLHHOCM and ROBLHHOCM which combined HHO with the genetic operators assisted by OBL strategies for selecting the proper chemical descriptor. The introduced wrapper FS is based on the variants of HHOCM and integrating the *k*-NN classifier that provides accurate and fast classification rates. To evaluate the proposed variants of the HHOCM algorithms, two common datasets of chemical information: the MAO dataset and the QSAR Biodegradation dataset, are considered in the performance evaluation process. The quantitative results revealed that the proposed algorithms HHOCM, OBLHHOCM and ROBLHHOCM achieve significant performance compared to seven well-established SI algorithms including the basic HHO, GWO, ALO, DA, WOA, GOA and SSA on both datasets. Moreover, it is concluded that the proposed ROBLHHOCM algorithm outperformed the competitor algorithms in terms of average standard deviations of fitness, accuracy, number of selected features, sensitivity and precision.

As a future work, the three variants of HHOCM can be used as multi-objective global optimization or FS paradigm for high dimensional with small instance in order to synchronously increase the classification rate and decrease the selection ratio of attributes. Another investigation is to consider the implementation of the HHOCM in parallel way to reduce the computation time.

Compliance with ethical standards

Conflict of interest No conflict of interest is declared by all authors.

References

1. Katsila T, Spyroulias GA, Patrinos GP, Matsoukas M-T (2016) Computational approaches in target identification and drug discovery. Comput Struct Biotechnol J 14:177–184

