An hybrid particle swarm optimization with crow search algorithm for feature selection

Article · July 2021

DOI: 10.1016/j.mlwa.2021.100108

4 authors, including:

Mohammed Abdullahi
Ahmadu Bello University

Ibrahim Hayatu Hassan
Ahmadu Bello University

Machine Learning with Applications 6 (2021) 100108

Contents lists available at ScienceDirect

Machine Learning with Applications

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

An hybrid particle swarm optimization with crow search algorithm for feature selection

Abdulhameed Adamu [a,b], Mohammed Abdullahi [a,*], Sahalu Balarabe Junaidu [a], Ibrahim Hayatu Hassan [a]

[a] *Department of Computer Science, Ahmadu Bello University Zaria, Nigeria*
[b] *Department of Computer Science, College of Education Akwanga, Nigeria*

ARTICLE INFO

Keywords:
Crow search algorithm
Particle swarm optimization
Opposition based learning
Feature selection
Wrapper method

ABSTRACT

The recent advancements in science, engineering, and technology have facilitated huge generation of datasets. These huge datasets contain noisy, redundant, and irrelevant features which negatively affects the performance of classification techniques in machine learning and data mining process. Feature selection is a pre-processing stage for reducing the dimensionality of datasets by selecting the most important attributes while increasing the accuracy of classification at the same time. In this paper, we present a novel hybrid binary version of enhanced chaotic crow search and particle swarm optimization algorithm (ECCSPSOA) to solve feature selection problems. In the proposed ECCSPSOA, in order to navigate the feature space, we hybridized the enhanced version of the CSA algorithm which has a better search strategy and particle swarm optimization (PSO) which is capable of converging into the best global solution in the search field. We further embed opposition-based learning technique in the local search of the hybrid algorithm. The ECCSPSOA was compared using 15 datasets from the UCI repository with four well-known optimization algorithms, such as particle swarm optimization (PSO), binary particle swarm optimization (BPSO), crow search algorithm (CSA), and chaotic crow search algorithm (CCSA). In the experiments with k-Nearest Neighbour (KNN) as a classifier, six different performance metrics were used. To tackle the over-fitting problem, each dataset is divided into training and testing data using K-fold cross-validation. The computational findings demonstrate that the proposed algorithm obtains an average accuracy rate of 89.67 % over 15 datasets, indicating that our technique exceeds state-of-the-art findings in 12 of the 15 datasets studied. Furthermore, the suggested approach outperforms state-of-the-art methods in terms of fitness value and standard deviation, obtaining the lowest value in 13 and 8 of the datasets studied respectively.

1. Introduction

Usually, datasets for classifications problems contains noisy, redundant, and meaningless features, which affects the performance of the learning algorithm. Feature selection is a pre-processing step required to reduce the dimensionality of the datasets. Feature selection methods are profoundly used for selecting informative features and removing irrelevant/redundant features to reduce computational cost and improve prediction/classification accuracy (Gheyas & Smith, 2010). The purpose of the selection of features is therefore to minimize the size of the dataset, increase the accuracy of the prediction, and understand the dataset for different machine learning applications which includes clustering, classification, regression, and computer vision (Chandrashekar & Sahin, 2013). Algorithms for Feature Selection are classified into three: filter, wrapper, and hybrid approach. In the filter approach, feature subsets are selected and evaluated without involving a machine learning (Du et al., 2019; Li et al., 2020; Nagpal & Singh, 2018). In the case of the wrapper method, the quality of a given classifier is used as an evaluation criterion to estimate the level of relevance of the selected features (Dash & Luo, 1997; Figueiredo & Jain, 2002; Figueiredo et al., 2004; Roth & lange, 2004). The hybrid method, takes the advantage of the filter and wrapper techniques through combination of the two methods (Li et al., 2006; Solorio-Fernandez et al., 2016). The filter methods are less time consuming as compared to both wrapper and hybrid methods, since they do not involve a classifier in the feature selection process. However, the wrapper and hybrid approach mostly find better feature subsets which are representative of the original features (Shah et al., 2020).

Moreover, search strategy is a key issue in the selection of features. Several subset search techniques were suggested to find a sufficiently good subset such as exhaustive search, sequential search, and random

* Corresponding author.
E-mail addresses: abdulhameedadamu@gmail.com (A. Adamu), abdullahilwafu@abu.edu.ng (M. Abdullahi), sahalu@abu.edu.ng (S.B. Junaidu), ihhassan@abu.edu.ng (I.H. Hassan).

https://doi.org/10.1016/j.mlwa.2021.100108
Received 19 April 2021; Received in revised form 8 June 2021; Accepted 10 July 2021
Available online 15 July 2021

search, sequential forward selection (SFS), and sequential backward selection (SBS) (Dash & Luo, 1997). For instance, SFS and SBS methods starts with empty or complete set of features, and try to add or remove features from the set in incremental order to obtain optimal features. These methods can obtain optimal result when dealing with a relatively small subset. However, the methods overlooked the dependency factor between certain features, and they can as well get trapped in local optima because of their fixed search strategy. As the size data generated these days are always very huge in dimension, this increases the search space's complexity, where the maximum number of possible solutions is 2^n for n features dataset. As n grows larger, it becomes computationally impracticable to enumerate all the features. Hence, the need for design of efficient feature selection algorithms for large scale classification problems.

Several metaheuristic algorithms have been used for feature selection to overcome the drawbacks of earlier heuristic-based techniques. Some of the feature selection based metaheuristic algorithms includes Genetic algorithm, artificial bee colony (ABC) optimization (Kuo & Hong, 2018), Grey wolf optimizer (GWO) (Emary et al., 2016), Cuttlefish algorithm (CFA) (Eesa et al., 2015), Bat Algorithm (BA) (Rodrigues et al., 2014), Grasshopper optimization algorithm (GOA) (Al-Sharaf et al., 2019), intelligent water drops (IWD) (Askazadeh, 2016), Whale Optimization Algorithm (WOA) (Mafarja & Mirjalili, 2018), competitive swarm optimizer (CSO) (Gupta et al., 2018), differential evolution (DE) algorithm (Hancer et al., 2018), and Crow search algorithm (CSA) (Sayed et al., 2017). These algorithms begin with a number of solutions, and these solutions are iteratively evolved. The quality of the solutions are evaluated using fitness function, and the weaker solutions are eliminated when evolving new solutions. The metaheuristic-based feature selection techniques have ability to efficiently cope with large solution search space and also to produce near-optimal results in a reasonable amount of time because of their faster convergence rate when compared with heuristic based techniques. Further, they can easily escape from local optima.

CSA is a recently introduced metaheuristic algorithm that has an efficient search strategy algorithm, have found to have competitive performance with the established metaheuristic algorithms. CSA was inspired by the intelligent behaviour exhibited by crows in terms of hiding and thieving of their foods. Since the introduction of CSA, it has been broadly applied to solve various optimization challenges which includes engineering (Abdallah & Algamal, 2020; Rizk et al., 2020; Xiaoxia et al., 2020), medical (Gupta et al., 2018; Devikanniga et al., 2020; Fred et al., 2020), power energy (Abdurazaq & Muhammet, 2021; Bishwajit et al., 2020; Saha et al., 2017; Sasan et al., 2019), image processing (Balasaheb & P, 2018; Mobeen et al., 2020; Pankaj & Jitender, 2020; Upadhyaya & Chhabra, 2019), and feature selection (Abhilasha & Tirath, 2021; Ahmed & Mumtaz, 2020; Anter et al., 2019; Reddy & Gopal, 2021).

The presence of more numbers of irrelevant and redundant features can generate many local optima in the large solution space. These local optima points can greatly challenge the efficiency of CSA algorithms in reaching global optimum which could results to sub-optimal features subsets. Also, according to the theorem of no free lunch (Wolpert & Macready, 1997) that, there can be no ultimate optimization approach that is better globally than any other strategy. Therefore, this paper proposed an alternative wrapper-based feature selection method based on the hybridization of CSA and PSO. To address the issue of diversity, a chaotic map was used to replace the random component of CSA, and the enhanced CSA is hybridized with PSO named Enhanced Chaotic Crow Search Particle Swarm Optimization (ECCSPSOA) to improve the exploitation ability of CSA for faster convergence. Further, an Opposition Based Learning (OBL) which is a local search technique is employed to prevent the proposed ECCSPSOA algorithm from entrapment in local optima. Then, K-nearest neighbour (KNN) was used as the learning model to evaluate the effectiveness of the proposed feature selection. The main contributions of the paper are as follows:

- Chaotic based CSA algorithm to improve population diversity for optimal feature subset selection.
- Hybrid CSA algorithm for faster convergence while ensuring optimal feature subset selection.
- OBL local search technique for avoiding entrapment in local optima.
- Performance evaluation of the proposed algorithm against the compared algorithms.

The rest of this paper is structured as follows. Section 2 overviews the related feature selection techniques. Section 3 describes the CSA and PSO preliminaries. Details of the proposed algorithm are provided in Section 4. The detailed experiment used in the verification of the proposed algorithm is mentioned in Section 5. Section 6 presents the conclusion and suggestions for future works.

2. Related works

The wrapper-based feature selection techniques depend on the learning algorithms to assess the superiority of features selected based on two features: first, it looks for the optimal subset of features (based on the learning algorithm) and secondly access the quality of features selected by a model. In this technique, a model is trained with a new set of features every time based on the fitness value for which the computational cost of the problem increases as compared with the filter approaches. Then the model is tested based on the test set and the error is computed. The search strategy of the wrapper method may be sequential or random (Cai et al., 2018). The sequential search methods select (add/remove) the features sequentially but they might be trapped in a local optimum. The most commonly used sequential search methods are sequential backward search, sequential forward search, and floating search. However, the random search techniques applied the randomness in the search process to escape from the local optimal solution (Al-Ani et al., 2013). The most commonly used random search methods are GA, PSO, DE, and ACO.

