Expert Systems With Applications 204 (2022) 117486

Contents lists available at ScienceDirect

Expert Systems With Applications

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

Novel optimized crow search algorithm for feature selection

Behrouz Samieiyan [a,*], Poorya MohammadiNasab [b], Mostafa Abbas Mollaei [c], Fahimeh Hajizadeh [d], Mohammadreza Kangavari [a]

[a] Department of Computer Engineering, Iran University of Science and Technology, Tehran, Iran
[b] Department of Electrical and Computer Engineering, University of Kashan, Kashan, Iran
[c] Department of Electrical and Computer Engineering, Tehran University, Tehran, Iran
[d] Department of Electrical and Computer Engineering, Kharazmi University, Tehran, Iran

ARTICLE INFO

Keywords:
Feature selection
Crow search algorithm
Metaheuristic method
Feature reduction
Classification

ABSTRACT

Feature selection techniques have been presented to allow us to choose a small subset of the original components' relevant features by removing irrelevant or redundant features. Feature selection is essential for many reasons such as simplification, performance, computational efficiency, and quality interpretability. Owing to the importance mentioned above, many researchers have proposed and developed many algorithms to solve the feature selection problem. Although these approaches produce useful results, they possess some shortcomings like inadequate feature reduction. In this paper, a novel feature selection algorithm based on the crow search algorithm is presented. The algorithm uses dynamic awareness probability to keep the balance between the local and global search processes. Moreover, a novel neighborhood assigning strategy has been introduced to optimize the local search. Considering the best-selected features in each iteration helps attain more benefits in global search. The main superiority of the proposed algorithm is the significant feature reduction along with retaining the accuracy. Compared to enhanced crow search algorithm, the proposed algorithm has improved the feature reduction metric and fitness metric by 27.12% and 5.16%, respectively, while losing the accuracy metric by only 0.53%. Several popular UCI datasets have been employed to evaluate the proposed feature selection algorithm. The experimental results show that the proposed algorithm outperformed other feature selection algorithms in state-of-the-art related works regarding feature reduction and accuracy.

1. Introduction

Feature Selection (FS) is one of the core concepts in machine learning and data mining (Crone, Lessmann, & Stahlbock, 2006). Recently, datasets have been very massive in several attributes and instances. Finding an attribute (feature) subset which preserves some particular properties of the original data without redundancy is the aim of FS (Dy, Brodley, Kak, Broderick, & Aisen, 2003). FS selects some of the most useful features from a group of features to reduce the feature space dimension. According to these properties of FS, it has been utilized in different applications, including classification and regression (Zhang, Mistry, Lim, & Neoh, 2018), optimization problems (Sadeghian, Akbari, & Nematzadeh, 2021), text classification (Labani, Moradi, Ahmadizar, & Jalili, 2018), therefore, FS has become an important research subject in recent years.

FS techniques can be classified into three different approaches: filters (Liu & Motoda, 1998), wrappers (Kohavi, John, et al., 1997) and hybrids. Fig. 1 shows these three different approaches.

The wrapper-based approaches are simple algorithms that select the best subset of features by using a classifier. It is worth mentioning that wrapper-based methods are computationally expensive (Dy & Brodley, 2004; Kim, Street, & Menczer, 2002) and they may have the risk of overfitting the model. On the other hand, filter-based approaches select the optimal feature subset according to some criteria (Bolón-Canedo, Sánchez-Marono, Alonso-Betanzos, Benítez, & Herrera, 2014). Also, the filter-based methods are faster and independent of the classifier and have lower performance than wrapper approaches. The third approach, i.e., the hybrid algorithms, combines the filter-based and wrapper-based methods (Xue, Zhang, Browne, & Yao, 2015). The search process includes the FS and the classifier learning integrated into a single operation in this approach. In the past few years, the hybrid algorithms have attracted a great deal of attention from researchers, and the computation approaches have been using to improve the hybrid algorithms. In Refs. Anter and Ali (2020), Bui, Tsangaratos, Ngo, Pham,

* Corresponding author.
E-mail addresses: samieiyan@alumni.iust.ac.ir (B. Samieiyan), poorya.m.n@std.kashanu.ac.ir (P. MohammadiNasab), m.abbasmollaei@ut.ac.ir (M.A. Mollaei), std_fahime.hajizadeh@khu.ac.ir (F. Hajizadeh), kangavari@iust.ac.ir (M. Kangavari).

https://doi.org/10.1016/j.eswa.2022.117486
Received 27 February 2021; Received in revised form 5 April 2022; Accepted 30 April 2022
Available online 14 May 2022
0957-4174/

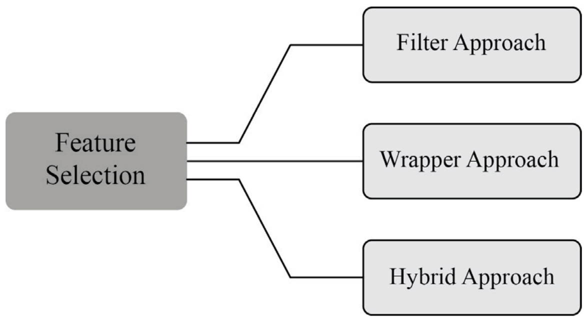

Fig. 1. Three different approaches of feature selection.

and Pham (2019) and Chen, Mi, and Lin (2020) fuzzy system has been combined with an FS algorithm.

Evolutionary Algorithms (EAs) have had a significant influence and performance in the FS process in recent years. In this regard, many algorithms have been proposed using this method such as Genetic Algorithm (GA) (Goldberg & Holland, 1988), Particle Swarm Optimization (PSO) (Kennedy & Eberhart, 1995) and Differential Evolution (DE) (Storn & Price, 1997). EA usually selects a random population and evaluates the subset's efficiency using the fitness function's iteration of the algorithm. EA uses two main operations to search the solution space: exploitation and exploration.

Exploration is the process of visiting the entire new regions of a search space. Still, exploitation is the process of visiting those regions of a search space within the neighborhood of previously visited points. For being successful, a search algorithm needs to establish a good ratio between exploration and exploitation. Many researchers believe that EAs are useful if there is a good ratio between exploration and exploitation. EA almost sticks in local optima because of the improper balancing between the local search (exploitation) and the global search (exploration).

One of the most common mathematical methods recently applied to boost the performance of EAs is called the chaos theory. Chaos is a mathematical approach that has been recently used to enhance the performance of the standard EAs. The chaos has been applied to different study fields such as chaos control (Zhang & Zhang, 2008) and some optimization research (Tavazoei & Haeri, 2007). In general, the chaos has three important dynamic properties including:

- the sensitive dependence on initial conditions,
- the quasi-stochastic property, and
- ergodicity.

Recently, FS algorithms have attracted researchers' wide attention, and many new ideas have been proposed. Crow search algorithm (CSA) (Askarzadeh, 2016) is one FS algorithm that attracted much attention from researchers since it was introduced. The evaluation results of CSA show that it is very efficient for optimization problems, especially problems that science and engineering have difficulty to solve. Meanwhile, this algorithm is easy to implement and has only a few parameters. However, it suffers from some problems such as entrapment in the local optimum due to the AP parameter. Furthermore, old versions of CSA use a random-based approach in both local and global search space.

Ref. Díaz et al. (2018) has utilized dynamic awareness probability (AP) and changed the two main features of CSA, i.e., AP and random perturbation, to prevent CSA from getting stuck in the local optima. Ref. Shi, Li, Zhang, and Huang (2017) has employed the adaptive weight factor to improve the local search. Then, Ref. Sayed, Hassanien, and Azar (2019) has employed the chaotic maps instead of the random parameter in the main algorithm. In Ref. Majhi, Sahoo, and Pradhan (2019), authors have replaced the random initialization with opposition based learning (OBL). Also, three new search approaches were introduced in Ref. Zamani, Nadimi-Shahraki, and Gandomi (2019) to enhance CSA performance. These methods are neighborhood-based local search (NLS), non-neighborhood global search (NGS), and wandering around based search (WAS). All of these algorithms are based on the CSA.

In this paper, the main effort is to reduce the number of the features considerably without failing to keep a useful accuracy metric because the most recent studies in the FS method have focused more on enhancing the accuracy and have not been able to reduce the features significantly, such as CSA, ICSA2, and BOA, while increasing the classification accuracy compared to previous algorithms, have a weak FR and do not help to reduce the dimensions. These studies have encouraging results in improving the knowledge related to this issue. However, requirements for developing more sophisticated approaches always exist.

The novelty of this research is to propose a new global search mechanism alongside a novel neighborhood concept to get better results in the global search and enhance the local search performance. Also, the proposed method has improved the balance between the local search and the global search. This paper utilizes datasets that other algorithms have used in the related articles as the only reference datasets. Further information has been described in Section 5.2.

The organization of this paper is as follows: Section 2 surveys prior works. In Section 3, the CSA has been explained. Section 4 deals with describing our proposed method. The experimental results have been included in Section 5. Finally, Section 6 presents the conclusions and future works.

2. Relevant works

In recent years, a wide range of FS algorithms has been proposed. Each of them uses a unique approach and has both advantages and disadvantages. According to No-Free-Lunch (NLF) theorem for optimization (Wolpert & Macready, 1997), no FS algorithm can be admitted ultimately better than the other ones.

Bio-inspired meta-heuristic algorithms are a new category of the FS algorithms that indicates a good influence and performance in optimization problems. Bio-inspired algorithms can be generally divided into two categories, including evolution algorithms and swarm intelligence algorithms. Some of the well-known and fundamental EAs are GA (Goldberg & Holland, 1988), evolution strategy (Beyer & Schwefel, 2002), and differential evolution (DE) (Storn & Price, 1997) algorithms. Since each algorithm has some imperfection, extensive research has been presented to improve the performance and the strength of these algorithms. For instance, a combination of three algorithms, including artificial bee colony (ABC) (Karaboga & Basturk, 2007), PSO and GA, were introduced to solve the optimization difficulty of the GA in Djellali, Djebbar, Zine, and Azizi (2018). In Ref. Maheshwari, Kumar, and Kumar (2016), based on the combination of GA and DE algorithms, a new algorithm was presented to increase the classification accuracy metric.

Swarm intelligence (SI) algorithms are another type of bio-inspired algorithm. SI algorithms are inspired by animal behavior. In these algorithms, complicated problems can be simulated, considering the animals' behavior until the optimum solution is found (BoussaïD, Lepagnot, & Siarry, 2013). In these algorithms, each animal is considered a search agent is exploring the search space using the simulated algorithm. To mention some of the critical SI algorithms, one can name PSO and ABC (Karaboga & Basturk, 2007). Based on what animal the SI algorithm is inspired by, SI algorithms can be divided into four categories considering the behavior of insects, terrestrial animals, aquatic animals, and birds.

Insects behavior based algorithms are the first category of the SI algorithms and to mention some of them, one can name ABC, GABC and qABC. In Ref. Karaboga and Basturk (2007), a new algorithm named

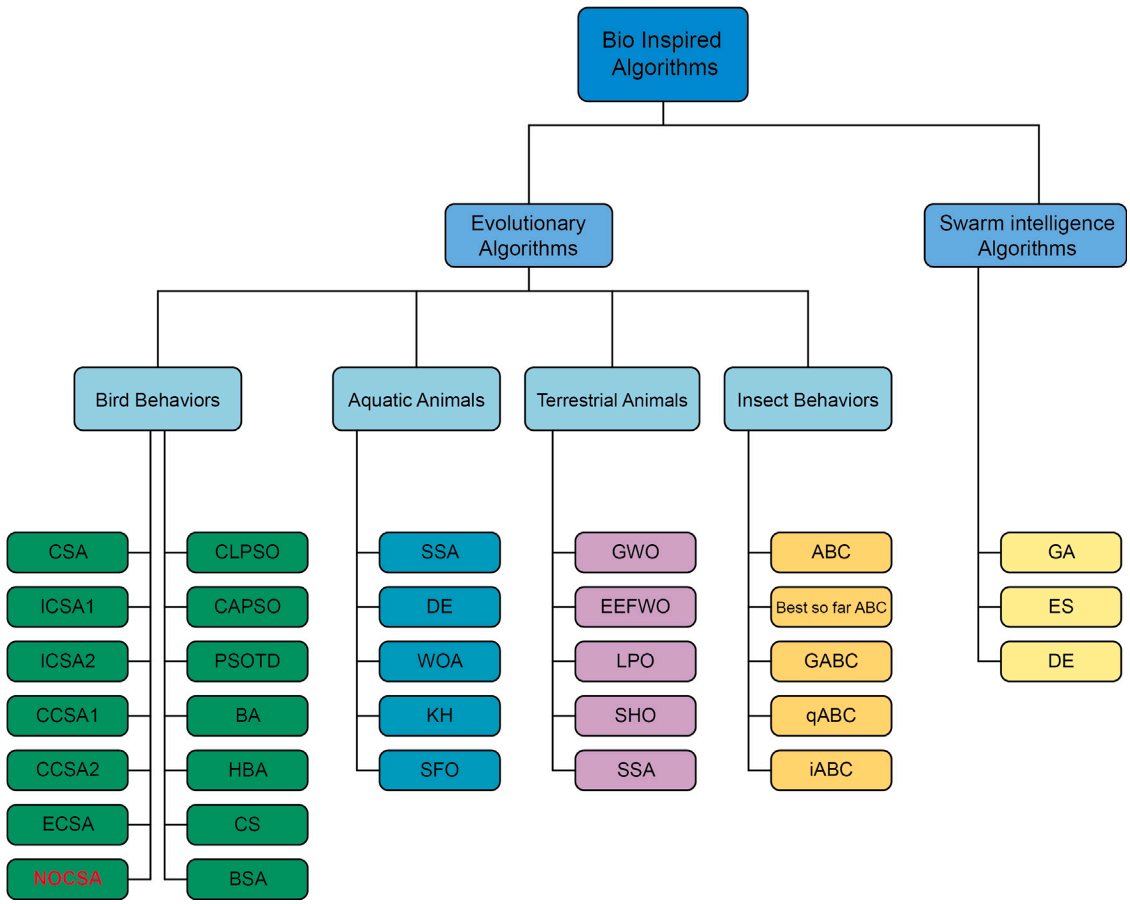

Fig. 2. Bio-inspired metaheuristic algorithms.