2. Lavecchia A (2015) Machine-learning approaches in drug discovery: methods and applications. Drug Discov Today 20(3):318–331
3. Hassan Baig M, Ahmad K, Roy S, Mohammad Ashraf J, Adil M, Haris Siddiqui M, Khan S, Amjad Kamal M, Provazník I, Choi I (2016) Computer aided drug design: success and limitations. Curr Pharm Des 22(5):572–581
4. Khan AU (2016) Descriptors and their selection methods in QSAR analysis: paradigm for drug design. Drug Discov Today 21(8):1291–1302
5. Masand VH, Rastija V (2017) PyDescriptor: a new PyMOL plugin for calculating thousands of easily understandable molecular descriptors. Chemometr Intell Lab Syst 169:12–18
6. Hashim FA, Houssein EH, Hussain K, Mabrouk MS, Al-Atabany W (2019) A modified Henry gas solubility optimization for solving Motif discovery problem. Neural Comput Appl 32(14):10759–10771
7. Lo Y-C, Rensi SE, Torng W, Altman RB (2018) Machine learning in chemoinformatics and drug discovery. Drug Discov Today 23(8):1538–1546
8. Neggaz N, Ewees AA, Elaziz MA, Mafarja M (2019) Boosting salp swarm algorithm by sine cosine algorithm and disrupt operator for feature selection. Expert Syst Appl 145:113103
9. Neggaz N, Houssein EH, Hussain K (2020) An efficient Henry gas solubility optimization for feature selection. Expert Syst Appl 152:113364
10. Cai J, Luo J, Wang S, Yang S (2018) Feature selection in machine learning: a new perspective. Neurocomputing 300:70–79
11. Sheikhpour R, Sarram MA, Gharaghani S, Chahooki MAZ (2017) A survey on semi-supervised feature selection methods. Pattern Recogn 64:141–158
12. Hashim FA, Houssein EH, Mabrouk MS, Al-Atabany W, Mirjalili S (2019) Henry gas solubility optimization: a novel physics-based algorithm. Future Gener Comput Syst 101:646–667
13. Li J, Cheng K, Wang S, Morstatter F, Trevino RP, Tang J, Liu H (2018) Feature selection: a data perspective. ACM Comput Surv 50(6):94
14. Apolloni J, Leguizamón G, Alba E (2016) Two hybrid wrapper-filter feature selection algorithms applied to high-dimensional microarray experiments. Appl Soft Comput 38:922–932
15. Heidari AA, Mirjalili S, Faris H, Aljarah I, Mafarja M, Chen H (2019) Harris Hawks optimization: algorithm and applications. Future Gener Comput Syst 97:849–872
16. Mirjalili S, Mirjalili SM, Lewis A (2014) Grey wolf optimizer. Adv Eng Softw 69:46–61
17. Mirjalili S, Lewis A (2016) The whale optimization algorithm. Adv Eng Softw 95:51–67
18. Mirjalili S, Gandomi AH, Mirjalili SZ, Saremi S, Faris H, Mirjalili SM (2017) Salp swarm algorithm: a bio-inspired optimizer for engineering design problems. Adv Eng Softw 114:163–191
19. Mirjalili S (2015) The ant lion optimizer. Adv Eng Softw 83:80–98
20. Saremi S, Mirjalili S, Lewis A (2017) Grasshopper optimisation algorithm: theory and application. Adv Eng Softw 105:30–47
21. Mirjalili S (2016) Dragonfly algorithm: a new meta-heuristic optimization technique for solving single-objective, discrete, and multi-objective problems. Neural Comput Appl 27(4):1053–1073
22. Forli S, Huey R, Pique ME, Sanner MF, Goodsell DS, Olson AJ (2016) Computational protein-ligand docking and virtual drug screening with the autodock suite. Nat Protoc 11(5):905
23. Jamali AA, Ferdousi R, Razzaghi S, Li J, Safdari R, Ebrahimie E (2016) Drugminer: comparative analysis of machine learning algorithms for prediction of potential druggable proteins. Drug Discov Today 21(5):718–724
24. Yuan S, Chan HS, Filipek S, Vogel H (2016) PyMOL and Inkscape bridge the data and the data visualization. Structure 24(12):2041–2042
25. Elaziz MA, Moemen YS, Hassanien AE, Xiong S (2018) Quantitative structure-activity relationship model for HCVNS5B inhibitors based on an antlion optimizer-adaptive neuro-fuzzy inference system. Sci Rep 8(1):1506
26. Hussien AG, Hassanien AE, Houssein EH, Bhattacharyya S, Amin M (2019) S-shaped binary whale optimization algorithm for feature selection, recent trends in signal and image processing. Springer, Berlin, pp 79–87
27. Taradeh M, Mafarja M, Heidari AA, Faris H, Aljarah I, Mirjalili S, Fujita H (2019) An evolutionary gravitational search-based feature selection. Inf Sci 497:219–239
28. Zheng Y, Li Y, Wang G, Chen Y, Xu Q, Fan J, Cui X (2018) A novel hybrid algorithm for feature selection based on whale optimization algorithm. IEEE Access 7:14908–14923
29. Ghosh M, Guha R, Sarkar R, Abraham A (2020) A wrapper-filter feature selection technique based on ant colony optimization. Neural Comput Appl 32:7839–7857
30. Faris H, Mafarja MM, Heidari AA, Aljarah I, AlàM A-Z, Mirjalili S, Fujita H (2018) An efficient binary salp swarm algorithm with crossover scheme for feature selection problems. Knowl Based Syst 154:43–67
31. Mafarja M, Mirjalili S (2018) Whale optimization approaches for wrapper feature selection. Appl Soft Comput 62:441–453
32. Mafarja M, Aljarah I, Heidari AA, Faris H, Fournier-Viger P, Li X, Mirjalili S (2018) Binary dragonfly optimization for feature selection using time-varying transfer functions. Knowl Based Syst 161:185–204
33. Hussien AG, Hassanien AE, Houssein EH (2017)Swarming behaviour of salps algorithm for predicting chemical compound activities. In: Eighth international conference on intelligent computing and information systems, IEEE, pp 315–320
34. Houssein EH, Hosney ME, Oliva D, Mohamed WM, Hassaballah M (2020) A novel hybrid Harris hawks optimization and support vector machines for drug design and discovery. Comput Chem Eng 133:106656
35. Houssein EH, Hosney MEA, Elhoseny M, Oliva D, Mohamed WM, Hassaballah M (2020) Hybrid Harris hawks optimization with cuckoo search for drug design and discovery in chemoinformatics. Sci Rep 10(1):1–22
36. Zainudin M, Sulaiman M, Mustapha N, Perumal T, Nazri A, Mohamed R, Manaf S (2017) Feature selection optimization using hybrid relief-f with self-adaptive differential evolution. Int J Intell Eng Inform 10(3):21–29
37. Martínez MJ, Razuc M, Ponzoni I (2019) MoDeSuS: A machine learning tool for selection of molecular descriptors in QSAR studies applied to molecular informatics. BioMed Res Int 1–12
38. Martínez MJ, Dussaut JS, Ponzoni I (2018) Biclustering as strategy for improving feature selection in consensus QSAR modeling. Electron Notes Discrete Math 69:117–124
39. Goh GB, Hodas NO, Vishnu A (2017) Deep learning for computational chemistry. J Comput Chem 38(16):1291–1307
40. Sabando MV, Ponzoni I, Soto AJ (2019) Neural-based approaches to overcome feature selection and applicability domain in drug-related property prediction. Appl Soft Comput 85:105777
41. Putra RID, Maulana AL, Saputro AG (2019) Study on building machine learning model to predict biodegradable-ready materials. In: AIP conference proceedings 60003–600010
42. Dutta A, Riba P, Lladós J, Fornés A (2020) Hierarchical stochastic graphlet embedding for graph-based pattern recognition. Neural Comput Appl 32:11579–11596
43. Goh GB, Sakloth K, Siegel C, Vishnu A, Pfaendtner J (2018) Multimodal deep neural networks using both engineered and