The wrapper centred feature selection approaches are still misty among the research community as many researchers are endlessly putting their determination to come-up with new feature selection approaches based on several optimization approach. Sayed et al. (2017) proposed an improved Crow search algorithm (CSA) by enhancing it using chaotic theory for tackling feature selection problem. Kuo and Hong (2018) solved feature selection problems using artificial bee colony (ABC) optimization. Emary et al. (2016) improved Grey wolf optimizer (GWO) and then used it for selection of features. Eesa et al. (2015) implemented Cuttlefish algorithm (CFA) in selecting appropriate features in intrusion detection system. Additionally, Rodrigues et al. (2014) employed Bat Algorithm (BA) in selecting relevant attributes of datasets. Aljarah et al. (2018) make used of Grasshopper optimization algorithm (GOA) in feature selection issues. Acharya and Singh (2018) employed the service of intelligent water drops (IWD) in selecting most helpful attributes for IDS. Mafarja and Mirjalili (2018) enhanced Whale Optimization Algorithm (WOA) and solved feature selection problems. Gu and Cheng (2018) picked competitive swarm optimizer (CSO) and used it in tackling feature selection problems. Hancer et al. (2018) solved features selection problems while using differential evolution (DE) algorithm. A wrapper feature selection algorithm based on continues brain storm optimization (CBSOFS) for feature selection was proposed in the work of (Zhang et al., 2018). Experimental results show that CBSOFS achieves comparable results in comparison with other algorithms. Emary and Zawbaa (2019) proposed a feature selection method based on Lèvy flights ant–lion optimization (LALO) which is an extension of the ant–lion optimization algorithm (ALO). Kumar and Bharti (2019) suggested a Modified Binary Particle Swarm Optimization (BPSO) algorithm to select the optimal subset of features in a clustering problem. To solve the problem of feature selection, Al-Tashi et al. (2019) presented a binary version of the hybrid grey wolf optimization (GWO) and particle swarm optimization (PSO) termed as

BGWOPSO. The proposed technique gives the best optimal features, and the computational time in comparison with other well known algorithms. A binary Butterfly Optimization Algorithm (BOA) was proposed to select the optimum subset of features to handle a classification problems (Arora & Anand, 2019). The BOA shows a higher performance than other algorithms. Ghosh et al. (2019) proposed a hybrid wrapper-filter feature selection approach based on Ant Colony Optimization (ACO). A new binary whale optimization algorithm (BWOA) was presented by Tawhid and Ibrahim (2020) to handle feature selection issues based on wrapper approach. Analysis reveals that the BWOA can offer an efficient tool for obtaining a minimal subset of the features.

Additionally, de Souza et al. (2020) proposed a Binary coyote optimization algorithm (BCOA) for selection of optimum subset of features which is applicable in a classification problems. Abdel-Basset et al. (2020) suggested a novel Grey Wolf Optimizer algorithm combined with a two-phase mutation to handle the difficulties in selection of features for classification based on the wrapper technique. The proposed method shows better performance in comparison with particle Swarm optimization (PSO), Flower Algorithm (FA), Multi-verse optimizer Algorithm, Bat Algorithm (BA) and Whale Optimization Algorithm (WOA). Jaya optimization algorithm (FSJaya) based wrapper feature selection method was presented in the work of Das et al. (2020). The analysis reveals that the accuracy of classification of FSJaya on greater number of the datasets is higher than the compared methods like Feature selection using Particle Swam Optimization (FSPSO), Genetic Algorithm (FSGA), and Differential Evaluation (FSDE). Sahebi et al. (2020) proposed a general wrapper-based attribute selection method, refers to as GeFeS, which is centred on a parallel novel intelligent genetic algorithm (GA). Almasoudy et al. (2020) also proposed a wrapper-based attribute selection approach centred on differential evolution technique for intrusion detection systems. In the work of Gokalp et al. (2020), a new wrapper feature selection method centred on Iterated Greedy (IG) metaheuristic for sentiment analysis was also proposed. In this work, a selection process that is based on pre-calculated filter scores for the greedy construction portion of the IG algorithm was also developed. Guha et al. (2020) presented an embedded whale optimization algorithm (WOA) referred to as embedded chaotic whale survival algorithm (ECWSA), which uses wrapper process to realize better classification accuracy and a filter technique to further enhance the subset selected with low computation cost. Agrawal et al. (2020) suggested a novel wrapper feature selection approach based quantum Whale Optimization Algorithm (QWOA), which is a combination of Quantum Notions and the Whale Optimization Algorithm (WOA).

Moreover, Nouri-Moghaddam et al. (2021) proposed a binary Forest Optimization for multi-objective feature selection issues centred on grid and region-based selection for sustaining Pareto front named (MOFOA). The MOFOA outperforms compared algorithms. Amini and Hu (2021) presented a GA based two-layer feature selection technique to improve prediction accuracy. A new Hyper Learning Binary Dragonfly Algorithm (HLBDA) for optimum selection of features was proposed in Too and Mirjalili (2021a, 2021b) in order to address the problem of local optimum entrapment. Abdel-Basset et al. (2021) presented a new feature selection method centred on a hybrid Harris Hawks Optimization algorithm based on Bitwise operations and Simulated Annealing (HHOBSA). Chaudhuri and Sahu (2021) proposed a Binary Crow Search Algorithm with Time Varying Flight Length (BCSA-TVFL) for solving feature selection issues. Tubishat et al. (2021) developed a Dynamic Salp swarm algorithm (DSSA) for feature selection problem. In Too and Mirjalili (2021a, 2021b), a wrapper-based feature selection method for biological dataset called general learning equilibrium optimizer (GLEO) was proposed. The general learning strategy was used as local search technique to improve the exploitation capability of the equilibrium optimizer. Afarja et al. (2021) also proposed different transfer functions for Whale Optimization Algorithm (WOA). The proposed transfer functions belong to either S-shaped or V-shaped. The proposed V-shaped based transfer functions produced superior performance over the S-shaped approaches. Alweshah et al. (2021) presented a hybridized mine blast algorithm (MBA) with simulated annealing (SA), SA was used as a local search technique to improve the exploitation capacity of MBA. lbashish et al. (2021) also, presented a hybrid method based on Binary Biogeography Optimization (BBO) for feature selection issues.

However, the evolutionary computation techniques are still challenged by issues such as imbalance between local and global search, entrapment in local optima, and low solution diversity which results to sub-optimal features subsets (Sayed et al., 2017). Among optimization researchers, interest in hybrid algorithms has recently risen drastically (Talbi et al., 2012). Hybridization of one algorithm of evolutionary computation techniques with another involves combining them to exploit and combine the benefits of the individual pure strategies (Tawhid & Dsouza, 2018). It is primarily due to the theorem of no free lunch (Wolpert & Macready, 1997) that, the generalized understanding of the algorithm of evolutionary computation strategies changed and people realized that there can be no ultimate optimization approach that is better globally than any other strategy. In addition, it almost always needs a specialized algorithm to be compiled of adequate components in order to solve a problem at hand most successfully. Therefore, the goal of this paper work is to come up with a new hybridized enhanced version of CCSA with PSO algorithm in order to solve feature selection problems. The advantages of the CSA search strategy and its rapid convergence of the PSO algorithm are predicted to be exhibited by the proposed algorithm named Enhanced Chaotic Crow Search Particle Swarm Optimization (ECCSPSOA). Further, the OBL local search technique is employed to prevent the proposed ECCSPSOA algorithm from entrapment in local optima.

3. Proposed work

3.1. Crow search algorithm (CSA)

Crow search algorithm (CSA), suggested by Askazadeh (2016), is a nature-inspired algorithm. This population-based evolutionary computation techniques algorithm imitates the conduct and social interaction of crow birds. Crows are certainly smart birds with a big brain relative to their size living in a group (flock) and hide their food in the secret locations that can be memorized and retrieved even after a few months. Moreover, in the mirror test, they are self-conscious. They can recall looks, if an unfriendly one is encountered, in a complex manner of communication, they warn the other crows. Crow, much like other social creatures, may at some stage engage in robbery by carefully watching other Crow's food hiding place and then stealing their food. When a crow suspects like another one is trailing him, he moves to another location far from the hiding location of the food to fool a thief.

3.2. Framework of CSA

The population contains N solutions (number of crows) and the problem dimension is d. At iteration t, the location of each crow i is denoted by a vector $q_i^t = \left[q_{i1}^t, q_{i2}^t, \ldots, q_{id}^t\right]$ $for\ i = 1, 2, 3, \ldots, N$, where q_i^t is the possible location solution in dimension d for crow i.

If a crow says i want to steal from another crow j, in this scenario, two situations could occur:

1. Crow j does not watch the crow i after it, crow i will discover the food's store of crow j and update its position as given in Eq. (1).

$$q_i^{(t+1)} = q_i^{(t)} + r_i * fl_j^{(t)} * (m_j^{(t)} - q_i^{(t)}). \tag{1}$$

where fl indicates the flight length. r_i been a random number $\in [0, 1]$.

2. When crow j understands that crow i follow her to discover her food's hiding place. The crow j moves randomly to fool crow i in this situation.

The two cases can be combined mathematically as follow;

$$q_i^{(t+1)} = \begin{cases} q_i^{(t)} + r_i * fl_j^{(t)} * \left(m_j^{(t)} - q_i^{(t)}\right), r_j \geq AP_i^t \\ Choose\ a\ random\ position, \quad Otherwise \end{cases} \tag{2}$$

where r_i *and* r_j are a random number $\in [0, 1]$ and AP_i^t is the awareness probability of crow *j* at *t iteration*. The searching capability of crows is influences by the value of *fl*. *fl*'s high values contribute to global search while low values contribute to local search (Askazadeh, 2016).

Every crow is assessed during the running of the algorithm using a clearly defined fitness function. Then, according to the fitness value, the crows update their positions. Each new position is verified as feasible. The crows' memories are updated in accordance with Eq. (3)

$$m_i^{(t+1)} = \begin{cases} q_i^{(t+1)} & if\ f\left(q_i^{(t+1)}\right)\ is\ better\ than\ f\left(q_i^{(t)}\right) \\ m_i^{(t)}, & otherwise \end{cases} \tag{3}$$

Algorithm 1 illustrates the pseudo-code for the original CSA.

Algorithm 1: The normal CSA

Crows location $q_i^t (1 = 1,2, \dots m)$ *get randomly initialize*
The values of fl and AP are set
The fitness function $f\left(q_i^{(t)}\right)$ *for each crow* q_i^t *are evaluated*
The memory m_i^q *for each crow* q_i^t *is set*
Set t equal to zero, maximum number of iteration equal to max_iter and flock of crows equal to flc
while *(t less than max_iter)* **do**
  for *k = 1 to flc*
    choose one of the crows to trail Randomly
    Outline an awareness probability
    If *(rand* $\geq AP_k^t$*)*
      $q_k^{(t+1)} = q_k^{(t)} + rand_k * fl_k^{(t)} * (m_k^{(t)} - q_k^{(t)})$
    else
      $q_k^{(t+1)}$ *=get a position in the search field randomly*
    end
  end
  Evaluate the viability of the new position $q_k^{(t+1)}$
  The new position of the crow $f\left(q_i^{(t+1)}\right)$ *get evaluated*
  Then the memory of the crow $m_i^{(t+1)}$ *is updated*
end while
Produce the best solution

3.3. Binary CSA (BCSA)

In BCSA, a new time-varying transformation function was introduced to increase the exploration and exploitation potential of the initial CSA (Sayed et al., 2018). The solution set is transformed to a binary form in which the solution is confined to a binary form of {0, 1}. The transformation of agents form continuous to binary space is achieved using Eq. (4).

$$q_i^{(t+1)} = \begin{cases} 1 & if\ (s(q_i^{(t+1)})) \geq rand() \\ 0 & Otherwise \end{cases} \tag{4}$$

$$\text{where } s = \frac{1}{1 + e^{10(q_i^{(t+1)} - 0.5)}} \tag{5}$$

3.4. Chaotic crow search algorithm (CCSA)

Chaos has been described as a phenomenon. Any change in its original state can result in a non-linear change in future actions. In CCSA, the random variables used to change the crow location are substituted by chaotic variables (Sayed et al., 2018; Thom et al., 2018). As the Crow location update determines the optimal solution and the convergence rate, the chaotic sequence created from the chaotic maps are used. The CSA method, coupled with chaotic sequences, is represented in Eq. (6).

$$q_i^{(t+1)} = \begin{cases} q_i^{(t)} + C_j * fl_j^{(t)} * \left(m_j^{(t)} - q_i^{(t)}\right), C_z \geq AP_i^t \\ Choose\ a\ random\ position, \quad Otherwise \end{cases} \tag{6}$$

where *C(j)* stands for value gotten from chaotic map at jth iteration and *C(z)* the value gotten from the chaotic map at zth iteration.