ABC was presented based on the behavior of three different groups of bees, but it has some problems such as global search weakness and failing to keep the balance between the local and global search. In Ref. Tsai, Pan, Liao, and Chu (2008), an algorithm with a new approach to the global search was introduced, and then Zhu and Kwong (2010) has presented the best-guided ABC (GABC) algorithm to balance the local and global search. Furthermore, the best so far ABC algorithm (Banharnsakun, Achalakul, & Sirinaovakul, 2011) has developed by analyzing different types of bees in the ABC algorithm. Then, in Ref. Karaboga and Gorkemli (2014) by presenting a quick ABC (qABC) algorithm, some problems such as local search problems and the ABC algorithm's convergence rate were improved.

Another category of SI algorithms are the ones based on the behavior of terrestrial animals which one of this category's well-known algorithms is the gray wolf optimizer (GWO) algorithm (Mirjalili, Mirjalili, & Lewis, 2014) inspired by wolves' hunting mechanism that it has inefficient performance in the global search. Later in Long, Jiao, Liang, and Tang (2018), the global search problem of GWO was solved. One of the other groups of terrestrial animals inspiring in optimization algorithms are lions, and the lion pride optimizer (LPO) algorithm (Wang, Jin, & Cheng, 2012) is one of the main algorithms developed founded on the lions' behavior. Some of the other terrestrial animals' algorithms are spotted hyena optimizer (SHO) (Dhiman & Kumar, 2017) and squirrel search algorithm (SSA) (Jain, Singh, & Rani, 2019) which are inspired by hyenas and squirrels, respectively.

The third category of the SI algorithms simulates the aquatic animals' behavior to solve the optimization problems. Some of the well-known aquatic animals' algorithms are salp swarm algorithm (SSA) (Mirjalili et al., 2017), dolphin echolocation (DE) (Kaveh, Vaez, Hosseini, & Fallah, 2016), whale optimizer algorithm (WOA) (Mirjalili & Lewis, 2016), krill herd (KH) (Gandomi & Alavi, 2012), and sailfish optimizer (SFO) (Shadravan, Naji, & Bardsiri, 2019). SSA is an algorithm that simulates salp behavior to solve multi-objective problems. DE is a dynamic algorithm based on the behavior of dolphins and is utilized for damage detection. The WOA mimics the social behavior of humpback whales. The bubble-net hunting strategy inspires the algorithm. KH is founded on the krill behavior and formulates the krill individuals' time-dependent position by three main factors, including movement induced by other individuals' presence, foraging activity, and random diffusion. SFO is another aquatic animals' algorithm that is inspired by sailfish behavior.

The last category of the SI algorithms is the algorithm on the basis of the behavior. To name some of the well-known birds' based algorithms, one can refer to comprehensive learning PSO (CLPSO) (Liang, Qin, Suganthan, & Baskar, 2006), centripetal accelerated PSO (CAPSO) (Beheshti, 2013) and PSO two differential (PSOTD) (Chen et al., 2017) which all of them improved PSO algorithm. For instance, CLPSO enables the diversity of the swarm to be preserved to discourage premature convergence. In CAPSO, Newtonian's motion laws have been employed to accelerate learning the process and increase accuracy in solving ML problems. PSOTD adapted two differential mutation operators with different characteristics. One mutation operator has good global exploration ability, and the other mutation operator has an excellent local exploitation ability. Bat algorithm (BA) (Yang & Gandomi, 2012) is one of the main algorithms in the birds' category that has attracted significant attention. Later in hybrid BA (HBA) (Liu, Wu, Xiao, Wang, & Zhang, 2018), three different changes have been made to improve the BA's performance. Some of the other algorithms in the birds' behavior-based algorithms are cuckoo search (CS) (Bulatović, Đorđević, & Đorđević, 2013), bird swarm algorithm (BSA) (Meng, Gao, Lu, Liu, & Zhang, 2016), and algorithms founded on CSA.

Table 1
Assigning neighbors for crows.

$Crow_i$	Neighbors($Crow_{i+1}$ to $Crow_{i+10}$)
$Crow_1$	Neighbors($Crow_2$ to $Crow_{11}$)
$Crow_2$	Neighbors($Crow_3$ to $Crow_{12}$)
$Crow_3$	Neighbors($Crow_4$ to $Crow_{13}$)
$Crow_4$	Neighbors($Crow_5$ to $Crow_{14}$)
$Crow_5$	Neighbors($Crow_6$ to $Crow_{15}$)
$Crow_6$	Neighbors($Crow_7$ to $Crow_{16}$)
$Crow_7$	Neighbors($Crow_8$ to $Crow_{17}$)

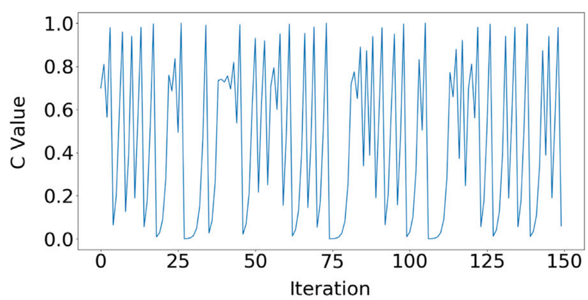

Fig. 3. Sine map.

Table 2
S-shaped and v-shaped families of transfer functions.

S-shaped family		V-shaped family	
Name	Transfer function	Name	Transfer function
S1	$T(x) = \frac{1}{1+e^{-2x}}$	V1	$T(x) = \lvert erf(\frac{\sqrt{\pi}}{2}x)\rvert$
S2	$T(x) = \frac{1}{1+e^{-x}}$	V2	$T(x) = \lvert \tanh(x)\rvert$
S3	$T(x) = \frac{1}{1+e^{(-x/2)}}$	V3	$T(x) = \lvert \frac{(x)}{\sqrt{1+x^2}}\rvert$
S4	$T(x) = \frac{1}{1+e^{(-x/3)}}$	V4	$T(x) = \lvert \frac{2}{\pi} arctan(\frac{\pi}{2}x)\rvert$

CSA is one of the well-known algorithms of the bird category. Some of the CSA problems are getting stuck in the local optimum and premature convergence rate, and in improved CSA (ICSA1) (Shi et al., 2017), the local search problem was solved by adding the weight factor. ICSA2, with changing two main features of the CSA, i.e., AP and random perturbation, removed the CSA problems (Díaz et al., 2018). CCSA1 has used chaos instead of a random factor (Sayed et al., 2019). Most recently, CCSA2 has introduced three approaches for the different positions of the crows (Zamani et al., 2019), and enhanced CSA (ECSA) has improved the performance metric by presenting a new approach in the local and global search using dynamic AP (Ouadfel & Abd Elaziz, 2020). All these algorithms have employed conservative approaches to maintain efficient accuracy. Hence, they were not capable of reducing the number of features significantly. The main focus of this paper is to select the most minimal subset of the features. Fig. 2 categorizes different bio-inspired algorithms mentioned in Section 2.

Using nature-inspired algorithms in Type-2 fuzzy logic controllers has recently yielded astonishing results (Valdez, Castillo, Cortes-Antonio, & Melin, 2020). In Miramontes, Guzman, Melin, and Prado-Arechiga (2018), BSA was used to increase the heart rate classification results using fuzzy systems. Also, in some cases, the use of fuzzy logic can improve the optimization results. For example, in Rodríguez et al. (2017), the famous GWO algorithm uses a fuzzy hierarchical operator. This operator is essentially a hierarchical transformation inspired by the Gray Wolf hierarchical social pyramid.

In algorithms such as Castillo and Amador-Angulo (2018), Lagunes, Castillo, Valdez, Soria, and Melin (2021), Roeva, Zoteva, and Castillo (2021) and Sánchez, Melin, and Castillo (2021), an attempt is made to select suitable parameters for the input of the optimization algorithm according to the problem, which is called dynamic parameter adaptation. Usually, these algorithms are based on fuzzy logic.

3. Overview of Crow search algorithm (CSA)

The CSA is a meta-heuristic algorithm based on the crows' social behavior, i.e., the mechanism for hiding their foods. The crows can conceal the foods and remember their place for several months. They live in a flock, and each tries to find the hiding place of the others' food. Also, the crows protect their food carefully by changing the hiding place regularly. A model is introduced to map the FS problem according to the crow's behavior. In this model, each crow position is a feature subset of the global feature set. Y_j^t is the position of the crow j in the iteration t. Each crow has a memory for the hiding place. N_i^t is the memory of the crow j in the iteration t. The memory of each crow holds the best position retrieved so far. In each iteration, the crow j wants to find the hiding place of the crow i. There are two possible scenarios:

- **Scenario 1:** The crow i does not know that the crow j is searching for its hiding place. So, the crow j will update the position by:

$$Y_j^{t+1} = Y_j^t + r_j \times fl_j^t \times (N_i^t - Y_j^t). \tag{1}$$

In Eq. (1), fl_j^t is the flight length of the crow i, and r_j is a random number between 0 and 1. N_j^t is the neighbor memory and Y_j^t is the current position of crow. This scenario is equivalent to the local search for CSA.

- **Scenario 2:** The crow i knows that the crow j is following the hiding place. Hence, the crow i changes the hiding place to protect it from the crow j. For this condition, the crow i chooses a random position. This scenario is equivalent to the global search of the algorithm.
These two scenarios are mathematically defined as follows:

$$Y_j^{t+1} = \begin{cases} Y_j^t + r_i \times fl_j^t \times (N_i^t - Y_j^t) & r_j \geq qAP_j^t \\ Select\ a\ random\ position & Otherwise, \end{cases} \tag{2}$$

where r_j is a random number between 0 and 1 and the AP_j^t is the awareness probability of the crow j in the iteration t. In other words, the AP value controls the balance between exploitation and exploration.

The CSA introduces a fitness function that measures the feasibility of each position. After updating the positions, the new position will be evaluated by the fitness function. If the crow's new position is better than the old one, the crow will update its memory to the new position. Eq. (3) shows this process mathematically:

$$N_i^{t+1} = \begin{cases} Y_i^{t+1} & Fitness(Y_i^{t+1}) > (Y_i^t) \\ N_i^t & Otherwise. \end{cases} \tag{3}$$

At the end of the iterations, the best position with the best fitness value is selected as the optimal solution. The pseudo-code of CSA is defined as Algorithm 1.

4. Proposed algorithm (Novel optimized CSA)

A significant CSA problem is unbalancing between the local and global search, which causes the CSA stuck in the local optima (Shi et al., 2017). The main reason for this problem is the AP parameter. Ref. Gong and Cai (2013) has inspired an idea to use the dynamic awareness probability (DAP). In this paper, we proposed an enhanced version of CSA. NOCSA is a novel approach has been introduced to improve both the local and global search. Moreover, selecting the best crows according to Pareto optimality in each iteration contributes to the proposed algorithm.

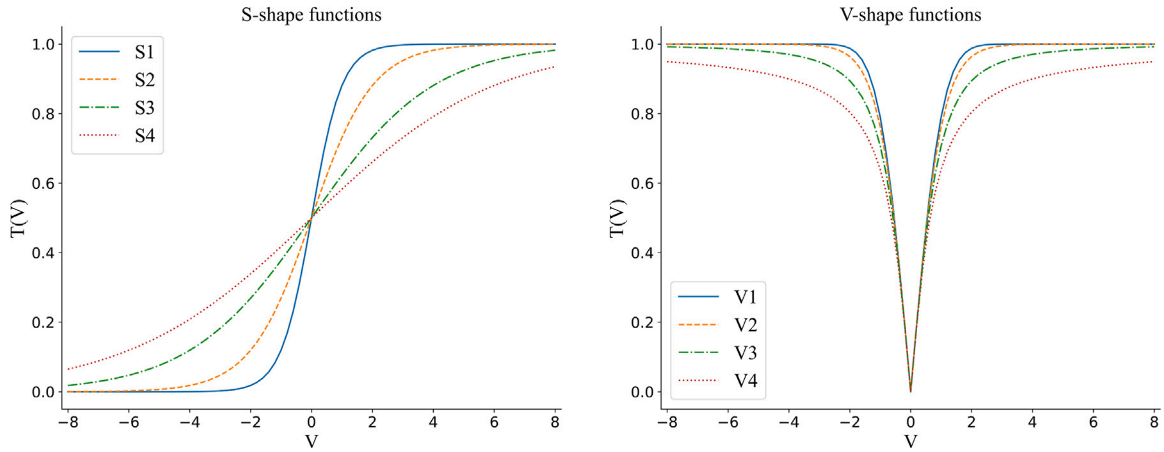

Fig. 4. S-shaped and v-shaped transfer functions.

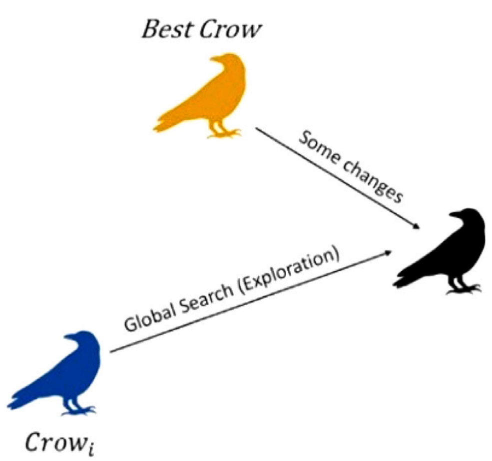

Fig. 5. Global search scheme.