learned representations for biodegradability prediction. arXiv:1808.04456

44. Goh GB, Siegel C, Vishnu A, Hodas N (2018) Using rule-based labels for weak supervised learning: a ChemNet for transferable chemical property prediction. In: 24th ACM SIGKDD international conference on knowledge discovery and data mining. ACM, pp 302–310
45. Atwood J, Towsley D (2016) Diffusion-convolutional neural networks. Adv Neural Inf Process Syst 29:1993–2001
46. Emary E, Zawbaa HM, Sharawi M (2019) Impact of Lévy flight on modern meta-heuristic optimizers. Appl Soft Comput 75:775–789
47. Han F, Yang C, Wu Y-Q, Zhu J-S, Ling Q-H, Song Y-Q, Huang D-S (2017) A gene selection method for microarray data based on binary PSO encoding gene-to-class sensitivity information. IEEE/ACM Trans Comput Biol Bioinf 14(1):85–96
48. Qian F, Chen Z, Xia Z (2020) A novel quasi-reflected Harris hawks optimization algorithm for global optimization problems. Soft Comput 24:14825–14843
49. Gupta S, Deep K, Heidari AA, Moayedi H, Wang M (2020) Opposition-based learning Harris Hawks optimization with advanced transition rules: principles and analysis. Expert Syst Appl 158:113510
50. Chen H, Heidari AA, Chen H, Wang M, Pan Z, Gandomi AH (2020) Multi-population differential evolution-assisted Harris hawks optimization: Framework and case studies. Future Gener Comput Syst 111:175–198
51. Andersen JL, Flamm C, Merkle D, Stadler PF (2016) A software package for chemically inspired graph transformation. In: International conference on graph transformation. pp 73–88
52. Ruiz IL, Gómez-Nieto MA (2017) Advantages of relative versus absolute data for the development of quantitative structure-activity relationship classification models. J Chem Inf Model 57(11):2776–2788
53. Ewees AA, Abd Elaziz M, Houssein EH (2018) Improved grasshopper optimization algorithm using opposition-based learning. Expert Syst Appl 112:156–172
54. Tizhoosh HR (2005) Opposition-based learning: a new scheme for machine intelligence. In: International conference on computational intelligence for modelling, control and automation and international conference on intelligent agents, web technologies and internet commerce (IEEE CIMCA-IAWTIC'06) 1. pp 695–701
55. Long W, Jiao J, Liang X, Cai S, Xu M (2019) A random opposition-based learning grey wolf optimizer. IEEE Access 7:113810–113825