3.5. Particle swarm optimization

Established by Kennedy and Ebenhart (1995) Particle Swarm Optimization (PSO) receives a tremendous amount of research interest in the feature selections based on the social behaviours associated with bird flocking to solve optimization problems. PSO belongs to swarm intelligence optimization proved to be less expensive computationally and converge faster. In PSO, every solution can be interpreted as a particle in a swarm where each particle has a certain velocity and position. According to experience and that of neighbours, each particle updates its position and velocity. The best previous position of the particle is termed as personal best while the best position achieved by the population of particles is known as global best. As a result, velocity and position of each particle are modified to find an optimal solution in accordance to the personal best and global best solution. Eventually, when a predefined criterion such as the best fitness value or the highest value of iterations is met, the algorithm halts.

3.6. Framework of PSO

Within the PSO algorithm, N particle positions and velocities in a space of d dimensions give solutions initialized in a random situation. The solution of particle i in iteration t is as defined in Eq. (7). The present solution for each particle is then upgraded for the local and global optima, calculated by means of Eqs. (8) and (10), accordingly.

$$Q_i^t = \left[q_{i,1}^t, q_{1,j}^t, \dots, q_{i,d}^t\right] \qquad i - 1, 2, \dots .N \tag{7}$$

$$V_{i,j}^{t+1} = wv_{i,j}^t + C_1 r_1 \left(Pbest_{i,j}^t - q_{i,j}^t\right) + C_2 r_2 (Gbest_j^t - q_{i,j}^t) \tag{8}$$

where C_1 and C_2 are generally refer to as cognitive and social parameters respectively. r_1 and r_2 are random values $\in [0, 1]$. w is termed the inertia weight which specifies how the particle's previous velocity affects the velocity of the next iteration. Eq. (9) specifies the value of w.

$$w = w_{max} - iteration \cdot \left(\frac{w_{max} - w_{min}}{max_{-iteration}}\right) \tag{9}$$

$$q_{i,j}^{t+1} = q_{i,j}^t + V_{i,j}^{t+1} \tag{10}$$

The Binary PSO pseudo-code is shown in algorithm 2.

3.7. The opposition base learning (OBL)

OBL proposed by Tizhoosh (2005) reflects an optimization strategy that has been used by many researches to increase the efficiency of their population solutions initiated and local searches by diversifying these solutions. The OBL technique operates in the search space by looking in both directions. These two paths contain the initial solution in one, while the opposite solution represents the other direction. The opposite position in D-dimensional space with $q(q_1, \dots, q_D)$ and $q_i \in \left[\alpha_i, \beta_i\right]$, $i = 1, 2 \dots \dots .D$ is computed by Eq. (11)

$$q_i^{t/OPP} = \alpha_i + \beta_i - q_i^t \tag{11}$$

Algorithm 2: Binary PSO algorithm

Get the c_1, c_2; weight factor w_{max}, w_{min}; maximum velocity v_{max}, and maximum iteration max.iter
The population get randomly set as q for each solution and D dimensional vector as the velocity
set t = 0
$w = w_{max} - iteration \cdot \left(\frac{w_{max} - w_{min}}{max_{-iteration}}\right)$
The fitness value for each of the solution is evaluated $f(q_{i,j}^t)$ while Pbest and gbest values are set
$\boldsymbol{for\ k = 1\ to\ SS}$
  $V_i^{t+1} = wv_i^t + C_1 m_{i1}\left(Pbest_i^t - q_i^t\right) + C_2 m_{i2}(Gbest^t - q_i^t)$
end
$\boldsymbol{for\ k = 1\ to\ SS}$
  $\boldsymbol{for\ j = 1\ to\ D}$
    $\boldsymbol{if}\ (v(i,j)\ greater\ than\ V_{max}$
      $v(i,j)\ equal\ to\ V_{max}$
    end
    $\boldsymbol{if}\ (v(i,j)\ less\ than\ -V_{max}$
      $v(i,j)\ equal\ to\ -V_{max}$
    end
    $s = \frac{1}{1 + e^{-v(i,j)}}$
    if *(rand less than s)*
      $q_{i,j}^{t+1}\ equal\ to\ 1$
    else
      $q_{i,j}^{t+1}\ equal\ to\ 0$
    end
  end
end
t := t + 1
Produce the best solution

4. The proposed algorithm

Algorithm 3 explains the key stages of the proposed ECCSPSOA approach in confronting the feature selection problems and the flowchart of its operation is shown in Fig. 1. Likewise, in this segment, the incorporation of the ideas described in the binary CSA and the binary PSO algorithms are combined which resulted into an algorithm that can gain from their integration, in the ECCSPSOA method, simply targeting some selected crows with the best foods enhances the performance of randomly following every crow in the original CSA. As given in algorithm 4, the method of OBL is then employed to generate the opposite positions of crows and used to update the positions in the PSO. This is done so that both algorithms can alternately explore the search space and not be influenced by the result produced from each other.

Another improvement is the replacement of the random variables in both CSA and PSO by chaotic sequence with 0.7 as the initial value.

1. Logistics map : $q_i^{t+1} = aq_i^t(1 - q_i^t), \quad a = 0.4\ and\ q_1 = 0.7$ (12)
2. Exponential map : $q_i^{t+1} = q_i^t e^{2(1-q_i^t)}, \quad q_1 = 0.7$ (13)

$C_{t+1} = k * q_i^{t+1}$,

k being the energetic parameter that governs the action of q_i^t. When k step up, q_i^t passes through more bifurcations, finally leading to chaos.

Further enhancement on the CSA involves using of tan transformation function (V-shaped) to transform the agents from continuous form to binary form as given in Eq. (13). If a specific random number is less than this threshold value, the current solutions will be changed and the Crow will switch to the new solution field.

$$V_{shape} = \left|\frac{2}{\pi}\arctan\left(\frac{\pi}{2} q_i^t\right)\right| \tag{14}$$

4.1. Features selection problems

In feature selection, if the size of the feature vector is N, the number of different feature combinations tends to be 2^N, which is also a very huge space for exhaustive searching. The suggested hybrid algorithm is used for this purpose to search the feature space dynamically and produce the right combination of features. Features selection fall within multi-objective problems because it must satisfy more than one objectives so as to get the best solution, which minimizes the subset of features selected and at the same time, maximizes the accuracy of the output for a given classifier.

On the basis of the above, the fitness function for determining solutions in this situation built to achieve a balance between the two objectives is set out in the Eq. (14).

$$fitness = \alpha \Delta_R(D) + \beta \frac{|Y|}{|T|} \tag{15}$$

where $\Delta_R(D)$ being the classifier's error rate. $|Y|$ is the size of the subset that the algorithm selects and |T| the total number of the features contains in the current dataset. α is a parameter $\in [0, 1]$ relating to the weight of the error rate of classification, respectively and $\beta = 1 - \alpha$ denotes the significance of the feature reduction. The classification accuracy is accorded a significant and weight rather than amount of features chosen. If the evaluation function only takes into account the accuracy of the classification, the effect would be the neglect of solutions that may have the same accuracy but have less selected features that serve as a major factor in reducing the problem of dimensionality.

Algorithm 3: ECCSPSOA

Established the size of the swarm N with dimension(D) same as the number dataset's attributes
Get the c_1, c_2; weight factor w_{max}, w_{min}; maximum velocity v_{max}, flight length fl, awareness probability AP, and maximum iteration max_iter.
The population get randomly set as q_i^t for each solution and D dimensional vector as the velocity
set t = 0
$w = w_{max} - iteration \cdot \left(\frac{w_{max} - w_{min}}{max_{-iteration}}\right)$
The fitness value for each of the solution is evaluated $f(q_{i,j}^t)$ while Pbest and gbest values are set
Run CCSA with q_i^t as the population, a set of crows with the best foods to be followed and a minimum crow
Inversely mutate the returned position by the CCSA using algorithm 4
Update the position of the swarms
$\boldsymbol{for\ k = 1\ to\ SS}$
  $V_i^{t+1} = wv_i^t + C_1 m_{i1}\left(Pbest_i^t - q_i^t\right) + C_2 m_{i2}\left(Gbest^t - q_i^t\right)$
end
$\boldsymbol{for\ k = 1\ to\ SS}$
  $\boldsymbol{for\ j = 1\ to\ D}$
    $\boldsymbol{if}\ (v(i,j)\ greater\ than\ V_{max}$
      $v(i,j)\ equal\ to\ V_{max}$
    end
    $\boldsymbol{if}\ (v(i,j)\ less\ than\ -V_{max}$
      $v(i,j)\ equal\ to\ -V_{max}$
    end
    $s = \frac{1}{1 + e^{-v(i,j)}}$
    ***if** (rand less than s)*
      $q_{i,j}^{t+1}\ equal\ to\ 1$
    else
      $q_{i,j}^{t+1}\ equal\ to\ 0$
    end
  end
end
t := t + 1
Produce the best solution

Algorithm 4: Opposite crow position

$for\ (i = 1; i \leq N; i++)\ do$
  $if (f(q_i^t) > minCrow^t)\ do$
    $q_i^{t/OPP} = \alpha_i + \beta_i - q_i^t$
  $end\ if$
$end\ for$

4.2. K-nearest neighbour classifier

K-nearest neighbour (KNN) (Kenyhercz & Passalacqua, 2016) is one of the supervised learning algorithms which depend on classifying new instance based on distance from the new instance to the training instances. The KNN classifier does not require any explicit training steps in order to construct a model, this method has been successfully used in a number of areas such as feature selection, artificial intelligence, categorical problems, statistical estimation, and pattern recognition. Also, KNN classifier have been found to be robust when dealing with noisy datasets. In this paper, KNN is used to determine the goodness of the selected features as used by many researchers (Ammar et al., 2019; Gehad et al., 2017; Madhusudhanan et al., 2019) as it produces a good result.