Algorithm 1 Crow Search Algorithm (CSA)

AP, fl, the Maximum iteration (t_{Max})
The number of search agents (SA)
Binary initialization for the search agents (crows)
Calculate the fitness of each crow position
Initialize the crow's memory
Set current iteration $t = 1$
while $t \leq t_{Max}$ **do**
  for $i = 1 : SA$ **do**
    choose a crow randomly to follow it (neighborhood meaning)
    if $r_j \geq AP_j^t$ **then**
      do local search (Eq. (1))
    else
      do global search (select random position)
  Calculate fitness for each new crow position Position updating (Eq. (3))
  $t = t + 1$
report the best solution

4.1. Dynamic awareness probability (DAP)

The DAP technique's primary idea is to search the crows with strong fitness locally and the other crows with weak fitness search globally to achieve higher fitness. In the CSA, the AP parameter is constant and usually set to 0.1. When the DAP technique is employed in the CSA, each crow's AP parameter will be changed according to its rank. In this strategy, after initializing the crows, all fitness values are calculated. Then, the crows are sorted in the descending order of the fitness. Afterward, the AP value of the crow i will be calculated as defined in:

$$DAP_i = AP_{min} + (AP_{max} - AP_{min}) \times \frac{rank_i}{N}, \tag{4}$$

where $rank_i$ is the rank of the crow i among all the crows, N is the total number. The AP_{min} and AP_{max} are the lower and upper bound of the AP, respectively. The AP_{min} and AP_{max} are set to 0.1 and 0.8, respectively, which are obtained experimentally. The crows with higher rank have lower AP and vice versa. A threshold should be assigned for the DAP metric to classify the crows into two sets. The first set contains those crows with the AP lower than the defined threshold, and the second set contains the crows with the AP higher than the defined threshold. The crows in the first set will search locally, and the crows in the second set will search globally for the next iteration. The DAP threshold should be initialized at the beginning of the algorithm. The DAP value is defined so that 7 out of 10 crows search locally. Algorithm 2 deals with the classification of the crows in each iteration of the proposed algorithm using the DAP technique, as shown below:

Algorithm 2 DAP classification

if $DAP_i < DAP\ limit$ **then**
  exploitation (search locally)
else
  exploration (search globally)

4.2. Proposed local search approach

In the CSA, some crows are selected to search locally to grab some practical features from other crows. The selected crows should search in their neighborhood. This paper has introduced a novel assigning neighborhood strategy based on crow's fitness. As the crows are sorted at the beginning of each iteration in descending order of fitness, the crows at the top of the list have higher fitness than the others. The next ten crows have been defined as the neighbors to guide the crows towards the best possible features. For example, the neighbors of the crow i are started from the crow $i + 1$ to the crow $i + 10$. Table 1 shows the neighbors for the top 7 crows. In each iteration, only 17 crows are needed. Hence, at the end of each iteration, the top 17 crows with the higher fitness will be saved. That employs in the next iteration, and the others will be ignored. Each crow generates ten new crows by merging with its neighbors. So, the seven crows will generate 70 new crows totally.

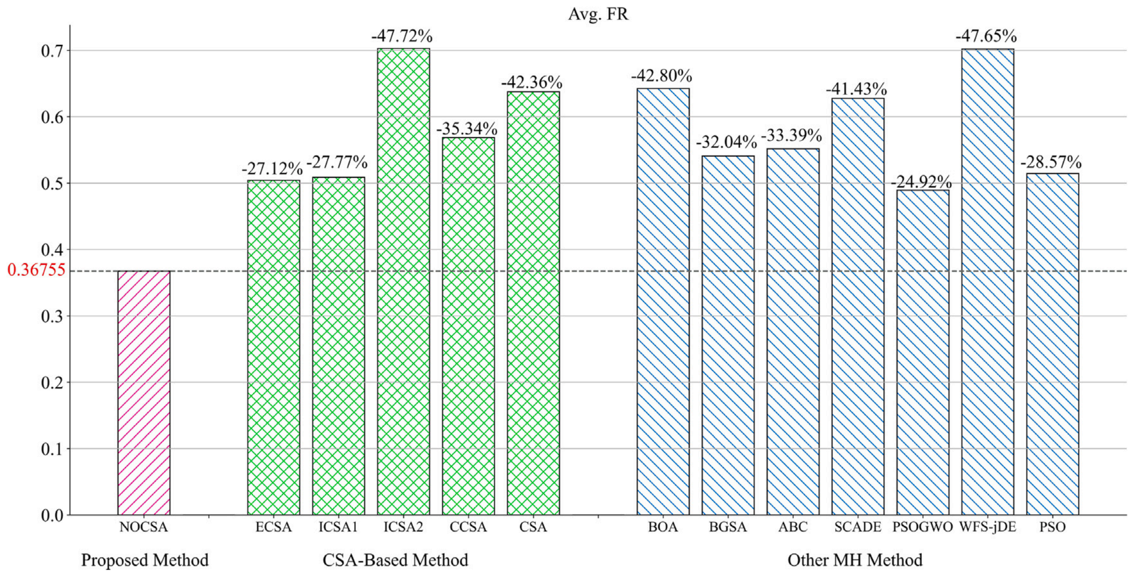

Fig. 6. Average of FR for all datasets.

Fig. 7. Average of fitness for all datasets.

The other contribution used to enhance the local search is defining a chaos function instead of a random function in Eq. (1). CCSA has inspired the use of a chaos function in this part of the proposed algorithm. The chaos function has been constructed from a chaotic sine map, and the formula is given in:

$$C_t = \begin{cases} \sin(\pi C_{t-1}) & t > 1 \\ 0.7 & t = 1. \end{cases} \tag{5}$$

Also, the chaotic sine diagram has been shown in Fig. 3. The coefficient of the chaos function coefficient has been defined so that the return values are between 0 and 1. The use of the chaos function instead of the random function improves the convergence rate.

After using the chaos parameter in the local search, the new exploitation equation can be expressed as:

$$Y_j^{t+1} = Y_j^t + C_j \times fl_j^t \times (N_i^t - Y_j^t), \tag{6}$$

where C_i is the value of the chaos function in the iteration i. N_i^t, Y_j^t and fl_j^t are same concepts in Eq. (1).

NOCSA is a binary FS algorithm because the input answer of this algorithm is crows composed of a binary string. The output of the algorithm is expressed as a binary string that represents the selected features. So in some parts of problem-solving, we need to convert continuous values to binary values.

In this article, we used eight transfer functions introduced by Mirjalili and Lewis (2013). We divided these functions into two main categories, s-shape and v-shape, due to the naming of their diagram shapes. Table 2 shows the formula for each of these functions and Fig. 4 illustrates their plot.

We tested the proposed algorithm with each of these functions on some of the datasets to select the best transfer function. Then we obtained the results of Table 3, which according to the results of the transfer table, V2 transfer function gives the best results compared to other functions.

4.3. Proposed global search approach

In each iteration of the CSA, some new crows are generated randomly by the global search. The global search has been modified using

Table 3
Comparison of S-shaped and v-shaped families of transfer functions.

Datasets	Stat. Measure	S-shape transfer functions				V-shape transfer functions			
		S1	S2	S3	S4	V1	V2	V3	V4
Breastcancer	Accuracy	0.9680	0.9665	0.9614	0.9594	0.9591	0.9617	0.9639	0.9562
	FR	0.5743	0.4657	0.4029	0.3057	0.2371	0.2286	0.2400	0.1943
	Fitness	0.0764	0.0786	0.0688	0.0726	0.0626	0.0593	0.0628	0.0622
CongressEW	Accuracy	0.9536	0.9485	0.9513	0.9545	0.9572	0.9632	0.9572	0.9582
	FR	0.5518	0.4661	0.3768	0.3429	0.2321	0.2000	0.1625	0.1339
	Fitness	0.0885	0.0913	0.0739	0.0768	0.0609	0.0568	0.0551	0.0489
Exactly1	Accuracy	0.8038	0.8280	0.8026	0.8200	0.9860	1.0000	0.9276	0.9866
	FR	0.5934	0.5253	0.4527	0.4857	0.4440	0.4659	0.4967	0.4571
	Fitness	0.2397	0.2086	0.2392	0.2143	0.0618	0.0537	0.1082	0.0628
Clean1	Accuracy	0.8676	0.8622	0.8689	0.8723	0.8710	0.8714	0.8769	0.8752
	FR	0.7495	0.6130	0.5242	0.5105	0.4592	0.4503	0.4024	0.3459
	Fitness	0.1951	0.1853	0.1724	0.1659	0.1639	0.1600	0.1481	0.1486
M-of-n	Accuracy	0.8898	0.9140	0.9086	0.8712	0.9770	1.0000	0.9426	0.9652
	FR	0.6462	0.6044	0.5077	0.4813	0.4308	0.4527	0.0994	0.4330
	Fitness	0.1640	0.1420	0.1392	0.1698	0.0730	0.0493	0.4044	0.0790
SonarEW	Accuracy	0.8346	0.8231	0.8269	0.8385	0.8423	0.8967	0.8346	0.8260
	FR	0.7162	0.5867	0.5329	0.4867	0.4795	0.4362	0.1848	0.3314
	Fitness	0.2236	0.2144	0.2128	0.1994	0.1899	0.1366	0.3767	0.1903
WineEW	Accuracy	0.9775	0.9764	0.9697	0.9483	0.9708	0.9794	0.9730	0.9573
	FR	0.5868	0.4549	0.4484	0.3297	0.3385	0.3231	0.2857	0.2703
	Fitness	0.0833	0.0795	0.0800	0.0804	0.0663	0.0516	0.0612	0.0687
Zoo	Accuracy	0.9287	0.9287	0.9089	0.9287	0.9248	0.9730	0.9386	0.9208
	FR	0.6321	0.1220	0.4518	0.4714	0.4054	0.3518	0.4071	0.3232
	Fitness	0.1304	0.5571	0.1389	0.1321	0.1120	0.0594	0.0990	0.1088
Avg. Accuracy		0.9030	0.9059	0.8998	0.8991	0.9360	0.9557	0.9268	0.9307
Avg. FR		0.6313	0.4798	0.4622	0.4267	0.3783	0.3636	0.2848	0.3112
Avg. Fitness		0.1501	0.1946	0.1406	0.1389	0.0988	0.0784	0.1644	0.0962

Table 4
List of used datasets.

Dataset	No. of features	No. of instance
Breastcancer	9	699
BreastEW	30	569
CongressEW	16	435
Exactly	13	1000
Exactly2	13	1000
HeartEW	13	270
IonosphereEW	34	351
Lymphography	18	148
Clean1	166	476
Clean2	166	6598
M-of-n	13	1000
SonarEW	60	208
SpectEW	22	267
Tic-tac-toe	9	958
Vote	16	300
WineEW	13	178
Semion	256	1593
Zoo	16	101
Isolet5	617	1559
Madelon	500	2600
Advertisements	1558	3279
CNAE	857	1080
Parkinson's Disease	754	756

Table 5
Setting parameters for the experiments.

Algorithm	Parameters	Value
ABC	Trail	100
	Modification rate (MR)	0.8
BOA	c	0.01
	α	0.1
	p	0.8
SCADE	α	2
	F, CR	0.3, 0.7
PSOGWO	c1, c2, c3	0.5
	w	0.5+rand()/2
WFS-jDE	CR	0.9
PSO	c1, c2	2

a new strategy to generate new crows. In this regard, a new crow always contains two groups of features. The first group represents all the best crow's features, and the second group is 20% of the all available features, chosen randomly. There may be some standard features in both groups, and this does not matter. The combination of these two groups of features is defined as the new crow's features. Thereby, all new crows are generated based on the best crow's features and some other random features. It is expected that these newly generated crows are more promising ones rather than previous ones due to placing the best crow as the primary basis for generating the new crows. In each iteration, 30 new crows are generated. A scheme of the proposed global search is shown in Fig. 5.

4.4. NOCSA for solving problems

In all datasets, there are usually features that are of no use or redundant. In these cases, the classification faces some difficulties. Also, in some cases, the number of features may be so high that the classification cannot be successful. Thus, FS algorithms try to provide an optimized subset of features that can solve this challenge. All effort is usually to reduce the number of features without making the accuracy worse, which is a big challenge in selecting the features. This article has provided the NOCSA to select the least possible number of the features by changing the primary CSA while maintaining acceptable accuracy. At the beginning of the algorithm, Eq. (7) is used to generate the random population of the crows:

$$X_i = rand(1, Dim). \tag{7}$$

Table 6
Comparison of NOCSA and normal classification without feature selection task.

Dataset	NOCSA		Without feature selection	
	Accuracy	Fitness	Accuracy	Fitness
breast_cancer	0.9617	0.0593	0.6020	0.4582
breastEW	0.9590	0.0757	0.5380	0.5158
clean1	0.9632	0.0568	0.9314	0.1617
clean2	1.0000	0.0537	1.0000	0.1000
congressEW	0.7580	0.2255	0.6469	0.4178
exactly1	0.8321	0.1827	0.6587	0.4072
exactly2	0.9308	0.0928	0.7415	0.3326
HeartEW	0.8594	0.1712	0.5133	0.5380
ionosphere	0.8714	0.1600	0.6684	0.3985
Lymphography	0.9773	0.0589	0.4571	0.5886
m-of-n	1.0000	0.0493	0.5122	0.5390
semeion	0.8967	0.1366	0.9008	0.1893
sonarEW	0.8380	0.1659	0.5193	0.5327
SPECT	0.8091	0.2248	0.7940	0.2854
tic-tac-toe	0.9543	0.0611	0.6533	0.4120
vote	0.9794	0.0516	0.6300	0.4330
wine	0.9725	0.0655	0.6430	0.4213
zoo	0.9730	0.0594	0.5129	0.5384
Advertisements	0.8496	0.1829	0.7874	0.2913
CNAE	0.7398	0.2839	0.9972	0.1025
isolet5	0.9752	0.0677	0.0549	0.9506
Madelon	0.9972	0.0452	0.5040	0.5464
Parkinson's Disease	0.8685	0.1604	0.8153	0.2662
Average	**0.91157**	**0.11700**	**0.65573**	**0.40984**

According to this equation, for every crow, a logic array of 0 and 1 is created (0 or 1 means either the feature is selected or not). Dim is the total number of dataset features.

