Neural Comput & Applic (2019) 31:991–1006
DOI 10.1007/s00521-017-3131-4

ORIGINAL ARTICLE

Chaotic multi-verse optimizer-based feature selection

Ahmed A. Ewees[1] · Mohamed Abd El Aziz[2] · Aboul Ella Hassanien[3]

Received: 23 November 2016 / Accepted: 19 June 2017 / Published online: 29 June 2017
© The Natural Computing Applications Forum 2017

Abstract The multi-verse optimizer (MVO) is a new evolutionary algorithm inspired by the concepts of multiverse theory namely, the white/black holes, which represents the interaction between the universes. However, the MVO has some drawbacks, like any other evolutionary algorithms, such as slow convergence and getting stuck in local optima (maximum or minimum). This paper provides a novel chaotic MVO algorithm (CMVO) to avoid these drawbacks, where chaotic maps are used to improve the performance of MVO algorithm. The CMVO algorithm is applied to solve the feature selection problem, in which five benchmark datasets are used to evaluate the performance of CMVO algorithm. The results of CMVO is compared with standard MVO and two other swarm algorithms. The experimental results show that logistic chaotic map is the best chaotic map that increases the performance of MVO, and also the MVO is better than other swarm algorithms.

Keywords Multi-verse optimizer · Chaotic maps · Feature selection · Dimensionality reduction

✉ Ahmed A. Ewees
ewees@du.edu.eg; a.ewees@hotmail.com

Mohamed Abd El Aziz
abd_el_aziz_m@yahoo.com

Aboul Ella Hassanien
aboitcairo@gmail.com

[1] Department of Computer, Damietta University, Damietta, Egypt

[2] Department of Mathematics, Faculty of Science, Zagazig University, Zagazig, Egypt

[3] Faculty of Computers and Information, Cairo University, Giza, Egypt

1 Introduction

Nowadays, there are huge amounts of data and resources in various fields which make the digital processing of raw data becomes a challenging issue. These data should be kept safe from damage or loss. So, it can be stored and arranged in physical disks in folders or datasets; these datasets may include a large number of attributes and features. However, not all of these features are important and may be irrelevant or redundant, which may adversely affect the processing accuracy and increase the computational time due to the large search space [1]. Therefore, when researchers want to manipulate these datasets, the best practice is to apply feature selection to choose the significant, best, important, and optimal subset of them [2]. Also, the irrelevant or redundant features are removed, which leads to improving the efficiency, achieving better accuracy, and reducing data complexity [1].

The searching for the optimal solution in a large search space is NP-hard problem [3] and is considered as a multiobjective problem (minimize the number of the selected feature and maximize the classification accuracy) [4]. Therefore, there are several models of feature selection considered as an optimization problem to avoid an expensive computational time and stagnation in local optima. Evolutionary computation and swarm intelligence methods are effective techniques used in this problem; the genetic algorithms (GA) are an example of evolutionary computation [5]. The particle swarm optimization (PSO) [6], artificial bee colony (ABC) [7], grey wolf optimization (GWO) [8], ant colony optimization (ACO) [9], and ant lion optimizer (ALO) [10] are examples of swarm intelligence.

These swarm intelligence algorithms are computational intelligence-based methods, which are made up of a population of artificial agents and inspired by the social behavior of animals in the real world. Most of these algorithms are applied in feature selection problem. In [1] a binary ABC algorithm is introduced for choosing optimal features and tested with ten benchmark datasets. This algorithm achieved the best classification performance in almost all cases as well as outperformed other methods such as GA and PSO, whereas [11] proposed a new algorithm based on binary bare bones PSO algorithm and *k*NN that proved best average classification accuracies for 88% of the experiments' datasets. In addition, [12] introduced a modified cuckoo search algorithm with rough sets for feature selection, and also, the authors in [13] proposed a binary ALO approach to select the optimal feature subset in order to maximize the classification accuracy and minimize the number of selected features. This approach was compared with three well-known optimization algorithms, namely PSO, GA, and binary bat algorithm; the results showed better accuracy than other algorithms. As well as, social spider algorithm [14], GWO [15], binary bat algorithm [16], and many other methods such as [17–19] provided good exploitation and exploration in general optimization problems, especially when they have been applied in feature selection problems; however, their accuracy, time consumption, and finding global optimum still require more efforts to improve them.