5. Experimental results

The experiments in this paper are carried out in two phases: During the first phase, the results of ECCSPSOA embedded with chaotic maps such as ECCSPSOA1 with logistic map, ECCSPSOA2 with exponential map and ECCSPSOA3 with both logistic and exponential maps (logistic map in CSA and exponential map in PSO) are compared to determine the optimum chaotic map. The optimum ECCSPSOA is then compared with some well-known evolutionary computation techniques algorithms during the second phase of the experiments.

5.1. Definition of dataset

This study witnessed the utilization of fifteen recognized standard datasets collected from the UCI repository (Bache & Lichman, 2019) to ascertain the consistency of ECCSPSOA. Many researchers use these data sets to compare performance in feature selection specialization. These datasets are resumed in Table 1.

In this analysis, fifteen recognized benchmark datasets from the UCI data repository (Bache & Lichman, 2019) were used to validate the consistency of ECCSPSOA. Many researchers use these data sets to compare performance in the field of feature selection. Table 1 nicely describes these datasets in detailed. Max–min normalization technique was employed in normalizing each dataset suitable for the evaluation.

5.2. Performance measures and parameters setting

Algorithms considered in this study were initialized randomly and allowed to run $P = 20$ times independently with 70 iterations, while the size of the population in each algorithm is kept at 20 with the dimension same as the number of attributes in the dataset. Table 2 detailed the parameters for the compared algorithms. All the testing in this study were carried out using Python language on computer with Intel Core™ i3-7100U 2.40 GHz CPU and 4.0 GB RAM. From the validation results, the following statistical measures are recorded.

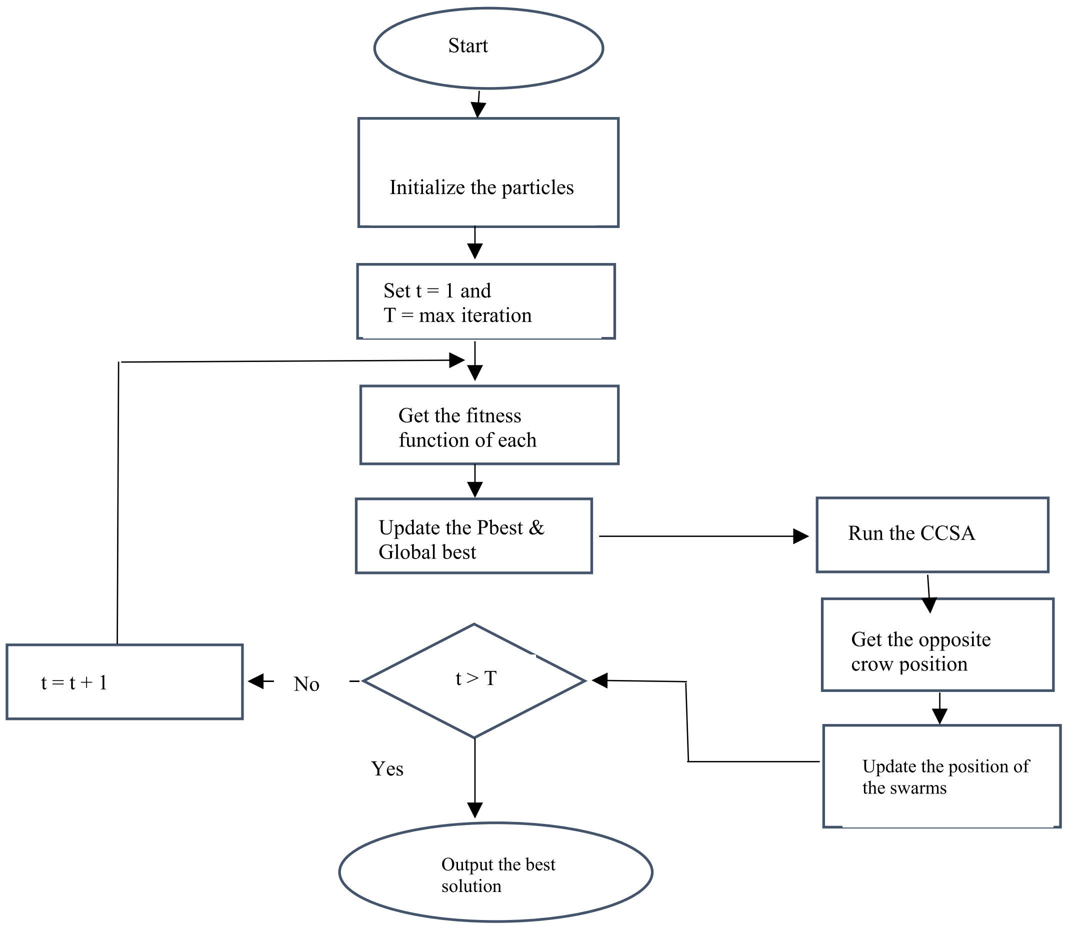

Fig. 1. Flowchart of the ECCSPSOA.

Table 1
Number of datasets utilized in the study.

	Dataset	Number of features	Number of instances
1	Wine	13	178
2	Dermatology	34	366
3	Heart	13	270
4	Ionosphere	34	351
5	Lung cancer	56	32
6	Thoracic surgery	16	454
7	Hepatitis	19	155
8	Parkinson	22	194
9	Phishing website	30	2455
10	Qsar biodegradation	41	1054
11	Absenteeism at work	21	733
12	Divorce	54	169
13	Wpdc	34	192
14	Risk factor cervical cancer	36	858
15	Wdpc	31	568

i. Mean fitness function: This function average the outcome of the fitness function as a result of running the particular algorithm P times, detailed in Eq. (15).

$$mean = \frac{1}{P}\sum_{I=1}^{P} B_i \tag{16}$$

where B_i represents the best fitness value attained at run *i*.

ii. Best fitness function: This function output the smallest number when the algorithm is run P times as described in Eq. (16).

$$Best = \min_{i=1} B_i \tag{17}$$

where B_i stands for the best fitness value found at run *i*.

iii. Worst fitness function: This function output the uppermost number when the algorithm is run P times as described in Eq. (17).

$$Worst = \max_{i=1}^{P} B_i \tag{18}$$

where B_i stands for the worst fitness value attained at run i

iv. Standard deviation: This indicates the algorithm's durability and hardiness. Larger standard deviation numbers will indicate wandering outcomes where smaller value implies the algorithm most of the time converges to the same value. This is calculated using Eq. (18)

$$std = \sqrt{\frac{1}{P-1}\sum_{i=1}^{P}(B_i - Mean)^2} \tag{19}$$

where B_i stands for the value gotten for standard deviation at run i

v. Average Performance (CA): this function produces the average value of the classifier's outcome when the algorithm is run P time the mean of the classification accuracy values resulted when the algorithm is run P times as given by Eq. (19).

$$CA = \frac{1}{P}\sum_{i=1}^{P} CA^i \tag{20}$$

where CA^i stands for the accuracy value acquired at run i

vi. Friedman test: Friedman test (García et al., 2010) (Friedman's two-way analysis of variances by rank) considered to be non-parametric similarity of the two-way parametric variance analysis. The aim of this test is to decide if a sample of results will infer that there is a difference between treatment outcomes. In calculating the test statistics, the first move is to transform the

Table 2
Parameters used in algorithms and fitness functions.

Parameter	Value
Number of iterations	70
Population magnitude	20
Dimension (D)	Same as number of attributes in the dataset
Search domain	[0 ,1]
Number of runs (P)	20
W_{max}	0.9
W_{min}	0.4
V_{max}	6
C_1	2
C_2	2
fl	2
AP	0.2
β	0.01
α	0.99

initial results into rankings. Therefore, it separately ranks the algorithms for each problem, the rank of 1 will be allocated to the best performing algorithm, the second best rank 2, and so on. The average ranks are calculated to resolve the case of ties if encountered. Eq. (20) describes the calculation of test statistic as recommended by Friedman.

$$T_1 = \frac{12}{nK(K+1)} \sum_{K=1}^{K} R^2 - 3n(K+1) \tag{21}$$

where $R = \sum_{i=1}^{n} R_{iK}$ stands for the summation of ranks for algorithm k over the n blocks. This statistical T1 has an asymptotic Chi-square distribution of $K-1$ degrees of freedom, under the null hypothesis, as n lean towards to infinity. The null hypothesis is dismissed at the α significance level if $T_1 \geq \chi^2_{K-1;1-\alpha}$, where $\chi^2_{K-1;1-\alpha}$ is the $(1-\alpha)$ quantile of the distribution of Chi-square with $K-1$ degrees of freedom.

vii. When significant difference is detected between treatment (algorithms), post-hoc test, which are many, is needed to find out which one differs from the other. Benjamini–Hochberg post hoc test (Benjamini & Hochberg, 1995) with 5% significance level was used in this study, which is one of the best procedures for controlling false discovery rate (FDR). This method works by initially test the m hypotheses under consideration to compute test-statistic, which is use to obtained a $p-value$ from the suitable distribution. Let $p_1 \leq p_2 \leq \cdots \leq p_m$ denotes the magnitude of the p-values, and H_q be the null hypothesis conforming to p_q. FDR control at α significance level is then obtain, we reject all H_q for $q = 1, 2, \ldots, k$, with the help of Eq. (21).

$$k = max\left\{q : p_q \leq \frac{q}{m} * \alpha\right\} \tag{22}$$

5.3. Experimental series 1: ECCSPSOA's effectiveness with different chaotic maps

The key purpose of this experimentation is to test with various chaotic maps the effectiveness of ECCSPSOA and to arrive at the optimal chaotic map. Table 3 shows ECCSPSOA integrated with various chaotic maps in relation to precision, average and standard deviation. According to average accuracy values, ECCSPSOA3 could take the highest average accuracy value of 89.76%, while ECCSPSOA2 and ECCSPSOA1 could take 89.44% and 89.28%, respectively. For the mean fitness values of the algorithms, ECCSPSOA3 is much better than the others, with only 0.1038 on average. ECCSPSOA3 also got the best result on standard deviation with the average of 0.0044.

Moreover, we used the Friedman test (Table 4), which is a non-parametric test for multiple (more than 2) dependent samples, to statistically examine the differences between the algorithms. The Chi-square value was 18.75, which was statistically significant (p < 0001).

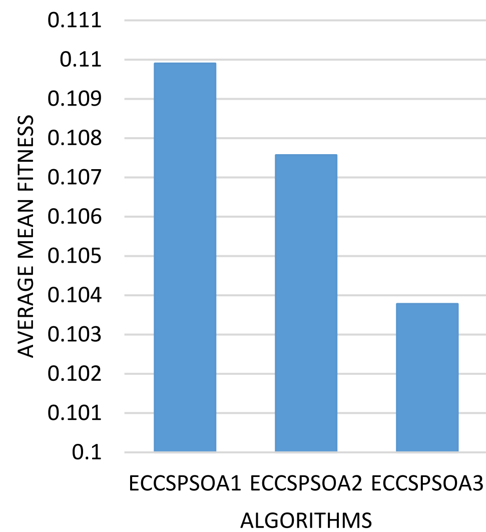

Fig. 2. The average fitness values among the algorithms.