After generating the first population of the crows by using the fitness function, every crow's position is evaluated. In the proposed algorithm, the fitness function is as:

$$fitness = Acc + \alpha \times (1 - (\frac{L_s}{L_t})). \tag{8}$$

In this equation, Acc is the accuracy resulted from the KNN algorithm with K = 3. L_s and L_t represent the selected features and the total dataset features, respectively. α is the weight argument and provides a proportion between the two algorithm performance criteria (i.e., the accuracy and the number of selected features). Results show that the more the α is, the algorithm tends to select the fewer features, which can adversely affect the accuracy. According to the fitness equation, as the fitness's numerical value is high, one of the crow's positions is better than the others. In this article, α is selected at 0.2 experimentally. The pseudo-code of the proposed algorithm is given in Algorithm 3.

Algorithm 3 Novel Optimized Crow Search Algorithm (NOCSA)

fl, the maximum iterations(t_{max}), the number of the search agents(SA), DAP limit(DAP_lim)
Initialize the chaos vector (Eq. (5))
Binary initialization for the search agents (crows)
Calculate the fitness of each crow position
Sort the crows using the fitness values
Calculate the DAP for each crow (Eq. (4))
Initialize the crows memory
Set the current iteration. $t = 1$
while $t < t_{max}$ **do**
    for $i = 1 : 10$ **do**
        for $j = i + 1 : i + 10$ **do**
            if $DAP_i < DAP\_lim$ **then** ▷ exploitation (local search)
                generate a new crow (Eq. (6))
                discretization of the Crow positions(V2 transfer function)
            else ▷ exploration (global search)
                create a random subset with 20% features
                $new_{crow} = best_{crow}$ | random subset
                discretization of the Crow positions(V2 transfer function)
    Calculate the fitness of each new crow
    The Position updating (Eq. (3))
    use Pareto optimality to select the next population of crows
Report the best solution

5. Experimental results

Several algorithms from the relevant literature have been selected and classified into two groups for evaluating the proposed algorithm. The first group is dedicated to CSA-based methods that include ICSA1, ICSA2, CCSA, and ECSA. The second group is dedicated to some of the other meta-heuristic methods that include binary butterfly optimization approaches (BOA) (Arora & Anand, 2019), gravitational search algorithm (BGSA) (Rashedi, Nezamabadi-Pour, & Saryazdi, 2009), ABC, WFS-jDE (Fister, Fister, Jagrič, & Brest, 2019), PSO (Xue, Zhang, & Browne, 2012), sine cosine algorithm and DE (SCADE) (Abd Elaziz, Ewees, Oliva, Duan, & Xiong, 2017) and PSO and GWO (PSOGWO) (Singh & Singh, 2017). These methods have been selected because they have the best performance among all meta-heuristic algorithms.

5.1. Datasets

In this paper, 23 different datasets from the UCI data repository have been used. The datasets used in this paper are the same as those that the other ten methods (being compared to the proposed algorithm) have utilized in their related articles with the same instances and dimensions. The specifications of each dataset have been presented in Table 4. For determining the learning and testing sets, 5-fold cross-validation has been used.

5.2. Evaluation metrics

In this paper, three famous metrics have been employed to evaluate the proposed FS algorithm. These three metrics are as follows:

5.2.1. Average accuracy

This metric introduces the average accuracy of the classification overrunning the algorithm t_{Max} times and is defined in:

$$Average\ accuracy = \frac{1}{t_{max}} \sum_{i=1}^{t_{max}} Acc_i, \tag{9}$$

where Acc_i is the accuracy of ith iteration of the classifier.

5.2.2. Average fitness

Average fitness introduces the average of calculated fitness overrunning the algorithm t_{Max}:

$$Average\ fitness = \frac{1}{t_{max}} \sum_{i=1}^{t_{max}} Fit_i, \tag{10}$$

where Fit_i is the best fitness value of ith iteration, and the fitness is calculated as:

$$fitness = \beta \times Acc_{Err} + (1 - \beta) \times \frac{L_s}{L_t}. \tag{11}$$

The reason for defining the fitness function is to compare the results of the proposed algorithm to the other methods. In Eq. (11) Acc_{Err} is the error in classification accuracy, and β represents the proportion factor between the accuracy and the selected features in the algorithm. β is selected at 0.9 experimentally. Eq. (11) shows that the lower fitness in the algorithm means that the less error in the accuracy and the number of selected features occur. After the algorithm is done, the optimized algorithm solution is compared to fitness.

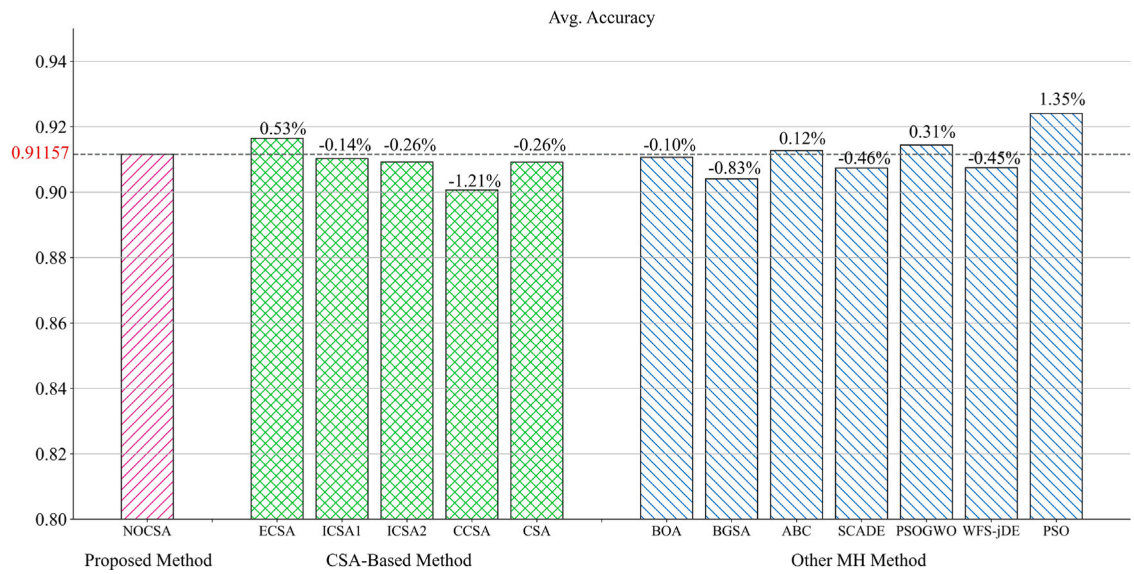

Fig. 8. Average of accuracy for all datasets.

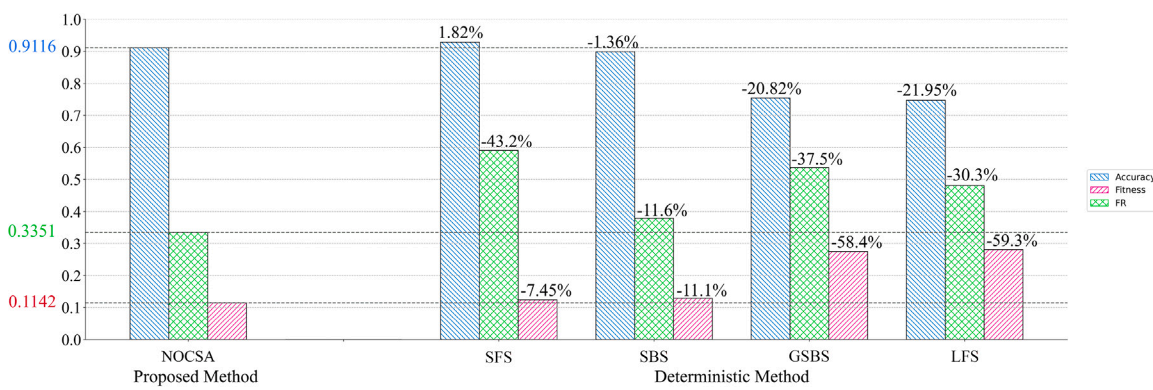

Fig. 9. Comparison of NOCSA and deterministic methods.

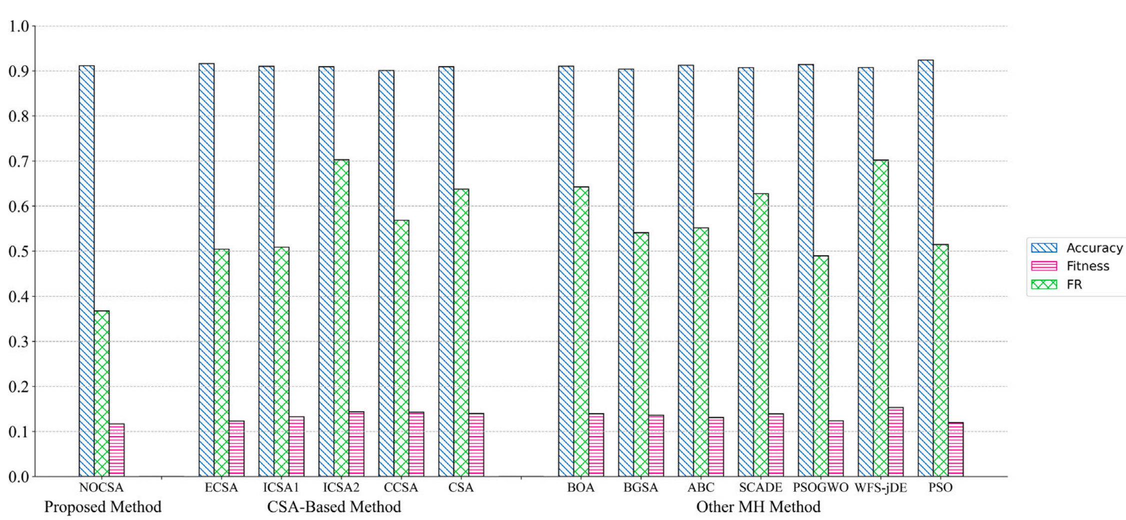

Fig. 10. Average of FR, fitness and accuracy and NOCSA improvement.

Table 7
Comparison of NOCSA and CSA based methods in terms of average and std of FR.

Datasets	Stat.measure	NOCSA	CSA based				
			ECSA	ICSA1	ICSA2	CCSA	CSA
Breastcancer	mean	0.2286	0.5556	0.5667	0.7500	0.6111	0.6278
	Std	0.0426	0.0806	0.1190	0.1074	0.1672	0.1409
BreastEW	mean	0.3590	0.4633	0.4533	0.4783	0.4800	0.6033
	Std	0.0461	0.0865	0.0988	0.0776	0.0875	0.0830
CongressEW	mean	0.2000	0.3000	0.2813	0.5688	0.3563	0.4875
	Std	0.0472	0.0851	0.0745	0.1215	0.1113	0.0801
Exactly	mean	0.4659	0.4615	0.5192	0.5192	0.6692	0.5231
	Std	0.0374	0.0000	0.0491	0.0423	0.1173	0.0402
Exactly2	mean	0.0945	0.3154	0.1269	0.7654	0.2769	0.7500
	Std	0.0192	0.3338	0.1584	0.0172	0.1547	0.0342
HeartEW	mean	0.3121	0.5154	0.5192	0.7000	0.5269	0.6846
	Std	0.0336	0.1561	0.1270	0.0785	0.1350	0.1113
IonosphereEW	mean	0.2966	0.3882	0.3941	0.6309	0.4485	0.5118
	Std	0.0237	0.0790	0.0584	0.1082	0.0786	0.0543
Lymphography	mean	0.4429	0.4833	0.5194	0.7472	0.5444	0.6833
	Std	0.0630	0.0767	0.1008	0.0775	0.0997	0.0828
Clean1	mean	0.4503	0.5980	0.6021	0.7676	0.6505	0.7105
	Std	0.0464	0.0127	0.0693	0.0062	0.0815	0.0413
Clean2	mean	0.4010	0.5566	0.5607	0.7262	0.6091	0.6691
	Std	0.0209	0.0078	0.0055	0.0087	0.0067	0.0057
M-of-n	mean	0.4527	0.4615	0.5154	0.5538	0.6462	0.5385
	Std	0.0000	0.0000	0.0439	0.0316	0.1099	0.0353
SonarEW	mean	0.4362	0.5083	0.5300	0.7692	0.5000	0.6317
	Std	0.0443	0.0523	0.0591	0.0493	0.0471	0.0518
SpectEW	mean	0.2259	0.4659	0.5000	0.7386	0.5136	0.5591
	Std	0.0755	0.0779	0.1084	0.1102	0.1044	0.1124
Tic-tac-toe	mean	0.5238	0.7778	0.7611	0.7778	0.6833	0.6111
	Std	0.0010	0.0000	0.0544	0.0000	0.0544	0.0987
Vote	mean	0.1982	0.3031	0.2719	0.5750	0.4625	0.4313
	Std	0.0465	0.0913	0.0867	0.1539	0.1509	0.1232
WineEW	mean	0.3231	0.5808	0.5731	0.7346	0.5962	0.6654
	Std	0.0637	0.1465	0.1073	0.0682	0.0823	0.1350
Semion	mean	0.4017	0.4561	0.4561	0.4561	0.4561	0.4561
	Std	0.0306	0.0938	0.0276	0.0820	0.0467	0.0639
Zoo	mean	0.3518	0.3781	0.5000	0.7031	0.5875	0.6750
	Std	0.0549	0.1773	0.1298	0.0970	0.1240	0.1006
isolet5	mean	0.4759	0.6520	0.6567	0.8899	0.7234	0.8090
	Std	0.0201	0.0331	0.0358	0.0958	0.0975	0.0813
madelon	mean	0.4978	0.6173	0.6372	0.8960	0.7467	0.7965
	Std	0.0178	0.0949	0.0687	0.0847	0.0202	0.0232
advertisements	mean	0.4514	0.6320	0.6049	0.8577	0.7268	0.7810
	Std	0.0248	0.0142	0.0609	0.0124	0.0511	0.0820
CNAE	mean	0.4251	0.5016	0.6291	0.7524	0.6079	0.6887
	Std	0.0098	0.0030	0.0081	0.0036	0.0032	0.0059
Parkinson's Disease	mean	0.4391	0.6279	0.5269	0.8123	0.6543	0.7728
	Std	0.0177	0.0071	0.0282	0.0039	0.0527	0.0702
Avg. Reduction		**0.36755**	**0.50433**	**0.50893**	**0.70305**	**0.56858**	**0.63770**

5.2.3. Average feature selection ratio

Introduces the ratio between the number of selected features and the dimension overrunning the algorithm t_{Max} and is defined as:

$$FR = \frac{1}{t_{max}} \sum_{i=1}^{t_{max}} \frac{len(BS_i)}{L_{total}}. \quad (12)$$

In this equation t_{max} is the total algorithm iterations and len(BS_i) shows the size of best algorithm solution until step i. And L_{total} is the number of dataset features. FR represents the average number of the selected features during the algorithm, and as less the FR is, the algorithm has more ability to reduce the features of the dataset.