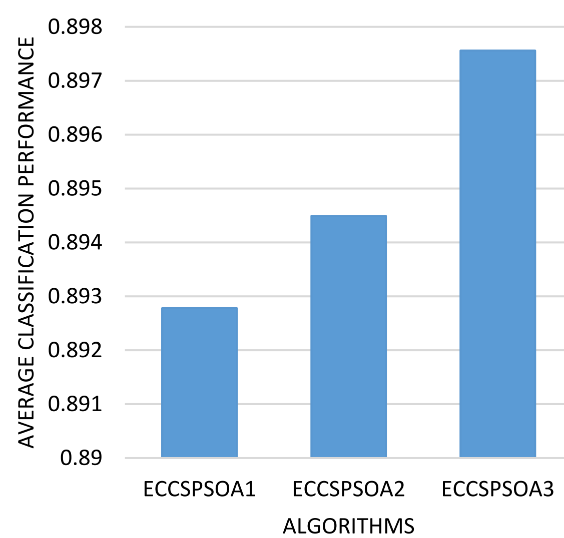

Fig. 3. The average-performance among the algorithms.

After finding the significance between algorithms, we further performed a post-hoc test to find the source of this difference. So, we applied the Nemenyi test (Table 5) to each algorithm combination. The p values obtained were 0.0001, 0.0454, and 0.0343 for the algorithm pairs ECCSPSOA3 vs. ECCSPSOA2, ECCSPSOA3 vs. ECCSPSOA1, and ECCSPSOA2 vs. ECCSPSOA1, respectively. This indicated that there was statistical significance of the accuracy performances between ECCSPSOA3 and ECCSPSOA2 on the other hand, $p = .0343 < 0.05$ indicates that both ECCSPSOA2 and ECCSPSOA1 are also different. Overall, we can choose ECCSPSOA3 as a best algorithm in this experiment because of it higher statistical results. According to these findings, logistic and exponential chaotic maps combined are chosen to be the suitable chaotic maps for ECCSPSOA (see Figs. 2–4).

5.4. Experimental series 2: ECCSPSOA vs. other optimization algorithms

In the second sequence of experiments, the output of ECCSPSOA is contrasted with four optimization algorithms that have been refutably applied to feature selection problems. The optimization algorithms chosen for the purpose of assessment in this study are particle swarm optimization (PSO) (Kennedy & Ebenhart, 1995), binary particle swarm

Table 3
Comparison based on average-performance, average-fitness and average standard deviation.

Dataset	Accuracy			Mean fitness			Standard deviation		
	ECCSPSOA1	ECCSPSOA2	ECCSPSOA3	ECCSPSOA1	ECCSPSOA2	ECCSPSOA3	ECCSPSOA1	ECCSPSOA2	ECCSPSOA3
Wine	**0.9814**	**0.9814**	0.9797	0.024	**0.0236**	0.0246	0.0022	**0.0020**	0.0025
Dermatology	0.9811	0.9821	**0.9840**	0.0237	0.0228	**0.0194**	0.0049	0.0042	**0.0037**
Heart	0.8483	0.8424	**0.8495**	0.1541	**0.1522**	0.1527	0.0065	**0.0030**	**0.0030**
Ionosphere	0.9017	0.9063	**0.9164**	0.1001	0.0950	**0.0841**	0.0094	0.0079	**0.0075**
Lung cancer	0.9352	0.9481	**0.9574**	0.0664	0.0536	**0.0431**	0.0162	0.0183	**0.0133**
Thoracic surgery	0.8528	**0.8531**	0.8529	0.1486	0.1484	**0.1478**	0.0013	0.0011	**0.0003**
Hepatitis	0.9542	0.9542	**0.9680**	0.0484	0.0487	**0.0343**	0.0158	0.0127	**0.0058**
Parkinson	0.9242	0.9247	**0.9263**	0.0782	0.0777	**0.0757**	0.0060	0.0065	**0.0037**
Phishing site	0.9466	**0.9505**	0.9497	0.0574	0.0542	**0.0533**	0.0018	0.0027	**0.0017**
Qsar biod	0.8676	**0.8670**	0.8666	0.1368	0.1368	**0.1359**	**0.0034**	0.0044	0.0047
Absenteeism	0.4154	0.4214	**0.4248**	0.5831	0.5765	**0.5718**	**0.0037**	0.0076	0.0075
Divorce	0.9879	0.9883	**0.9901**	0.0131	0.0126	**0.0103**	0.0026	**0.0003**	0.0027
Wpdc	0.8307	0.8337	**0.8341**	0.1715	0.1682	**0.1626**	**0.0099**	0.0114	0.0101
Risk factor cerv	**0.9833**	**0.9833**	**0.9833**	0.0187	0.0189	**0.0180**	**0.0003**	0.0006	**0.0003**
Wdpc	**0.9813**	0.9809	0.9806	0.0244	0.0243	**0.0231**	0.0027	0.0025	**0.0012**
AVERAGE	**0.8928**	**0.8945**	**0.8976**	**0.1099**	**0.1076**	**0.1038**	**0.0058**	**0.0057**	**0.0045**

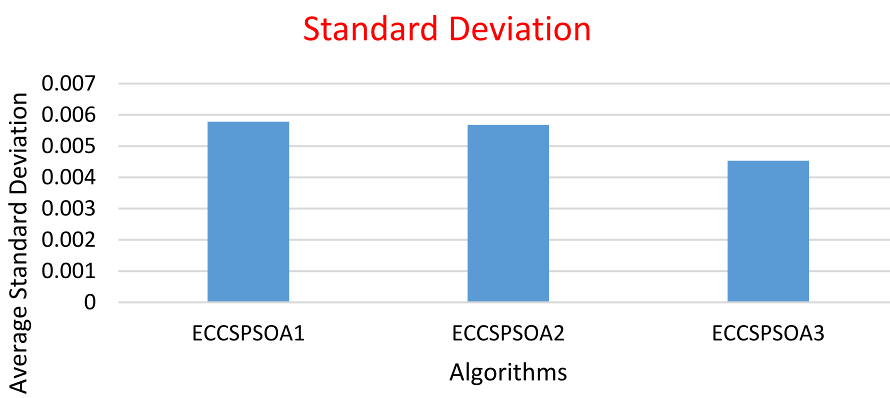

Fig. 4. Comparison of standard deviations between algorithms.

Table 4
Friedman ranking for the fitness function of the three proposed algorithms.

	Data set	ECCSPSOA1	ECCSPSOA2	ECCSPSOA3
1	Wine	2	1	3
2	Dermatology	3	2	1
3	Heart	3	1	2
4	Ionosphere	3	2	1
5	Lung cancer	3	2	1
6	Thoracic surgery	3	2	1
7	Hepatitis	2	3	1
8	Parkinson	3	2	1
9	Phishing website	3	2	1
10	Qsar biodegradation	2.5	2.5	1
11	Absenteeism at work	3	2	2
12	Divorce	3	2	1
13	Wpdc	3	2	1
14	Risk factor cervical cancer	2	3	1
15	Wdpc	3	2	1
	SUM	41.5	30.5	19
	AVERAGE	2.77	2.03	1.27
CHI-SQUARE	18.74576271			
Qcrit2	5.991			
p-value	0.0001			

optimization (BPSO) (Kennedy & Ebenhart, 1995), crow search algorithm (CSA) (Askazadeh, 2016) and chaotic crow search algorithm (CCSA) (Sayed et al., 2018). The settings for all approved optimization algorithms are shown in Table 2. Classification accuracy, number of selected features, mean fitness, best fitness, worst fitness, standard deviation, Friedman test and Benjamini–Hochberg post-hoc test are the performance measures used to evaluate the performance of the algorithms in consideration. A comparison is rendered in Table 8 on the basis of the best classifier's output precision obtained by the algorithms considered in the study. ECCSPSOA leads other selected algorithms with highest average accuracy value of 89.76%, followed by BPSO with 89.32%, PSO with 87.75%, CCSA with 87.50% and CSA coming last with 87.23%. In addition, Table 5 indicates the average number of features selected, As shown in this table, ECCSPSOA exhibits the capability of selecting fewest features of 0.2612 selected on average and also preserving the precision of the classification, which is an indication that ECCSPSOA poses a potential of satisfying two conflicting objectives of optimization. Table 6 describes the results obtained on the basis of the average fitness values. Examining the results, ECCSPSOA produces least fitness values of 0.1038 on average compare to the values output by the other algorithms.

Furthermore, the standard deviation reported in Table 7 test if the convergence of these algorithms is stable, repeatable and robust, ECCSPSOA and BPSO came first with 0.0044 while the CSA got the worst result. An insight from this table, shows the ECCSPSOA ability to converge repeatedly no matter the usage of random number in the initialization techniques. The Friedman test indicated that there was a statistically significant difference between ECCSPSOA and the other optimization algorithms, $\chi^2(4, N = 15) = 43.57$, $p < .0001$. A pairwise Benjamin-Hochberg post hoc test with adjusted p value using ECCSPSOA as a control algorithm was significant for ECCSPSOA vs. CSA ($p = .000004$, adj $\alpha = 0.0125$), ECCSPSOA vs. CCSA ($p = .000005$, adj $\alpha = 0.025$), ECCSPSOA vs. PSO ($p = .000006$, adj $\alpha = 0.0375$) and ECCSPSOA vs. BPSO ($p = .004885$, adj $\alpha = 0.05$). Consequently, we may infer from all of these tests that ECCSPSOA's rigidity and performance are superior to the other algorithms selected in the study, and finally conclude that, ECCSPSOA outperformed the four other optimization algorithms. As a result of the ECCSPSOA algorithm's achieved results, it is clear that it benefited the search strategy and rapid convergence

Table 5
Average of the total feature selected.

	Dataset	ECCSPSOA	PSO	BPSO	CSA	CCSA
1	Wine	**0.5038**	0.5808	0.5385	0.5115	0.6538
2	Dermatology	0.3588	0.5059	0.4882	**0.3485**	0.7809
3	Heart	**0.3192**	0.4269	0.3846	0.3269	0.5731
4	Ionosphere	**0.1294**	0.3779	0.2941	0.1897	0.1397
5	Lung cancer	**0.0946**	0.3884	0.3295	0.225	0.1813
6	Thoracic surgery	0.2156	0.3063	0.2438	0.1781	**0.1156**
7	Hepatitis	0.2684	0.4605	0.2974	**0.2028**	0.2632
8	Parkinson	**0.2773**	0.3568	0.3386	0.2795	0.3273
9	Phishing website	**0.3500**	0.5133	0.4900	0.3567	0.6633
10	Qsar biodegradation	0.3878	0.5439	0.4683	**0.3122**	0.7220
11	Absenteeism at work	0.2405	0.4619	0.3881	**0.1714**	0.2143
12	Divorce	**0.0426**	0.2148	0.2287	0.1546	0.0593
13	Wpdc	0.2574	0.4368	0.4118	0.2691	**0.2441**
14	Risk factor cervical cancer	**0.1543**	0.3557	0.3186	0.2086	0.2157
15	Wdpc	**0.3184**	0.4903	0.4629	0.3274	0.7129
	Average	**0.2612**	0.4280	0.3789	0.2708	0.3911

Table 6
Mean of the total fitness function.

	Data set	ECCSPSOA	PSO	BPSO	CSA	CCSA
1	Wine	0.0246	0.0282	**0.0222**	0.0427	0.0284
2	Dermatology	**0.0194**	0.0363	0.0219	0.0602	0.0422
3	Heart	0.1527	0.1654	**0.151**	0.1726	0.1722
4	Ionosphere	**0.0841**	0.1199	0.1073	0.1145	0.1146
5	Lung cancer	**0.0431**	0.0919	0.062	0.0921	0.1063
6	Thoracic surgery	**0.1478**	0.1527	0.1479	0.1491	0.1492
7	Hepatitis	**0.0343**	0.0943	0.0365	0.0826	0.0919
8	Parkinson	**0.0757**	0.0986	0.0769	0.1042	0.1053
9	Phishing website	**0.0533**	0.0594	0.0552	0.0763	0.0668
10	Qsar biodegradation	**0.1359**	0.1468	0.1394	0.1589	0.1545
11	Absenteeism at work	**0.5718**	0.6000	0.5806	0.5888	0.5972
12	Divorce	**0.0103**	0.0195	0.0168	0.0163	0.0142
13	Wpdc	**0.1626**	0.1969	0.1759	0.1973	0.2010
14	Risk factor cervical cancer	**0.0180**	0.0444	0.0230	0.0442	0.0369
15	Wdpc	**0.0231**	0.0295	0.0246	0.0364	0.0329
	Average	**0.1038**	0.1257	0.1094	0.1291	0.1276

Table 7
Standard deviation of the fitness function.

	Data set	ECCSPSOA	PSO	BPSO	CSA	CCSA
1	Wine	0.0025	0.0048	**0**	0.0087	0.0025
2	Dermatology	0.0037	0.0068	**0.0032**	0.0136	0.0038
3	Heart	0.003	0.0118	**0**	0.0087	0.0061
4	Ionosphere	0.0075	0.0076	**0.0044**	0.0059	0.0102
5	Lung cancer	**0.0130**	0.0249	0.0181	0.0221	0.0181
6	Thoracic surgery	**0.0003**	0.0019	0.0003	0.0009	0.0009
7	Hepatitis	**0.0050**	0.0166	0.0088	0.0173	0.0148
8	Parkinson	**0.0037**	0.0132	0.0044	0.0111	0.0102
9	Phishing website	**0.0015**	0.0035	0.0016	0.0063	0.0018
10	Qsar biodegradation	0.0047	0.0046	0.0034	0.0066	**0.0025**
11	Absenteeism at work	0.0075	0.0087	**0.0055**	0.0115	0.0100
12	Divorce	0.0027	**0.0004**	0.0025	0.0027	0.0039
13	Wpdc	**0.0101**	**0.0101**	0.0057	0.0105	0.0125
14	Risk factor cervical cancer	**0.0003**	0.0161	0.0063	0.0096	0.0073
15	Wdpc	**0.0012**	0.0027	0.0018	0.0022	0.0018
	Average	**0.0044**	0.0089	**0.0044**	0.0092	0.0071

features from the CSA and the PSO, respectively. Furthermore, the improvements to its local search by introducing opposition-based learning technique and chaotic map have made it more suitable in avoiding entrapment in local optima and improving population diversity for optimal feature subset selection (see Figs. 5–10 and Tables 9–11).

6. Conclusion

We have proposed in this study, a hybridized algorithm consisting of an enhanced crow search algorithm and particle swarm optimization to aid in removing features termed as noisy and irrelevant in

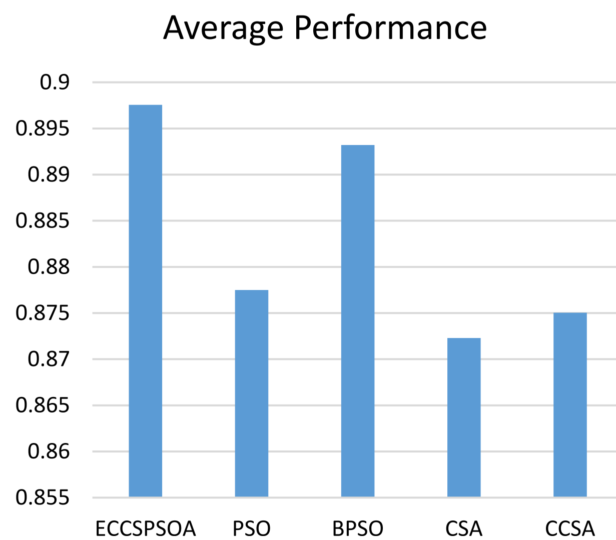

Fig. 5. Average classification performance among the algorithms.

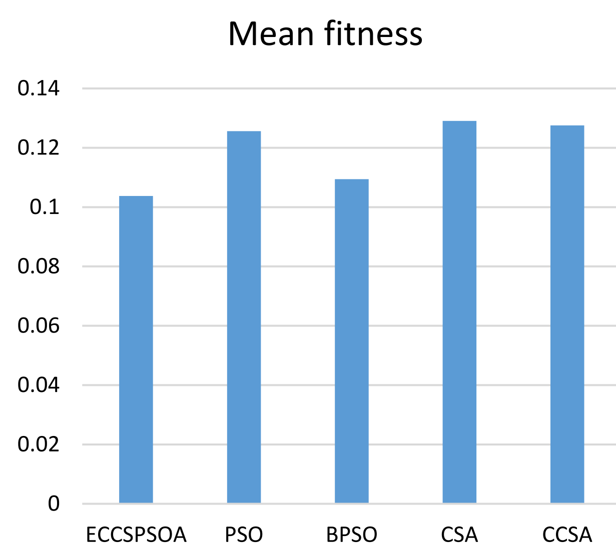

Fig. 6. Average mean fitness function among the algorithms.

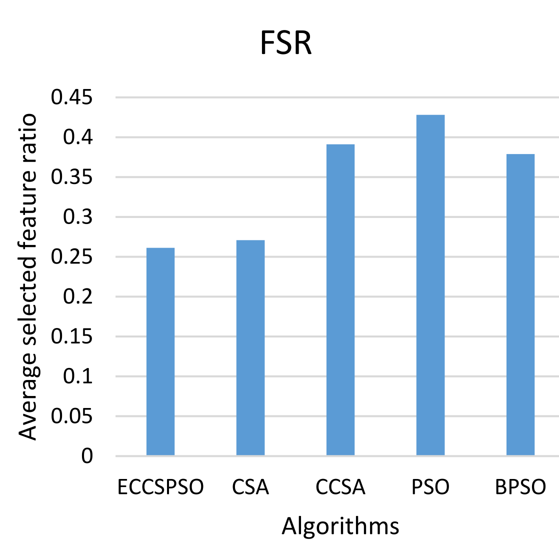

Fig. 7. The average number of features selected by the algorithms.

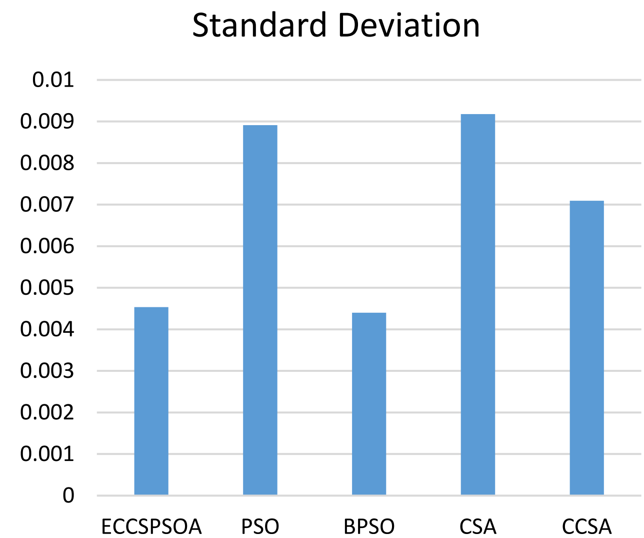

Fig. 8. Average standard deviation.

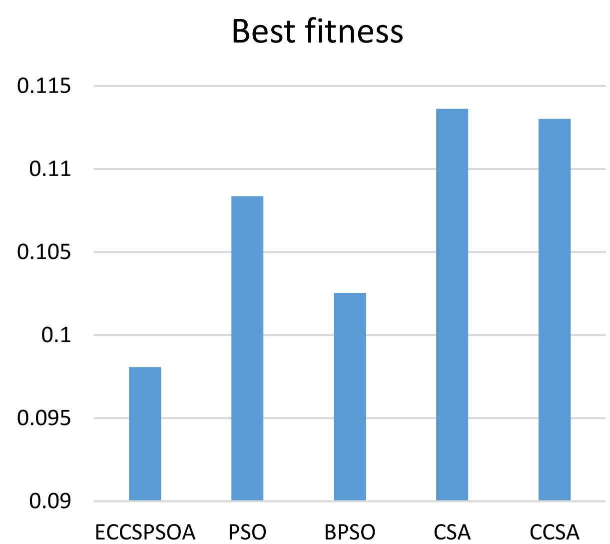

Fig. 9. The total average best fitness values.

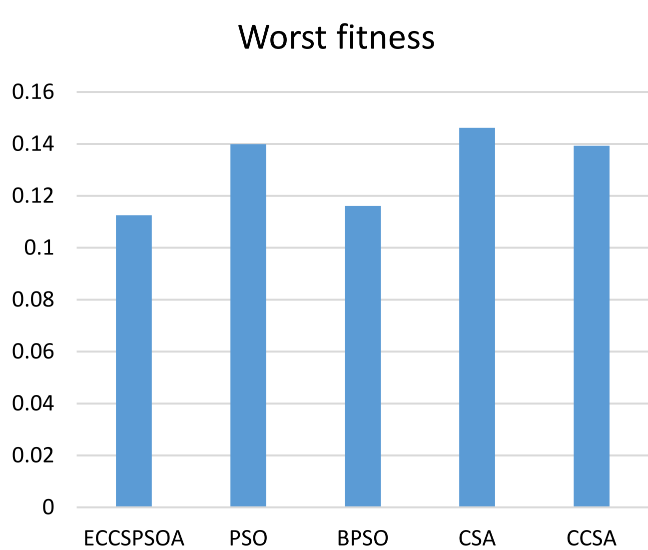

Fig. 10. The total average worst fitness values.