5.2.4. Wilcoxon test

The Wilcoxon signed rank sum test is an example of a non-parametric or distribution-free test (Derrac, García, Molina, & Herrera, 2011). As for the sign test, the Wilcoxon signed-rank sum test is used to test the null hypothesis that a distribution's median is equal to some value. It can be used in place of a one-sample t-test or a paired t-test. This paper uses algorithms output as paired data and then employs the signed-rank test to determine whether two dependent samples are selected from populations with the same distribution or not. To implement the Wilcoxon signed rank-sum test, the following seven main steps should be done:

- **Step 1.** Set median difference (M) equal to zero (State of the null hypothesis).

Table 8
Comparison of NOCSA and other methods in terms of average and std of FR.

Datasets	Stat.measure	NOCSA	Other methods						
			BOA	BGSA	ABC	SCADE	PSOGWO	WFS-jDE	PSO
Breastcancer	mean	0.2286	0.6330	0.6111	0.6222	0.6111	0.5722	0.6075	0.4357
	Std	0.0426	0.0961	0.0987	0.0981	0.1110	0.0828	0.0382	0.0901
BreastEW	mean	0.3590	0.6017	0.5033	0.5083	0.6100	0.4800	0.8567	0.4972
	Std	0.0461	0.1017	0.0809	0.0858	0.0765	0.0775	0.0628	0.0701
CongressEW	mean	0.2000	0.5344	0.3375	0.4094	0.4625	0.3000	0.8531	0.4400
	Std	0.0472	0.1191	0.1320	0.1173	0.0821	0.0851	0.0777	0.0940
Exactly	mean	0.4659	0.5577	0.5269	0.4615	0.5231	0.4615	0.6221	0.5469
	Std	0.0374	0.0423	0.0516	0.0000	0.0402	0.0000	0.0427	0.0326
Exactly2	mean	0.0945	0.7308	0.1962	0.4846	0.7192	0.0392	0.6154	0.3785
	Std	0.0192	0.1210	0.1941	0.3088	0.1544	0.3502	0.0145	0.0485
HeartEW	mean	0.3121	0.6808	0.6115	0.5923	0.6654	0.5500	0.4615	0.5231
	Std	0.0336	0.0718	0.0682	0.1200	0.1255	0.1037	0.0273	0.0688
IonosphereEW	mean	0.2966	0.5338	0.4574	0.4176	0.5206	0.3412	0.5962	0.3426
	Std	0.0237	0.0649	0.0544	0.0802	0.0782	0.0818	0.0454	0.0323
Lymphography	mean	0.4429	0.6094	0.5639	0.5583	0.6528	0.5083	0.5556	0.5578
	Std	0.0630	0.1005	0.1100	0.1293	0.1018	0.0924	0.0183	0.0157
Clean1	mean	0.4503	0.7488	0.6244	0.6413	0.6947	0.5822	0.7948	0.4902
	Std	0.0464	0.0063	0.0761	0.0659	0.0706	0.0436	0.0675	0.0177
Clean2	mean	0.4010	0.7074	0.5830	0.5999	0.6533	0.5408	0.7885	0.4944
	Std	0.0209	0.0074	0.0013	0.0036	0.0007	0.0046	0.0526	0.0348
M-of-n	mean	0.4527	0.5423	0.5538	0.4615	0.5231	0.4615	0.9154	0.5562
	Std	0.0000	0.0393	0.0535	0.0000	0.0402	0.0000	0.0398	0.0740
SonarEW	mean	0.4362	0.6065	0.5092	0.5333	0.6325	0.4842	0.6960	0.5244
	Std	0.0443	0.0641	0.0615	0.0741	0.0579	0.0611	0.0520	0.0276
SpectEW	mean	0.2259	0.5977	0.4705	0.4795	0.6045	0.4795	0.7477	0.5735
	Std	0.0755	0.0811	0.0947	0.0983	0.0645	0.1166	0.0605	0.1151
Tic-tac-toe	mean	0.5238	0.6111	0.5667	0.7778	0.7778	0.7611	0.6653	0.6656
	Std	0.0010	0.0987	0.0497	0.0000	0.0000	0.0544	0.0124	0.0487
Vote	mean	0.1982	0.4210	0.3594	0.3875	0.4500	0.2500	0.7477	0.6438
	Std	0.0465	0.1440	0.1280	0.1084	0.1276	0.0608	0.0210	0.1075
WineEW	mean	0.3231	0.6500	0.5500	0.5538	0.6077	0.4923	0.8375	0.4897
	Std	0.0637	0.1157	0.1067	0.1047	0.1342	0.0765	0.0489	0.0374
Semion	mean	0.4017	0.4561	0.4561	0.4561	0.4561	0.4561	0.8470	0.4680
	Std	0.0306	0.0926	0.0352	0.0727	0.0817	0.0027	0.0691	0.0233
Zoo	mean	0.3518	0.6688	0.6156	0.4906	0.6094	0.4938	0.8000	0.6875
	Std	0.0549	0.0931	0.1078	0.1480	0.1403	0.1868	0.0968	0.0149
isolet5	mean	0.4759	0.7948	0.6901	0.7139	0.7852	0.6282	0.5341	0.4996
	Std	0.0201	0.0882	0.0807	0.0692	0.0007	0.0618	0.0364	0.0127
madelon	mean	0.4978	0.8512	0.7467	0.6471	0.7965	0.6123	0.6825	0.5156
	Std	0.0178	0.0013	0.0126	0.0006	0.0847	0.0446	0.1673	0.0289
advertisements	mean	0.4514	0.8442	0.6275	0.6997	0.6997	0.5778	0.7315	0.5146
	Std	0.0248	0.0936	0.0686	0.0773	0.0717	0.0194	0.0800	0.0067
CNAE	mean	0.4251	0.6972	0.6802	0.6079	0.7227	0.5739	0.5070	0.4708
	Std	0.0098	0.0002	0.0014	0.0019	0.0062	0.0074	0.0280	0.0147
Parkinson's Disease	mean	0.4391	0.7026	0.6016	0.5884	0.6587	0.6147	0.6887	0.5191
	Std	0.0177	0.0907	0.0692	0.0478	0.0116	0.0014	0.0206	0.0032
Avg. Reduction		**0.36755**	**0.64266**	**0.54098**	**0.55185**	**0.62768**	**0.48960**	**0.70225**	**0.51455**

- **Step 2.** Calculate each paired difference:

$$\begin{cases} \Delta_i = |y_i - x_i| \\ S_i = Sign(y_i - x_i). \end{cases} \tag{13}$$

- **Step 3.** Order and rank the pairs according to Δ_i.
- **Step 4.** Calculate W^+, the sum of the ranks of positive $\Delta_i S_i$, and W^-, the sum of the ranks of negative $\Delta_i S_i$
  Note that $(W^+ + W^-)$ should be equal to $\frac{n(n+1)}{2}$, where n is the number of pairs of observations in the sample. The parameter W measures the difference in location between pairs of points. For instance, if $W = 0$, then samples are most similar.
- **Step 5.** Calculate μ_w and σ_w.

Note that for large samples ($n > 20$), W has approximately normal distribution and will be calculated by:

$$\begin{cases} \mu_w = 0 \\ \sigma_w = \sqrt{\frac{n(n+1)(n+2)}{6}}. \end{cases} \tag{14}$$

- **Step 6.** Calculate Z-value:

$$Z = \frac{W}{\sigma_w}. \tag{15}$$

- **Step 7.** Calculate P-value using the normal probability table to estimate the P-value (usually is referred to as Z-table).

Table 9
Comparison of NOCSA and CSA based methods in term of average of fitness.

Datasets	NOCSA	CSA based				
		ECSA	ICSA1	ICSA2	CCSA	CSA
Breastcancer	0.0593	0.0811	0.0848	0.1018	0.0882	0.0898
BreastEW	0.0757	0.0845	0.0884	0.0918	0.0894	0.1050
CongressEW	0.0568	0.0603	0.0613	0.0968	0.0698	0.0866
Exactly	0.0537	0.0462	0.0533	0.0527	0.1706	0.0530
Exactly2	0.2255	0.2413	0.2294	0.2821	0.2519	0.2813
HeartEW	0.1827	0.2049	0.2167	0.2307	0.2127	0.2334
IonosphereEW	0.0928	0.1007	0.1099	0.1445	0.1171	0.1259
Lymphography	0.1712	0.1697	0.1912	0.2088	0.1921	0.1991
Clean1	0.1600	0.1786	0.1786	0.1786	0.1786	0.1786
Clean2	0.0589	0.0827	0.0892	0.1051	0.1031	0.1005
M-of-n	0.0493	0.0462	0.0518	0.0563	0.1177	0.0543
SonarEW	0.1366	0.1174	0.1402	0.1594	0.1352	0.1426
SpectEW	0.1659	0.1847	0.1975	0.2251	0.1978	0.2114
Tic-tac-toe	0.2248	0.2196	0.2319	0.2257	0.2374	0.2280
Vote	0.0611	0.0660	0.0686	0.1063	0.0860	0.1027
WineEW	0.0516	0.0719	0.0752	0.0909	0.0774	0.0825
Semion	0.0655	0.0714	0.0714	0.0714	0.0714	0.0714
Zoo	0.0594	0.0534	0.0651	0.0792	0.0734	0.0737
isolet5	0.1829	0.1932	0.2177	0.2140	0.2103	0.2067
madelon	0.2839	0.2886	0.3435	0.3038	0.3123	0.3151
advertisements	0.0677	0.0670	0.0684	0.0676	0.0738	0.0704
CNAE	0.0452	0.0411	0.0457	0.0493	0.0542	0.0457
Parkinson's Disease	0.1604	0.1668	0.1780	0.1668	0.1636	0.1606
Avg. Fitness	**0.11700**	**0.12336**	**0.13295**	**0.14386**	**0.14279**	**0.13992**

Table 10
Comparison of NOCSA and other methods in term of average of fitness.

Datasets	NOCSA	Other method						
		BOA	BGSA	ABC	SCADE	PSOGWO	WFS-jDE	PSO
Breastcancer	0.0593	0.0907	0.0893	0.0879	0.0888	0.0835	0.1033	0.0728
BreastEW	0.0757	0.1050	0.0958	0.0943	0.1044	0.0879	0.1187	0.0660
CongressEW	0.0568	0.0925	0.0761	0.0742	0.0837	0.0609	0.0970	0.0790
Exactly	0.0537	0.0595	0.0554	0.0462	0.0527	0.0462	0.1646	0.0613
Exactly2	0.2255	0.2821	0.2385	0.2646	0.2892	0.2156	0.2685	0.2466
HeartEW	0.1827	0.2332	0.2443	0.2142	0.2294	0.2137	0.1795	0.1576
IonosphereEW	0.0928	0.1311	0.1263	0.1083	0.1279	0.0984	0.1515	0.0799
Lymphography	0.1712	0.1920	0.2044	0.1842	0.1997	0.1797	0.1176	0.1557
Clean1	0.1600	0.1786	0.1786	0.1786	0.1786	0.1786	0.1792	0.0976
Clean2	0.0589	0.0944	0.0972	0.0907	0.0921	0.0836	0.0844	0.0644
M-of-n	0.0493	0.0545	0.0576	0.0462	0.0525	0.0462	0.1476	0.0633
SonarEW	0.1366	0.1409	0.1463	0.1396	0.1422	0.1263	0.1755	0.0820
SpectEW	0.1659	0.2164	0.2104	0.1980	0.2125	0.1905	0.1157	0.1524
Tic-tac-toe	0.2248	0.2282	0.2261	0.2186	0.2275	0.2254	0.2646	0.2397
Vote	0.0611	0.0851	0.0817	0.0804	0.0905	0.0630	0.1051	0.0916
WineEW	0.0516	0.0822	0.0798	0.0716	0.0795	0.0652	0.0854	0.0601
Semion	0.0655	0.0714	0.0714	0.0714	0.0714	0.0714	0.0975	0.0632
Zoo	0.0594	0.0744	0.0780	0.0637	0.0744	0.0628	0.1160	0.1135
isolet5	0.1829	0.2048	0.1845	0.1994	0.2012	0.1792	0.1995	0.1789
madelon	0.2839	0.3095	0.2833	0.2953	0.3236	0.2896	0.3032	0.2549
advertisements	0.0677	0.0711	0.0765	0.0684	0.0724	0.0670	0.1179	0.0656
CNAE	0.0452	0.0447	0.0506	0.0488	0.0475	0.0416	0.0549	0.0512
Parkinson's Disease	0.1604	0.1668	0.1813	0.1780	0.1556	0.1748	0.2835	0.2585
Avg. Fitness	**0.11700**	**0.13952**	**0.13624**	**0.13141**	**0.13900**	**0.12395**	**0.15351**	**0.11983**

5.3. Experimental analysis

All algorithms used in this work are randomly initialized. Additionally, all statistical results are recorded over 20 independent runs. The α parameter, which determines the weight of the classification accuracy in the fitness equation, is set to 0.9. These values are used in most papers in the relevant literature that deals with the FS problem using metaheuristics. Some of the algorithms mentioned above need particular parameters. These parameters' numerical values are given in 5, which they are set according to their corresponding papers.