Table 8
Average classification accuracy.

	Data set	ECCSPSOA	PSO	BPSO	CSA	CCSA
1	Wine	0.9797	0.9768	**0.9831**	0.9610	0.978
2	Dermatology	**0.9840**	0.9685	0.9828	0.9427	0.9653
3	Heart	0.8495	0.8372	**0.8513**	0.8290	0.8318
4	Ionosphere	**0.9164**	0.8827	0.8946	0.8863	0.8857
5	Lung cancer	**0.9574**	0.9111	0.9407	0.9093	0.8944
6	Thoracic surgery	0.8529	0.8489	**0.8531**	0.8512	0.8504
7	Hepatitis	**0.9680**	0.9094	0.9662	0.9186	0.9098
8	Parkinson	**0.9263**	0.9040	0.9257	0.8975	0.8969
9	Phishing website	**0.9497**	0.9452	0.9492	0.9265	0.9392
10	Qsar biodegradation	**0.8666**	0.8577	0.8639	0.8427	0.8512
11	Absenteeism at work	**0.4248**	0.3986	0.4175	0.4070	0.3989
12	Divorce	**0.9901**	0.9825	0.9854	0.9854	0.9853
13	Wpdc	**0.8341**	0.8059	0.8247	0.8032	0.7996
14	Risk factor cervical cancer	**0.9833**	0.9588	0.9800	0.9575	0.9650
15	Wdpc	**0.9806**	0.9752	0.9799	0.9666	0.9740
	Average	**0.8976**	0.8775	0.8932	0.8723	0.875

Table 9
The Friedman ranking, statistical test, critical value and *p*-value of the studied algorithm with PSO, BPSO, CSA and CCSA.

	ECCSPSOA	PSO	BPSO	CSA	CCSA
	2	3	1	5	4
	1	3	2	5	4
	2	3	1	5	4
	1	5	2	3	4
	1	3	2	4	5
	1	5	2	3	4
	1	5	2	3	4
	1	3	2	4	5
	1	3	2	5	4
	1	3	2	5	4
	1	5	2	3	4
	1	5	4	3	2
	1	3	2	4	5
	1	5	2	4	3
	1	3	2	5	4
Sum	17	57	30	61	60
Average	2.125	7.125	3.750	7.625	7.500
F(t)	43.57333333				
Critical value	9.487729037				
p-value	7.86777E−09				

Table 10
The significant level and number of tests.

α	0.05
k	4

Table 11
Benjamini–Hochberg test.

Algorithm	Z	*p*-value	Rank	adj α	BH sig
CSA	9.526279442	0.000004	1	0.0125	YES
CCSA	9.309773091	0.000005	2	0.025	YES
PSO	8.660254038	0.000006	3	0.0375	YES
BPSO	2.814582562	0.004885	4	0.05	YES

datasets generally known as features selection. The suggested algorithm is the Enhanced Crow Search Particle Swarm Optimization Algorithm (ECCSPSOA). This combined algorithm provides greater performance than that of each of its component's algorithm. In order to check the soundness and accuracy of the algorithm, the proposed ECCSPSOA algorithm was checked and analysed over 15 UCI repository datasets. The ECCSPSOA algorithm was likened with a variety of well-known optimization algorithms (CSA, CCSA, PSO, and BPSO) grounded on a number of considerations. These considerations include accuracy of the classifier's result, fitness, standard deviation, and the number of features selected. The comparative tests and analyses reveal that in terms of the average classification accuracy, the proposed ECCSPSOA gives the best result on most of the datasets, except on Wine, Heart and Thoracic Surgery datasets which is outperformed by the BPSO algorithm. In terms of the average number of features selected, the ECCSPSOA obtained the best result on 9 of the 15 datasets. Moreover, the original CSA obtained the best result on Dermatology, Hepatitis, Oscar biodegradation and Absenteeism at work datasets, while CCSA had the best result in wpdc and Thoracic surgery dataset. Moreover, in terms of the fitness, the ECCSPSOA had the lowest mean fitness value in 13 of the datasets tested, while the BPSO had the lowest in Wine and Heart datasets. In terms of the standard deviation, which define the stability, repeatability and robustness of the algorithm, the ECCSPSOA obtained a lower standard deviation in Lung cancer, Thoracic Surgery, Hepatitis, Parkinson, Phishing websites, wdpc and Risk factor cervical cancer datasets. Additionally, the PSO and ECCSPSOA had the lowest value on wpdc dataset, while the PSO had the lowest on Divorce dataset. The BPSO had the lowest fitness standard deviation on Wine, Dermatology, Heart, and Absenteeism at work, while CCSA had the lowest standard deviation on Oscar biodegradation dataset. Conclusively, the results shows that the ECCSPSOA is able to contend and/or attained better results compared to other algorithms on most of the datasets.

The proposed ECCSPSOA wrapper feature selection mode has significant practical implications. ECCSPSOA can be extended to other fields such as data mining, data science, sentiment analysis, medical applications, engineering applications, and a lot more. This is focused on the findings of this study performed on different categories of datasets. Though the classification accuracy is the most important measure, but the selection of more features than other optimization algorithms over seven datasets from the 15 datasets used is one drawback of the proposed ECCSPSOA. As a result, a novel selection technique may be utilized in enhancing this algorithm to pick lest features which can be considered as potential future work. Real-world problems like sentiment analysis and cancer detection can be explored with the application of ECCSPSOA for future work. In addition, ECCSPSOA can be examined on various types of datasets. Furthermore, another classifier other than KNN can be used in studying the performance of ECCSPSOA.

CRediT authorship contribution statement

Abdulhameed Adamu: Conceptualization, Methodology, Software, Writing - original draft. **Mohammed Abdullahi:** Investigation, Writing - review & editing. **Sahalu Balarabe Junaidu:** Validation, Supervision. **Ibrahim Hayatu Hassan:** Writing - review & editing.

References

Abdallah, G. Y., & Algamal, Z. Y. (2020). A QSAR classiffication model of skin sensitization potential based on improving binary crow search algorithm. *Electronic Journal of Applied Statistical Analysis*, 86–95.

Abdel-Basset, M., Ding, W., & El-Shahat, D. (2021). A hybrid Harris Hawks optimization algorithm with simulated annealing for feature selection. *Artificial Intelligence Review*, 593–637.

Abdel-Basset, M., El-Shahat, D., El-henawy, I., de Albuquerque, V. H., & Mirjalili, M. (2020). A new fusion of grey wolf optimizer algorithm with a two-phase mutation for feature selection. *Expert Systems with Applications*, Article 112824.

Abdurazaq, E., & Muhammet, T. G. (2021). Advances in science, technology and engineering systems journal. *Optimal Sizing of a Renewable Energy Hybrid System in Libya using Integrated Crow and Particle Swarm Algorithms*, 264–268.

Abhilasha, C., & Tirath, P. S. (2021). Feature selection using binary crow search algorithm with time varying flight length. *Expert Systems with Applications*, 1–16.

Acharya, N., & Singh, S. (2018). An IWD-based feature selection method for intrusion detection system. *Soft Computing*, *22*(13), 4407–4416.

Afarja, M., Jaber, I., Ahmed, S., & Thaher, T. (2021). Whale optimisation algorithm for high-dimensional small-instance feature selection. *International Journal of Parallel, Emergent and Distributed Systems*, 80–96.

Agrawal, R. K., Kaur, B., & Sharma, S. (2020). Quantum based whale optimization algorithm for wrapper feature selection. *Applied Soft Computing*, Article 106092.

Ahmed, M. A., & Mumtaz, A. (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*, 1565–1584.

Al-Ani, A., Alsukker, A., & Khushaba, R. N. (2013). Feature subset selection using differential evolution and a wheel based search strategy. *Swarm and Evolutionary Computation*, 15–26.

Al-Sharaf, H., Bi, Y., Chen, Q., Lensen, A., Mei, Y., Sun …, Y., & Zhang, M. (2019). A survey on evolutionary machine learning power systems based on crow search algorithm. *Archives of Electrical Engineering*, 123–128.

Al-Tashi, Q., Kadir, S. J., Rais, H. M., Mirjalili, S., & Alhussian, H. (2019). Binary optimization using hybrid grey wolf optimization for feature selection. *IEEE Access*, 39496–39508.

Aljarah, I., Ala'M, A. Z., Faris, H., Hassonah, M. A., Mirjalili, S., & Saadeh, H. (2018). Simultaneous feature selection and support vector machine optimization using the grasshopper optimization algorithm. *Cognitive Computation*, *10*(3), 478–495.

Almasoudy, F. H., Al-Yaseen, W. L., & Idrees, A. K. (2020). Differential evolution wrapper feature selection for intrusion detection system. *Procedia Computer Science*, 1230–1239.

Alweshah, M., Alkhalaileh, S., Albashish, D., Mafarja, M., Bsoul, Q., & Dorgham, O. (2021). A hybrid mine blast algorithm for feature selection problems. *Soft Computing*, 517–534.

Amini, F., & Hu, G. (2021). A two-layer feature selection method using genetic algorithm and elastic net. *Expert Systems with Applications*, Article 114072.

Ammar, A., Mohammed, A., Khalayleh, S. A., Mohammed, A.-R., & Riyadh, Q. (2019). Metaheuristic algorithms-based feature selection approach for intrusion detection: Principles, algorithms, and practices. In *Machine learning for computer and cyber security* (pp. 184–208).

Anter, A. M., Hassenian, A. E., & Oliva, D. (2019). An improved fast fuzzy c-means using crow search optimization algorithm for crop identification in agricultural. *Expert Systems with Applications*, 340–354.

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

Askazadeh, A. (2016). A novel metaheuristic method for solving constrained engineering optimization problems: Crow search algorithm. *Computer Structure*, *169*, 1–12.

Bache, K., & Lichman, M. (2019). UCI machine learning repository. Retrieved from http://archive.ics.uci.edu/ml.

Balasaheb, H. P., & P, M. P. (2018). Crow search algorithm with discrete wavelet transform to aid mumford shah inpainting model. *Evolutionary Intelligence*, (2018), 73–87.

Benjamini, Y., & Hochberg, Y. (1995). Controlling the false discovery rate: A practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society*, 289–300.

Bishwajit, D., Biplab, B., Apoorv, S., & Kumar, S. (2020). Solving energy management of renewable integrated microgrid systems using crow search algorithm. *Soft Computing*, (2020), 10433–10454.

Cai, J., Luo, J., Wang, S., & Yang, S. (2018). Feature selection in machine learning: a new perspective. *Neurocomputing*, 70–79.

Chandrashekar, G., & Sahin, F. (2013). A survey on feature selection methods. In *Electrical and microelectronic engineering*. Rochester Institute of Technology.

Chaudhuri, A., & Sahu, T. P. (2021). Feature selection using binary crow search algorithm with time varying flight length. *Expert Systems with Applications*, Article 114288.