5.4. NOCSA overall performance

The main goals of FS are two things, one is to increase the classification accuracy, and the second is to reduce the computation cost. Due to the increasing volume and complexity of data, FS is essential for the data mining task. To better understand this issue in Table 6, the classification accuracy and Fitness are compared with all dataset without using FS and when using NOCSA. It is pretty clear that the use of NOCSA is very effective in the data mining process.

5.5. Feature reduction (FR) metric

In this part, the proposed algorithm has been compared to the other FS algorithms in terms of feature reduction of the datasets. The comparison between different methods is presented in two Tables 7 and 8, one for CSA methods and the other for meta-heuristic. Finally, Fig. 6 shows a summary of the both algorithm classes' performance. In these tables, mean and std are the average and standard deviation of the FR over 20 runs of the algorithms, respectively. The proposed

Table 11
Comparison of NOCSA and CSA based methods in terms of average and std of accuracy.

Datasets	Stat.measure	NOCSA	CSA based				
			ECSA	ICSA1	ICSA2	CCSA	CSA
Breastcancer	mean	0.9617	0.9716	0.9687	0.9702	0.9699	0.9700
	Std	0.0114	0.0019	0.0019	0.0019	0.0021	0.0020
BreastEW	mean	0.9590	0.9576	0.9522	0.9512	0.9540	0.9504
	Std	0.0076	0.0035	0.0026	0.0030	0.0029	0.0041
CongressEW	mean	0.9632	0.9663	0.9631	0.9556	0.9620	0.9579
	Std	0.0266	0.0043	0.0035	0.0033	0.0042	0.0029
Exactly	mean	1.0000	1.0000	0.9985	0.9991	0.8848	0.9992
	Std	0.0000	0.0000	0.0046	0.0027	0.1055	0.0029
Exactly2	mean	0.7580	0.7669	0.7592	0.7716	0.7509	0.7708
	Std	0.0019	0.0129	0.0043	0.0082	0.0105	0.0144
HeartEW	mean	0.8321	0.8296	0.8169	0.8215	0.8222	0.8167
	Std	0.0213	0.0101	0.0084	0.0094	0.0106	0.0061
IonosphereEW	mean	0.9308	0.9312	0.9217	0.9095	0.9197	0.9170
	Std	0.0041	0.0069	0.0040	0.0051	0.0066	0.0054
Lymphography	mean	0.8594	0.8652	0.8453	0.8510	0.8470	0.8547
	Std	0.0158	0.0127	0.0090	0.0092	0.0110	0.0071
Clean1	mean	0.8714	0.8563	0.8563	0.8563	0.8563	0.8563
	std	0.0052	0.0044	0.0077	0.0096	0.0008	0.0039
Clean2	mean	0.9773	0.9701	0.9633	0.9640	0.9532	0.9628
	std	0.0016	0.0030	0.0060	0.0041	0.0033	0.0010
M-of-n	mean	1.0000	1.0000	0.9997	0.9990	0.9410	0.9995
	Std	0.0000	0.0000	0.0009	0.0017	0.0587	0.0017
SonarEW	mean	0.8967	0.9260	0.9031	0.9084	0.9053	0.9118
	Std	0.0094	0.0131	0.0063	0.0061	0.0113	0.0052
SpectEW	mean	0.8380	0.8466	0.8361	0.8320	0.8373	0.8272
	Std	0.0115	0.0111	0.0088	0.0056	0.0094	0.0044
Tic-tac-toe	mean	0.8091	0.8424	0.8269	0.8356	0.8122	0.8146
	Std	0.0249	0.0045	0.0085	0.0057	0.0166	0.0021
Vote	mean	0.9543	0.9603	0.9540	0.9458	0.9558	0.9338
	Std	0.0166	0.0054	0.0028	0.0052	0.0044	0.0045
WineEW	mean	0.9794	0.9846	0.9801	0.9806	0.9803	0.9823
	Std	0.0039	0.0040	0.0039	0.0039	0.0050	0.0049
Semion	mean	0.9725	0.9725	0.9725	0.9725	0.9725	0.9725
	std	0.0019	0.0040	0.0094	0.0035	0.0020	0.0034
Zoo	mean	0.9730	0.9827	0.9832	0.9901	0.9837	0.9931
	Std	0.0056	0.0044	0.0047	0.0045	0.0048	0.0047
Isolet5	mean	0.8496	0.8598	0.8530	0.8445	0.8443	0.8462
	Std	0.0072	0.0062	0.0088	0.0078	0.0083	0.0068
Madelon	mean	0.7398	0.7509	0.7472	0.7331	0.7375	0.7457
	Std	0.0106	0.0042	0.0051	0.0029	0.0064	0.0062
Advertisements	mean	0.9752	0.9859	0.9772	0.9762	0.9654	0.9654
	Std	0.0011	0.0001	0.0086	0.0060	0.0023	0.0041
CNAE	mean	0.9972	0.9982	0.9912	0.9783	0.9923	0.9959
	Std	0.0014	0.0048	0.0052	0.0061	0.0052	0.0037
Parkinson’s Disease	mean	0.8685	0.8529	0.8668	0.8659	0.8676	0.8673
	Std	0.0032	0.0088	0.0023	0.0062	0.0091	0.0028
Avg. Accuracy		**0.91157**	**0.91642**	**0.91027**	**0.90922**	**0.90066**	**0.90918**

algorithm improves FR significantly more than all the other algorithms in each dataset. It means that the proposed algorithm always selects the smallest subset of the features instead of the other optimizers. Fig. 6 shows the average of FR for each algorithm, ECSA has the best FR among CSA-based algorithms, and the proposed algorithm outperforms it by 27.12%. The proposed algorithm operates 24.92% better than PSOGWO, which has the best FR in the other meta-heuristic algorithms.

5.6. Fitness metric

The fitness metric is another evaluation metric to measure the performance of the FS algorithms. The fitness for all algorithms has been shown in Tables 9 and 10. From the CSA-based table, it can be seen that the proposed algorithm performed better than all CSA algorithms. Although, ECSA has the best performance among the other algorithms, which it is overcome by the proposed algorithm in the fitness index by 0.64%. The MH-methods table shows that the best fitness index belongs to PSO. The fitness index of the PSO is 0.11983 which the proposed algorithm was able to enhance this index to 0.11700. Fig. 7 represents the comparison among different methods and provides a summary of obtained results. This figure also shows that the proposed algorithm enhances fitness more than ECSA and PSO by 5.16% and 2.35%, respectively.

5.7. Accuracy metric

The accuracy metric determines if the selected features are good candidates for all the features. Tables 11 and 12 represent the accuracy

Table 12
Comparison of NOCSA and other methods in terms of average and std of accuracy.

Datasets	Stat.measure	NOCSA	Other method						
			BOA	BGSA	ABC	SCADE	PSOGWO	WFS-jDE	PSO
Breastcancer	mean	0.9617	0.9696	0.9687	0.9715	0.9692	0.9708	0.9527	0.9675
	Std	0.0114	0.0016	0.0020	0.0018	0.0012	0.0014	0.0578	0.0047
BreastEW	mean	0.9590	0.9502	0.9495	0.9517	0.9518	0.9557	0.9633	0.9819
	Std	0.0076	0.0033	0.0031	0.0033	0.0028	0.0028	0.0083	0.0101
CongressEW	mean	0.9632	0.9566	0.9529	0.9631	0.9584	0.9657	0.9871	0.9611
	Std	0.0266	0.0036	0.0041	0.0044	0.0029	0.0033	0.0059	0.0067
Exactly	mean	1.0000	0.9959	0.9970	1.0000	0.9996	1.0000	0.8863	0.9927
	Std	0.0000	0.0066	0.0065	0.0000	0.0020	0.0000	0.0236	0.0295
Exactly2	mean	0.7580	0.7678	0.7568	0.7598	0.7586	0.7648	0.7700	0.7681
	Std	0.0019	0.0178	0.0071	0.0173	0.0120	0.0093	0.0022	0.0065
HeartEW	mean	0.8321	0.8165	0.7965	0.8278	0.8191	0.8237	0.8519	0.8830
	Std	0.0213	0.0058	0.0104	0.0084	0.0062	0.0063	0.0482	0.0331
IonosphereEW	mean	0.9308	0.9137	0.9105	0.9261	0.9157	0.9286	0.8979	0.9493
	Std	0.0041	0.0032	0.0081	0.0083	0.0053	0.0048	0.0080	0.0155
Lymphography	mean	0.8594	0.8544	0.8355	0.8574	0.8507	0.8568	0.9310	0.8890
	Std	0.0158	0.0081	0.0110	0.0105	0.0103	0.0092	0.0042	0.0343
Clean1	mean	0.8714	0.8709	0.8563	0.8563	0.9040	0.8563	0.8892	0.9460
	std	0.0052	0.0070	0.0098	0.0067	0.0049	0.0004	0.0129	0.0103
Clean2	mean	0.9773	0.9474	0.9569	0.9660	0.9350	0.9673	0.9938	0.9833
	std	0.0016	0.0017	0.0082	0.0014	0.0097	0.0068	0.0026	0.0028
M-of-n	mean	1.0000	0.9997	0.9975	1.0000	0.9998	1.0000	0.9377	0.9915
	Std	0.0000	0.0008	0.0056	0.0000	0.0009	0.0000	0.0111	0.0416
SonarEW	mean	0.8967	0.9108	0.8940	0.9041	0.9123	0.9135	0.8823	0.9672
	Std	0.0094	0.0050	0.0099	0.0107	0.0062	0.0080	0.0507	0.0129
SpectEW	mean	0.8380	0.8260	0.8185	0.8333	0.8311	0.8416	0.9545	0.8943
	Std	0.0115	0.0048	0.0072	0.0094	0.0066	0.0070	0.0258	0.0089
Tic-tac-toe	mean	0.8091	0.8143	0.8118	0.8435	0.8337	0.8341	0.7800	0.8076
	Std	0.0249	0.0024	0.0031	0.0044	0.0061	0.0045	0.0012	0.0102
Vote	mean	0.9543	0.9522	0.9492	0.9537	0.9495	0.9578	0.9663	0.9697
	Std	0.0166	0.0033	0.0039	0.0037	0.0049	0.0039	0.0037	0.0685
WineEW	mean	0.9794	0.9809	0.9725	0.9820	0.9792	0.9823	0.9982	0.9876
	Std	0.0039	0.0038	0.0065	0.0039	0.0037	0.0027	0.0105	0.0161
Semion	mean	0.9725	0.9787	0.9725	0.9725	0.9725	0.9725	0.9858	0.9818
	std	0.0019	0.0003	0.0016	0.0074	0.0036	0.0043	0.0038	0.0024
Zoo	mean	0.9730	0.9916	0.9817	0.9837	0.9851	0.9851	0.9600	0.9503
	Std	0.0056	0.0036	0.0048	0.0048	0.0051	0.0051	0.0516	0.0025
Isolet5	mean	0.8496	0.8538	0.8477	0.8555	0.8513	0.8581	0.8377	0.8567
	Std	0.0072	0.0676	0.0520	0.0405	0.0656	0.0229	0.0294	0.0122
Madelon	mean	0.7398	0.7465	0.7391	0.7479	0.7398	0.7450	0.7389	0.7741
	Std	0.0106	0.0175	0.0201	0.0942	0.0366	0.0987	0.0328	0.0193
Advertisements	mean	0.9752	0.9820	0.9772	0.9772	0.9742	0.9742	0.9503	0.9843
	Std	0.0011	0.0052	0.0094	0.0046	0.0074	0.0015	0.0031	0.0023
CNAE	mean	0.9972	0.9882	0.9922	0.9892	0.9972	0.9962	0.9954	0.9954
	Std	0.0014	0.0074	0.0033	0.0037	0.0078	0.0064	0.0043	0.0013
Parkinson's Disease	mean	0.8685	0.8772	0.8589	0.8694	0.7816	0.8807	0.7615	0.7705
	Std	0.0032	0.0031	0.0015	0.0096	0.0036	0.0057	0.0007	0.0088
Avg. Accuracy		**0.91157**	**0.91065**	**0.90406**	**0.91268**	**0.90737**	**0.91438**	**0.90746**	**0.92403**

of all algorithms for each dataset. The superiors in CSA-based and the other meta-heuristic algorithms are ECSA and PSO by 0.91642 and 0.92403, respectively. The accuracy of the proposed method is 0.91157 and has negligible differences with the best counterparts. The average accuracy for each algorithm is represented in Fig. 8. The proposed algorithm is just 0.53% and 1.35% less accurate than ECSA and PSO, respectively.

5.8. Comparison with deterministic methods

In the next step, the proposed algorithm is compared with deterministic feature selection algorithms. The main difference among these algorithms and metaheuristic methods is that in different runs the algorithm on the same dataset, always a unique subset is returned as the output of the algorithm. Obviously, in this unique subset the accuracy, FR, and Fitness are constant. This paper uses four algorithms i.e SFS, SBS, GSBS and LFS as deterministic algorithms to compare their performance with NOCSA algorithm. Deterministic algorithms suffer from a major problem. These algorithms will have a huge run time for datasets with a feature number of above 100, which is also mentioned in the articles (Gheyas & Smith, 2010; Xue, Zhang, & Browne, 2014). As shown in Table 13, the deterministic algorithms were executed on 16 of the datasets used in the article because, they were either unusable or took a very long time to execute in other datasets. Meanwhile, the

Table 13
Comparison of NOCSA and deterministic approaches.

Dataset	Metric	NOCSA	SFS	SBS	GSBS	LFS
Breastcancer	Accuracy	0.9617	0.9594	0.9771	0.7505	0.7222
	FR	0.2286	0.4000	0.4000	0.7685	0.2806
	Fitness	0.0593	0.0765	0.0606	0.3014	0.2781
BreastEW	Accuracy	0.9590	0.9654	0.9701	0.8289	0.8812
	FR	0.3590	0.9000	0.4667	0.4118	0.8248
	Fitness	0.0757	0.1211	0.0736	0.1951	0.1894
CongressEW	Accuracy	0.9632	0.9747	0.9747	0.7653	0.8813
	FR	0.2000	0.4375	0.4375	0.4148	0.3927
	Fitness	0.0568	0.0665	0.0665	0.2527	0.1461
Exactly	Accuracy	1.0000	0.9780	0.7150	0.7445	0.5886
	FR	0.4659	0.6154	0.4615	1.4319	0.4990
	Fitness	0.0537	0.0813	0.3027	0.3731	0.4202
Exactly2	Accuracy	0.7580	0.7970	0.7580	0.6088	0.6204
	FR	0.0945	0.8462	0.0769	0.0865	0.6176
	Fitness	0.2255	0.2673	0.2255	0.3607	0.4034
HeartEW	Accuracy	0.8321	0.8889	0.8815	0.6748	0.7653
	FR	0.3121	0.6923	0.4615	0.6154	0.4615
	Fitness	0.1827	0.1692	0.1528	0.3542	0.2574
IonosphereEW	Accuracy	0.9308	0.9416	0.9516	0.7810	0.8667
	FR	0.2966	0.1765	0.1765	0.8824	0.1176
	Fitness	0.0928	0.0702	0.0612	0.2853	0.1317
Lymphography	Accuracy	0.8594	0.8722	0.8919	0.6472	0.8182
	FR	0.4429	0.7778	0.3889	0.3024	0.5029
	Fitness	0.1712	0.1928	0.1362	0.3478	0.2139
Clean1	Accuracy	0.8714	0.9732	0.9748	0.8090	0.7211
	FR	0.4503	0.5689	0.2515	0.3650	0.6881
	Fitness	0.1600	0.0810	0.0478	0.2084	0.3198
M-of-n	Accuracy	1.0000	1.0000	0.8720	0.8184	0.6771
	FR	0.4527	0.5385	0.4615	0.4126	0.4824
	Fitness	0.0493	0.0538	0.1614	0.2047	0.3389
SonarEW	Accuracy	0.8967	0.9077	0.9808	0.6825	0.7778
	FR	0.4362	0.4833	0.4833	0.8000	0.0500
	Fitness	0.1366	0.1314	0.0656	0.3658	0.2050
SpectEW	Accuracy	0.8380	0.8764	0.8464	0.7996	0.6317
	FR	0.2259	0.9091	0.4091	0.4423	0.7980
	Fitness	0.1659	0.2021	0.1791	0.2246	0.4113
Tic-tac-toe	Accuracy	0.8091	0.8520	0.7255	0.7554	0.7734
	FR	0.5238	0.8889	0.4444	0.3484	0.6750
	Fitness	0.2248	0.2220	0.2915	0.2550	0.2715
Vote	Accuracy	0.9543	0.9333	0.9633	0.7535	0.7938
	FR	0.1982	0.1875	0.1875	0.2747	0.3002
	Fitness	0.0611	0.0788	0.0518	0.2493	0.2156
WineEW	Accuracy	0.9794	0.9663	0.9663	0.8519	0.7407
	FR	0.3231	0.4615	0.4615	0.6154	0.5385
	Fitness	0.0516	0.0765	0.0765	0.1948	0.2872
Zoo	Accuracy	0.9730	0.9703	0.9406	0.8000	0.7005
	FR	0.3518	0.5625	0.5000	0.4118	0.4706
	Fitness	0.0594	0.0830	0.1035	0.2212	0.3166
Average	Accuracy	**0.9116**	**0.9285**	**0.8993**	**0.7545**	**0.7475**
	FR	**0.3351**	**0.5904**	**0.3793**	**0.5365**	**0.4812**
	Fitness	**0.1142**	**0.1234**	**0.1285**	**0.2746**	**0.2754**

SFS algorithm has the best classification accuracy with a classification accuracy of 0.9285, which is 1.82% better than NOCSA. However, the proposed algorithm performed better in the FR and Fitness metrics than the SBS and SFS algorithms by 11.6% and 7.45%, respectively. These algorithms have the best performance in these metrics, among deterministic algorithms. Fig. 9 shows a general comparison between the proposed algorithm and the deterministic algorithms.

5.9. Wilcoxon test

In this paper, a statistical non-parametric test called Wilcoxon has been utilized to investigate the proposed algorithm's efficiency (Derrac et al., 2011). The significance level of the test has been set equal to 0.05 (5%). Table 14 contains the P-values obtained from the comparison of the proposed algorithm with other algorithms. In this table, the less the P-value is than 0.05, the more influential the proposed algorithm will be than the other algorithms, and this is a piece of strong evidence to prove the algorithm's efficiency. As shown in Table 14, a tiny number of P-values are more significant than 0.05. Also, many table numbers are much smaller than 0.05, which indicates that the proposed algorithm has a high level of impact compared to the other algorithms.

5.10. Summary

Minimizing the features' size is the basis of the proposed algorithm to select a minimal subset among all features. Also, retaining other performance metrics has been considered as much as possible. Fig. 10 illustrates the average of each three evaluation metrics for each algorithm. Available algorithms in this figure have been classified into CSA-based and the other meta-heuristic optimizers. The proposed algorithm has fallen into the same range as the other best algorithms in terms of fitness and accuracy. One thing that sets the proposed algorithm apart from the other algorithms is the massive difference in the feature reduction metric. In this way, compared to the other FS algorithms, the proposed algorithm can select the smallest subset of the features while retaining the accuracy and the fitness within an acceptable range.

In addition to the advantages of the proposed algorithm, it should be noted that this algorithm cannot be pioneer among different algorithms in accuracy metric, and even in some cases, it offers lower accuracy than other competitors. Therefore, it is better to use this algorithm in cases where there is a need for significant FR. In cases where accuracy is the most important index and also the need for FR is not significant, other leading competitors should be used.

6. Conclusion and future work

In this study, a novel FS algorithm based on CSA has been introduced. The proposed algorithm uses DAP to balance exploitation (local search) and exploration (global search). A novel neighborhood assigning strategy and chaotic map have been employed to improve the local search. Furthermore, the new method has used a new approach according to each iteration's best results to make the global search purposeful. The experiments have been done over 23 UCI datasets and have been compared to CSA-based and other meta-heuristic algorithms. The observations reveal that the proposed algorithm is superior in the feature reduction metric and compresses the final selected features significantly alongside retaining an acceptable accuracy metric compared to the other best algorithm. Accordingly, the proposed algorithm is more applicable to engineering applications that have huge features. Also, by comparing the results of the algorithm on medical datasets (such as BreastCancer, BreastEW, HeartEW, Lymphography, SPECT, and Parkinson's disease), it can be seen that the proposed method has presented relatively good results compared to **other algorithms studied in the article**. Dimension Reduction also helps to better understand the underlying causes of the disease.

As future works, combining the ant colony algorithm and the proposed algorithm will be an attractive research study. Moreover, one can present a new initialization method to generate crows with better local search performance. Also, combining this algorithm with fuzzy logic can create a fuzzy-based nature-inspired algorithm that can have better results than this article because fuzzy logic generally has excellent results in optimization problems. We will also use dynamic parameter adaptation fuzzy algorithms to improve the algorithm's performance to obtain the optimal input parameters to run the algorithm and improve the classification results.

Table 14
Wilcoxon test results ($p \geq 0.05$ are in bold).

SRCSA	ECSA	ICSA1	ICSA2	CCSA	CSA	BOA	BGSA	ABC	SCADE	PSOGWO	WFS-jDE	PSO
Breastcancer	1.73E−04	3.38E−04	1.94E−03	8.93E−03	8.86E−05	3.20E−03	2.19E−04	1.63E−04	8.85E−05	1.63E−04	5.17E−04	**5.22E−02**
BreastEW	3.33E−02	1.94E−03	6.05E−03	6.42E−03	**8.03E−02**	1.89E−04	2.20E−03	**5.24E−02**	8.03E−03	**8.59E−02**	1.40E−04	2.98E−05
CongressEW	1.89E−04	2.51E−02	8.26E−05	**4.79E−01**	8.92E−04	1.03E−04	7.26E−03	4.66E−02	4.49E−04	1.41E−03	5.11E−03	1.94E−03
Exactly	**1.00E+04**	2.62E−03	2.28E−02	8.96E−05	3.04E−02	9.34E−03	8.92E−04	**1.00E+04**	**2.18E−01**	**1.00E+04**	6.81E−04	6.75E−05
Exactly2	1.89E−04	1.37E−02	9.03E−03	7.80E−04	**5.38E−02**	2.82E−03	**6.74E−02**	7.19E−03	1.83E−02	6.81E−04	3.84E−05	**1.08E−01**
HeartEW	1.71E−03	1.03E−04	1.94E−03	2.10E−04	6.02E−05	5.27E−05	8.09E−04	4.06E−02	3.90E−04	1.51E−03	**5.22E−02**	5.17E−04
IonosphereEW	**5.69E−02**	3.20E−04	8.83E−05	1.89E−04	6.83E−04	6.68E−05	8.62E−05	1.52E−02	1.03E−04	**1.76E−01**	8.96E−05	1.20E−04
Lymphography	1.51E−03	1.52E−02	2.76E−02	5.17E−04	**5.22E−02**	1.00E−02	1.93E−04	**1.00E+00**	6.92E−04	1.71E−02	1.94E−03	8.83E−05
Clean1	2.20E−03	4.55E−03	2.50E−02	3.17E−04	1.52E−02	8.84E−05	2.28E−02	1.03E−04	8.23E−05	1.43E−04	**6.74E−02**	1.40E−04
Clean2	5.57E−05	6.92E−03	8.85E−05	1.09E−04	2.12E−04	5.56E−05	4.79E−02	1.75E−05	8.81E−05	1.20E−04	8.79E−05	1.00E−02
M-of-n	**1.00E+04**	1.42E−03	5.17E−04	8.81E−05	1.44E−02	2.51E−02	2.04E−04	**1.00E+04**	1.00E−02	**1.00E+04**	4.49E−04	7.80E−04
SonarEW	2.82E−04	5.32E−05	1.36E−03	4.77E−05	2.54E−04	1.23E−05	1.89E−04	1.03E−04	8.26E−05	7.54E−05	1.11E−02	6.81E−04
SpectEW	5.17E−04	9.13E−03	5.73E−03	8.97E−03	1.83E−04	8.86E−05	8.50E−04	1.00E−02	4.89E−03	4.10E−02	9.90E−03	**2.79E−01**
Tic-tac-toe	8.88E−03	3.90E−04	5.50E−05	1.02E−03	6.42E−03	7.57E−05	6.81E−04	3.65E−05	5.85E−05	6.02E−05	7.80E−04	5.17E−04
Vote	8.49E−05	**6.74E−02**	**1.35E−01**	8.97E−03	6.47E−03	1.03E−02	9.90E−03	1.50E−03	5.73E−03	4.83E−04	8.26E−05	6.25E−05
WineEW	1.03E−04	6.42E−03	5.44E−04	7.19E−03	9.19E−04	1.20E−04	**1.00E−01**	5.17E−04	4.49E−04	6.81E−04	**6.74E−02**	5.11E−03
Semion	1.69E−05	7.11E−03	5.05E−05	8.79E−05	4.69E−04	8.92E−04	7.80E−04	6.25E−05	3.19E−03	1.63E−04	1.00E−02	1.03E−04
Zoo	1.83E−04	8.85E−05	8.92E−05	1.03E−04	7.20E−05	8.30E−03	5.93E−04	1.40E−04	1.03E−04	1.03E−04	4.49E−04	8.26E−05
Isolet5	1.03E−04	7.80E−04	7.50E−05	1.00E−02	3.59E−03	2.50E−03	**5.07E−02**	1.37E−02	3.90E−04	**5.69E−02**	1.03E−04	8.83E−05
Madelon	1.94E−03	1.40E−04	6.81E−04	3.65E−05	5.85E−05	6.02E−05	8.83E−05	1.03E−04	**1.00E+04**	2.28E−02	1.00E−02	2.76E−02
Advertisements	7.57E−05	6.81E−04	**6.74E−02**	3.90E−04	2.19E−04	1.03E−04	6.42E−03	5.17E−04	4.79E−02	2.76E−02	3.65E−05	2.50E−03
CNAE	1.03E−04	9.90E−03	**8.59E−02**	1.89E−04	2.20E−03	1.69E−05	5.11E−03	5.05E−05	2.54E−04	2.50E−03	1.20E−04	6.47E−03
Parkinson's Disease	1.20E−04	**1.00E−01**	1.19E−05	1.89E−04	1.03E−04	3.38E−04	1.94E−03	5.93E−04	8.86E−05	8.92E−04	5.17E−04	**5.07E−02**

CRediT authorship contribution statement

Behrouz Samieiyan: Conceptualization, Methodology, Resources, Writing – original draft, Project administration, Software. **Poorya MohammadiNasab:** Methodology, Conceptualization, Software, Validation, Resources, Data curation. **Mostafa Abbas Mollaei:** Methodology, Software, Formal analysis, Resources, Visualization. **Fahimeh Hajizadeh:** Methodology, Validation, Formal analysis, Writing – original draft, Visualization, Writing – review & editing. **Mohammadreza Kangavari:** Supervision, Writing – review & editing.

Declaration of competing interest

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

References

Abd Elaziz, M. E., Ewees, A. A., Oliva, D., Duan, P., & Xiong, S. (2017). A hybrid method of sine cosine algorithm and differential evolution for feature selection. In *International conference on neural information processing* (pp. 145–155). Springer.