Das, H., Naik, B., & Behera, H. S. (2020). A jaya algorithm based wrapper method for optimal feature selection in supervised classification. *Journal of King Saud University-Computer and Information Sciences*.

Dash, M., & Luo, H. (1997). Feature selection for classification. *Intelligent Data Analysis*, 131–156.

de Souza, R. C., de Macedo, C. A., dos Santos, C. L., Pierezan, J., & Mariani, V. C. (2020). Binary coyote optimization algorithm for feature selection. *Pattern Recognition*, Article 107470.

Devikanniga, D., Ramu, A., & Haldorai, A. (2020). Efficient diagnosis of liver disease using support vector machine optimized with crows search algorithm. *EAI Endorsed Transactions on Energy Web*, 1–10.

Du, L., Lv, X., Ren, C., & Chen, Y. (2019). A filter-based unsupervised feature selection method via improved local structure preserving. In *2019 5th international conference on big data and information analystics (BigDIA)* (pp. 162–169).

Eesa, A. S., Orman, Z., & Brifcani, A. M. A. (2015). A novel feature-selection approach based on the cuttlefish optimization algorithm for intrusion detection systems. *Expert Systems with Applications*, *42*(5), 2670–2679.

Emary, E., & Zawbaa, H. M. (2019). Feature selection via Lèvy Antlion optimization. *Pattern Analysis and Applications*, 857–876.

Emary, E., Zawbaa, H., & Hassanien, A. (2016). Binary gray wolf optimization approaches for feature selection. *Neurocomputing*, 371–381.

Figueiredo, M. A., & Jain, A. K. (2002). Unsupervised learning of finite mixture models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 381–396.

Figueiredo, A. T., Member, S., Law, M. H., Member, S., & Jain, A. K. (2004). Simultaneous feature selection and clustering using mixture models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1154–1166.

Fred, A. L., Kumar, F., Padmanaban, S., Gulyas, P. B., & Kumar, H. A. (2020). Fuzzy-crow search optimization for medical image segmentation. In *Applications of hybrid metaheuristic algorithms for image processing* (pp. 413–439). Switzerland: Springer.

García, S., Fernández, A., Luengo, J., & Herreea, F. (2010). Advanced nonparametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining: Experimental analysis of power. *Information Sciences*, 2044–2064.

Gehad, I. S., Aboul, E. H., & Ahmad, T. A. (2017). Feature selection via a novel chaotic crow search algorithm. *Neural Computing and Applications*, 1–18.

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

Ghosh, M., Guha, R., Sarkar, R., & Abraham, A. (2019). A wrapper-filter feature selection technique based on ant colony optimization. *Neural Computing and Applications*, 1–19.

Gokalp, O., Tasci, E., & Ugur, A. (2020). A novel wrapper feature selection algorithm based on iterated greedy metaheuristic for sentiment classification. *Expert Systems with Applications*, Article 113176.

Gu, S., & Cheng, R. (2018). Feature selection for high-dimensional classification using a competitive swarm optimizer. *Soft Computing*, *22*(3), 811–822.

Guha, R., Ghosh, M., Mutsuddi, S., Sarkar, R., & Mirjalili, S. (2020). Embedded chaotic whale survival algorithm for filter–wrapper feature selection. *Soft Computing*, 12821–12843.

Gupta, D., Sundaram, S., Khanna, A., Hassanien, A. E., & Victor, H. A. (2018). Improved diagnosis of Parkinson's disease using optimize crow search algorithm. *Computer and Engineering*, 412–424.

Hancer, E., Xue, B., & Zhang, M. (2018). Differential evolution for filter feature selection based on information theory and feature ranking. *Knowledge-Based Systems*, *140*, 103–119.

Kennedy, J., & Ebenhart, R. (1995). Particle swarm optimization in neural networks. In *Proceeding of IEEE international conference* (pp. 1942–1948).

Kenyhercz, M. W., & Passalacqua, N. V. (2016). Missing data imutation methods and their performance with biodistance analyses. *Biological Distance Analysis*, 181–194.

Kumar, L., & Bharti, K. K. (2019). An improved BPSO algorithm for feature selection. In *Recent trends in communication, computing, and electronics* (pp. 505–513). Singapore: Springer.

Kuo, R. J., & Hong, C. W. (2018). Integration of genetic algorithm and particle swarm optimization for investment portfolio optimization. *Applied Mathematics & Information Sciences*, 2397–2408.

lbashish, D., Hammouri, A. I., Braik, M., Atwan, J., & Sahran, S. (2021). Binary biogeography-based optimization based SVM-RFE for feature selection. *Applied Soft Computing*, Article 107026.

Li, Y., Lu, B. L., & Wu, Z. F. (2006). A hybrid method of unsupervised feature selection based ranking. In *International conference on pattern recognition* (pp. 687–690).

Li, Y., Lu, B. L., & Wu, Z. F. (2020). Unsupervised feature selection algorithm based on information gain. 94(Bdece) 63-67.

Madhusudhanan, B. P. S., Shunmuga-Karpagam, N., Mahesh, A., & Suhi, A. P. (2019). An hybrid metaheuristic approach for efficient feature selection. *Cluster Computing*, 14541–14549.

Mafarja, M., & Mirjalili, S. (2018). Whale optimization approaches for wrapper feature selection. *Applied Soft Computing*, *62*, 441–453.

Mobeen, A., Muhammad, A., Hyeonjoon, M., Seong, J. Y., & Han, D. (2020). Image classification based on automatic neural architecture search using binary crow search algorithm. *IEEE Access*, Article 189891-189912.

Nagpal, A., & Singh, V. (2018). A feature selection algorithm based on quantitative mutual information for cancer microarray data. *Procidia Computer Science*, 244–252.

Nouri-Moghaddam, B., Ghazanfari, M., & Fathian, M. (2021). A novel multi-objective forest optimization algorithm for wrapper feature selection. *Expert Systems with Applications*, Article 114737.

Pankaj, U., & Jitender, K. C. (2020). Kapur's entropy based optimal multilevel image segmentation using Crow Search Algorithm. *Applied Soft Computing*, 1–15.

Reddy, S., & Gopal, K. S. (2021). An efficient metaheuristic algorithm based feature selection and recurrent neural network for DoS attack detection in cloud computing environment. *Applied Soft Computing*, 1–11.

Rizk, R.-A. M., Aboul, E. H., & Adam, S. (2020). Multi-objective orthogonal opposition-based crow search algorithm for large-scale multi-objective optimization. *Neural Computing and Applications*, 1–25.

Rodrigues, D., Pereira, L. A., Nakamura, R. Y., Costa, K. A., Yang, X. S., Souza, A. N., & Papa, J. P. (2014). A wrapper approach for feature selection based on bat algorithm and optimum-path forest. *Expert Systems with Applications*, *41*(5), 2250–2258.

Roth, V., & lange, T. (2004). Feature selection in clustering problems. In *Advances in neural information processing* (pp. 1–17).

Saha, A., Bhattacharya, A., Das, P., & Chakraborty, A. K. (2017). Crow search algorithm for solving optimal power problem. In *Proc. 2nd int. conf. electr. comput. commun. technol. (ICECCT)* (pp. 1–8).

Sahebi, G., Movahedi, P., Ebrahimi, M., Pahikkala, T., Plosila, J., & Tenhunen, T. (2020). GeFeS: A generalized wrapper feature selection approach for optimizing classification performance. *Computers in Biology and Medicine*, Article 103974.

Sasan, M., Mehdi, B., Majid, M., & Pierluigi, S. (2019). Designing of stand-alone hybrid PV/wind/battery system using improved crow search algorithm considering reliability index. *International Journal of Energy and Environmental Engineering*, 429–449.

Sayed, G. I., Darwish, A., & Hassanien, A. (2018). *Chaotic crow search algorithm for engineering and constarained problems*. IEEE.

Sayed, G. I., Hassanien, A. E., & Azar, A. T. (2017). Feature selection via a novel chaotic crow search algorithm. *Neural Computing and Application*, *2101*, 7–2988.

Shah, S. A., Shabbir, H. M., & Rehman, S. U. (2020). A comparative study of feature selection approaches: 2016-2020. *International Journal of Scientific & Engineering Research*, 469–478.

Solorio-Fernandez, S., Carrasco-Ochoa, J. A., & Mertinez-Trinidad, J. F. (2016). A new hybrid filter-wrapper feature selection method for clustering based on ranking. *Neurocomputing*, 866–880.

Talbi, E. G., Basseur, M., Nebro, A. J., & Alba, E. (2012). Multi-objective optimization using metaheuristics: non-standard algorithms. *nternational Transactions in Operational Research*, *19*(1–2), 283–305.

Tawhid, M. A., & Dsouza, K. B. (2018). Hybrid binary bat enhanced particle swarm optimization algorithm for solving feature selection problems. *Applied Computing and Informatics*, *16*(1/2), 117–136.

Tawhid, M. A., & Ibrahim, A. M. (2020). Feature selection based on rough set approach, wrapper approach, and binary whale optimization algorithm. *International Journal of Machine Learning and Cybernetics*, 573–602.

Thom, R. C., Macedo, C. A., Coelho, L. S., & Pierezan, J. (2018). A v-shaped binary crow search algorithm for feature selection. In *IEEE congress on evolution computing*.

Tizhoosh, H. R. (2005). Opposition-based learning: A new scheme for machine intelligence. In *Proc. of 2005 intern conf on comput intelli for modelling, control and automation, and international confe, intelligent agents, web technologies and internet commerce (CIMCA-IAWTIC'05)*. IEEE.

Too, J., & Mirjalili, S. (2021a). A hyper learning binary dragonfly algorithm for feature selection: A COVID-19 case study. *Knowledge-Based Systems*, Article 106553.

Too, J., & Mirjalili, S. (2021b). General learning equilibrium optimizer: a new feature selection method for biological data classification. *Applied Artificial Intelligence*, 247–263.

Tubishat, M., Ja'afar, S., Alswaitti, M., Mirjalili, S., Idris, N., Ismail, M. A., & Omar, M. S. (2021). Dynamic salp swarm algorithm for feature selection. *Expert Systems with Applications*, Article 113873.

Upadhyaya, K., & Chhabra, K. J. (2019). Kapur's entropy based optimal multilevel image segmentation using crow search algorithm. *Applied Computing*.

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

Xiaoxia, H., Quanxi, X., Lin, Y., Yingchao, D., Gang, X., & Xinying, X. (2020). An improved crow search algorithm based on spiral search mechanism for solving numerical and engineering optimization problems. *IEEE Access*, 1–23.

Zhang, X. T., Zhang, Y., Gao, H. R., & He, C. L. (2018). A wrapper feature selection algorithm based on brain storm optimization. In *International conference on bio-inspired computing: Theories and applications* (pp. 308–315). Singapore: Springer.