Anter, A. M., & Ali, M. (2020). Feature selection strategy based on hybrid crow search optimization algorithm integrated with chaos theory and fuzzy c-means algorithm for medical diagnosis problems. *Soft Computing*, *24*(3), 1565–1584.

Arora, S., & Anand, P. (2019). Binary butterfly optimization approaches for feature selection. *Expert Systems with Applications*, *116*, 147–160.

Askarzadeh, A. (2016). A novel metaheuristic method for solving constrained engineering optimization problems: crow search algorithm. *Computers and Structures*, *169*, 1–12.

Banharnsakun, A., Achalakul, T., & Sirinaovakul, B. (2011). The best-so-far selection in artificial bee colony algorithm. *Applied Soft Computing*, *11*(2), 2888–2901.

Beheshti, Z. (2013). *Centripetal accelerated particle swarm optimization and its applications in machine learning* (Ph.D. thesis), Universiti Teknologi Malaysia.

Beyer, H.-G., & Schwefel, H.-P. (2002). Evolution strategies–a comprehensive introduction. *Natural Computing*, *1*(1), 3–52.

Bolón-Canedo, V., Sánchez-Marono, N., Alonso-Betanzos, A., Benítez, J. M., & Herrera, F. (2014). A review of microarray datasets and applied feature selection methods. *Information Sciences*, *282*, 111–135.

BoussaïD, I., Lepagnot, J., & Siarry, P. (2013). A survey on optimization metaheuristics. *Information Sciences*, *237*, 82–117.

Bui, D. T., Tsangaratos, P., Ngo, P.-T. T., Pham, T. D., & Pham, B. T. (2019). Flash flood susceptibility modeling using an optimized fuzzy rule based feature selection technique and tree based ensemble methods. *Science of the Total Environment*, *668*, 1038–1054.

Bulatović, R. R., Đorđević, S. R., & Đorđević, V. S. (2013). Cuckoo search algorithm: a metaheuristic approach to solving the problem of optimum synthesis of a six-bar double dwell linkage. *Mechanism and Machine Theory*, *61*, 1–13.

Castillo, O., & Amador-Angulo, L. (2018). A generalized type-2 fuzzy logic approach for dynamic parameter adaptation in bee colony optimization applied to fuzzy controller design. *Information Sciences*, *460*, 476–496.

Chen, Y., Li, L., Peng, H., Xiao, J., Yang, Y., & Shi, Y. (2017). Particle swarm optimizer with two differential mutation. *Applied Soft Computing*, *61*, 314–330.

Chen, J., Mi, J., & Lin, Y. (2020). A graph approach for fuzzy-rough feature selection. *Fuzzy Sets and Systems*, *391*, 96–116.

Crone, S. F., Lessmann, S., & Stahlbock, R. (2006). The impact of preprocessing on data mining: An evaluation of classifier sensitivity in direct marketing. *European Journal of Operational Research*, *173*(3), 781–800.

Derrac, J., García, S., Molina, D., & Herrera, F. (2011). A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms. *Swarm and Evolutionary Computation*, *1*(1), 3–18.

Dhiman, G., & Kumar, V. (2017). Spotted hyena optimizer: a novel bio-inspired based metaheuristic technique for engineering applications. *Advances in Engineering Software*, *114*, 48–70.

Díaz, P., Pérez-Cisneros, M., Cuevas, E., Avalos, O., Gálvez, J., Hinojosa, S., et al. (2018). An improved crow search algorithm applied to energy problems. *Energies*, *11*(3), 571.

Djellali, H., Djebbar, A., Zine, N. G., & Azizi, N. (2018). Hybrid artificial bees colony and particle swarm on feature selection. In *IFIP international conference on computational intelligence and its applications* (pp. 93–105). Springer.

Dy, J. G., & Brodley, C. E. (2004). Feature selection for unsupervised learning. *Journal of Machine Learning Research*, *5*(Aug), 845–889.

Dy, J. G., Brodley, C. E., Kak, A., Broderick, L. S., & Aisen, A. M. (2003). Unsupervised feature selection applied to content-based retrieval of lung images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *25*(3), 373–378.

Fister, D., Fister, I., Jagrič, T., & Brest, J. (2019). Wrapper-based feature selection using self-adaptive differential evolution. In *Swarm, evolutionary, and memetic computing and fuzzy and neural computing* (pp. 135–154). Springer.

Gandomi, A. H., & Alavi, A. H. (2012). Krill herd: a new bio-inspired optimization algorithm. *Communications in Nonlinear Science and Numerical Simulation*, *17*(12), 4831–4845.

Gheyas, I. A., & Smith, L. S. (2010). Feature subset selection in large dimensionality domains. *Pattern Recognition*, *43*(1), 5–13.

Goldberg, D. E., & Holland, J. H. (1988). Genetic algorithms and machine learning.

Gong, W., & Cai, Z. (2013). Differential evolution with ranking-based mutation operators. *IEEE Transactions on Cybernetics*, *43*(6), 2066–2081.

Jain, M., Singh, V., & Rani, A. (2019). A novel nature-inspired algorithm for optimization: Squirrel search algorithm. *Swarm and Evolutionary Computation*, *44*, 148–175.

Karaboga, D., & Basturk, B. (2007). A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm. *Journal of Global Optimization*, *39*(3), 459–471.

Karaboga, D., & Gorkemli, B. (2014). A quick artificial bee colony (qABC) algorithm and its performance on optimization problems. *Applied Soft Computing*, *23*, 227–238.

Kaveh, A., Vaez, S. R. H., Hosseini, P., & Fallah, N. (2016). Detection of damage in truss structures using simplified dolphin echolocation algorithm based on modal data. *Smart Structures and Systems*, *18*(5), 983–1004.

Kennedy, J., & Eberhart, R. (1995). Particle swarm optimization. In *Proceedings of ICNN'95-international conference on neural networks, Vol. 4* (pp. 1942–1948). IEEE.

Kim, Y., Street, W. N., & Menczer, F. (2002). Evolutionary model selection in unsupervised learning. *Intelligent Data Analysis*, *6*(6), 531–556.

Kohavi, R., John, G. H., et al. (1997). Wrappers for feature subset selection. *Artificial Intelligence*, *97*(1–2), 273–324.

Labani, M., Moradi, P., Ahmadizar, F., & Jalili, M. (2018). A novel multivariate filter method for feature selection in text classification problems. *Engineering Applications of Artificial Intelligence*, *70*, 25–37.

Lagunes, M. L., Castillo, O., Valdez, F., Soria, J., & Melin, P. (2021). A new approach for dynamic stochastic fractal search with fuzzy logic for parameter adaptation. *Fractal and Fractional*, *5*(2), 33.

Liang, J. J., Qin, A. K., Suganthan, P. N., & Baskar, S. (2006). Comprehensive learning particle swarm optimizer for global optimization of multimodal functions. *IEEE Transactions on Evolutionary Computation*, *10*(3), 281–295.

Liu, H., & Motoda, H. (1998). *Feature extraction, construction and selection: a data mining perspective, Vol. 453*. Springer Science & Business Media.

Liu, Q., Wu, L., Xiao, W., Wang, F., & Zhang, L. (2018). A novel hybrid bat algorithm for solving continuous optimization problems. *Applied Soft Computing*, *73*, 67–82.

Long, W., Jiao, J., Liang, X., & Tang, M. (2018). An exploration-enhanced grey wolf optimizer to solve high-dimensional numerical optimization. *Engineering Applications of Artificial Intelligence*, *68*, 63–80.

Maheshwari, R., Kumar, M., & Kumar, S. (2016). Optimization of feature selection in face recognition system using differential evolution and genetic algorithm. In *Proceedings of fifth international conference on soft computing for problem solving* (pp. 363–374). Springer.

Majhi, S. K., Sahoo, M., & Pradhan, R. (2019). Oppositional crow search algorithm with mutation operator for global optimization and application in designing FOPID controller. *Evolving Systems*, 1–26.

Meng, X.-B., Gao, X. Z., Lu, L., Liu, Y., & Zhang, H. (2016). A new bio-inspired optimisation algorithm: Bird swarm algorithm. *Journal of Experimental & Theoretical Artificial Intelligence*, *28*(4), 673–687.

Miramontes, I., Guzman, J. C., Melin, P., & Prado-Arechiga, G. (2018). Optimal design of interval type-2 fuzzy heart rate level classification systems using the bird swarm algorithm. *Algorithms*, *11*(12), 206.

Mirjalili, S., Gandomi, A. H., Mirjalili, S. Z., Saremi, S., Faris, H., & Mirjalili, S. M. (2017). Salp swarm algorithm: A bio-inspired optimizer for engineering design problems. *Advances in Engineering Software*, *114*, 163–191.

Mirjalili, S., & Lewis, A. (2013). S-shaped versus V-shaped transfer functions for binary particle swarm optimization. *Swarm and Evolutionary Computation*, *9*, 1–14.

Mirjalili, S., & Lewis, A. (2016). The whale optimization algorithm. *Advances in Engineering Software*, *95*, 51–67.

Mirjalili, S., Mirjalili, S. M., & Lewis, A. (2014). Grey wolf optimizer. *Advances in Engineering Software*, *69*, 46–61.

Ouadfel, S., & Abd Elaziz, M. (2020). Enhanced crow search algorithm for feature selection. *Expert Systems with Applications*, Article 113572.

Rashedi, E., Nezamabadi-Pour, H., & Saryazdi, S. (2009). GSA: a gravitational search algorithm. *Information Sciences*, *179*(13), 2232–2248.

Rodríguez, L., Castillo, O., Soria, J., Melin, P., Valdez, F., Gonzalez, C. I., et al. (2017). A fuzzy hierarchical operator in the grey wolf optimizer algorithm. *Applied Soft Computing*, *57*, 315–328.

Roeva, O., Zoteva, D., & Castillo, O. (2021). Joint set-up of parameters in genetic algorithms and the artificial bee colony algorithm: an approach for cultivation process modelling. *Soft Computing*, *25*(3), 2015–2038.

Sadeghian, Z., Akbari, E., & Nematzadeh, H. (2021). A hybrid feature selection method based on information theory and binary butterfly optimization algorithm. *Engineering Applications of Artificial Intelligence*, *97*, Article 104079.

Sánchez, D., Melin, P., & Castillo, O. (2021). Fuzzy dynamic parameter adaptation for particle swarm optimization of modular granular neural networks applied to time series prediction. In *Recent advances of hybrid intelligent systems based on soft computing* (pp. 189–204). Springer.

Sayed, G. I., Hassanien, A. E., & Azar, A. T. (2019). Feature selection via a novel chaotic crow search algorithm. *Neural Computing and Applications*, *31*(1), 171–188.

Shadravan, S., Naji, H., & Bardsiri, V. K. (2019). The sailfish optimizer: A novel nature-inspired metaheuristic algorithm for solving constrained engineering optimization problems. *Engineering Applications of Artificial Intelligence*, *80*, 20–34.

Shi, Z., Li, Q., Zhang, S., & Huang, X. (2017). Improved crow search algorithm with inertia weight factor and roulette wheel selection scheme. In *2017 10th international symposium on computational intelligence and design (ISCID), Vol. 1* (pp. 205–209). IEEE.

Singh, N., & Singh, S. (2017). Hybrid algorithm of particle swarm optimization and grey wolf optimizer for improving convergence performance. *Journal of Applied Mathematics*, *2017*.

Storn, R., & Price, K. (1997). Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization*, *11*(4), 341–359.

Tavazoei, M. S., & Haeri, M. (2007). Comparison of different one-dimensional maps as chaotic search pattern in chaos optimization algorithms. *Applied Mathematics and Computation*, *187*(2), 1076–1085.

Tsai, P.-W., Pan, J.-S., Liao, B.-Y., & Chu, S.-C. (2008). Interactive artificial bee colony (iabc) optimization. In *ISI2008, Vol. 12*.

Valdez, F., Castillo, O., Cortes-Antonio, P., & Melin, P. (2020). A survey of type-2 fuzzy logic controller design using nature inspired optimization. *Journal of Intelligent & Fuzzy Systems*, *39*(5), 6169–6179.

Wang, B., Jin, X., & Cheng, B. (2012). Lion pride optimizer: An optimization algorithm inspired by lion pride behavior. *Science China. Information Sciences*, *55*(10), 2369–2389.

Wolpert, D. H., & Macready, W. G. (1997). No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation*, *1*(1), 67–82.

Xue, B., Zhang, M., & Browne, W. N. (2012). Particle swarm optimization for feature selection in classification: A multi-objective approach. *IEEE Transactions on Cybernetics*, *43*(6), 1656–1671.

Xue, B., Zhang, M., & Browne, W. N. (2014). Particle swarm optimisation for feature selection in classification: Novel initialisation and updating mechanisms. *Applied Soft Computing*, *18*, 261–276.

Xue, B., Zhang, M., Browne, W. N., & Yao, X. (2015). A survey on evolutionary computation approaches to feature selection. *IEEE Transactions on Evolutionary Computation*, *20*(4), 606–626.

Yang, X.-S., & Gandomi, A. H. (2012). Bat algorithm: a novel approach for global engineering optimization. *Engineering Computations*.

Zamani, H., Nadimi-Shahraki, M. H., & Gandomi, A. H. (2019). CCSA: Conscious neighborhood-based crow search algorithm for solving global optimization problems. *Applied Soft Computing*, *85*, Article 105583.

Zhang, L., Mistry, K., Lim, C. P., & Neoh, S. C. (2018). Feature selection using firefly optimization for classification and regression models. *Decision Support Systems*, *106*, 64–85.

Zhang, L., & Zhang, C. (2008). Hopf bifurcation analysis of some hyperchaotic systems with time-delay controllers. *Kybernetika*, *44*(1), 35–42.

Zhu, G., & Kwong, S. (2010). Gbest-guided artificial bee colony algorithm for numerical function optimization. *Applied Mathematics and Computation*, *217*(7), 3166–3173.