In this direction, there are various studies provided many tries to overcome these drawbacks. One of the most effective tries is improving optimization methods by adding chaotic sequences instead of random sequences; this technique proved its performance in escaping from local minima than other stochastic methods [20]. In [21], chaotic krill herd (CKH) was introduced for solving optimization problems; the Singer map is used to adjust the three main movements of the krill by regulating the KH's inertia weights. The result of CKH showed better performance than basic KH and other robust optimization approaches, whereas [22] improved fruit fly optimization algorithm (FOA) by introduced a new parameter (alpha) to generate food sources. This parameter was integrated with chaotic maps to produce chaotic FOA (CFOA) and tested in 14 well-known test functions. The Chebyshev map proved best performance and CFOA showed fast convergence rate and good ability to find the global optimum. In addition, there are many other studies utilized chaotic to solve optimization problem and proved effectiveness against standard methods, such as [23–26], and [27].

In the other hand, chaotic maps are applied in feature selection problems; in this trend, [4] proposed a chaotic ALO (CALO) by adapted the parameter which is used to improve the trade-off between exploration and exploitation. The performance of CALO was better than ALO, PSO, and GA. In the same efforts, [28] provided chaotic binary PSO (CBPSO), and it used two chaotic maps, namely logistic and Tent to set the inertia weight of BPSO. The results showed that the BPSO with tent map achieved higher accuracy than logistic map. Improved chaotic genetic algorithm (ICGA) was introduced by Li et al. [29]. It used the tent map to generate the initial population and logistic map in mutation operation. The results showed better performance than other methods.

Therefore, the ability of chaotic is that it can help optimization methods by overcoming many drawbacks, such as trapping in local minima, slow searching, premature convergence, and time consuming; all these are motivating us to utilize chaotic maps to improve MVO and maintain the population diversity in the problem of interest. In this paper, we propose a new feature selection algorithm, which combines chaotic maps with MVO. The main advantages of this improving are to maximize the classification accuracy and minimize the size of the selected feature. The rest of this paper is arranged as follows: Section 2 provides the concept of MVO algorithm. Chaotic maps are introduced in Sect. 3. The proposed algorithm is given in Sect. 4. Section 5 discusses the experiments results. The conclusion of this paper and future work are given in the last section.

2 Materials and methods

2.1 Multi-verse optimizer (MVO) algorithm

Multi-verse optimizer (MVO) is a new nature-inspired algorithm that emulates the theory of multi-verse in physics, and it simulates the interaction between universes. It was introduced by Mirjalili to deal with various optimization problems [30].

2.1.1 Inspiration

Following [30], the authors mentioned that in the multi-verse theory states, there is more than one big bang that causes the generation of a universe. The concept of multi-verse points to the existence of other universes in addition to our universe [31], and this is opposite of universe. These universes can be interact and/or collide with each other based on the multi-verse theory. There are three concepts in the multi-verse theory, namely white holes, black holes, and wormholes. A white hole (also called the big bang) is created when the collisions between parallel universes occur; therefore, it is considered as the main component for the birth of a universe, and in our universe, these holes are not seen. Unlike a white whole, black holes (have been observed frequently) attract everything including light beams with their extremely high gravitational force [32]. Finally, the wormholes connect different parts of a universe together, and they act as time/space travel tunnels where objects are able to travel instantly between any corners of a universe (or even from one universe to another). The objects are allowed to move between different universes through white/black hole tunnels. When a white/black tunnel is established between two universes, the universe with higher inflation rate is considered to have a white hole, whereas the universe with less inflation rate is assumed to own black holes. The objects are then transferred from the white holes of the source universe to black holes of the destination universe.

2.1.2 Mechanism

In this section, MVO algorithm is illustrated based on the concept of a white hole and black hole in order to explore search spaces and the wormholes are used to improve the quality of MVO in exploiting the search spaces.

In MVO algorithm [30], white holes tend to transmit objects to other universes, whereas black hole tend to receive these objects. So, over the iterations, the inflation rates of all universes are enhanced. Wormholes helps in maintaining the diversity of universes to improve the exploration phase of the search space and the exploitation phase, and prevent getting trapped in local optima. So, the MVO algorithm starts with generating random universes, in every iteration, objects use the white/black holes to transferee from the universe with high inflation rates to other universe with low inflation rates. In addition, the objects in any universe are moved by random teleportations via wormholes toward the best universe. These processes are repeated until the end criteria is satisfied.

Therefore, there are three rules are implemented in MVO algorithm: (1) If the inflation rate is high, this increases the probability of having a white hole and decreases the probability of having a black hole. (2) The universes send objects from a white hole and receive them from a black hole, according to their inflation rate. (3) The universes' objects may be updated by the objects of the universe, which has the best inflation rate by wormholes.

In MVO algorithm [30], each universe represents a solution, and each object in the universe is a variable in the solution. Each universe also, has an inflation rate, which is proportional to the corresponding fitness function value of the solution.

The mathematical model of the white/black hole tunnels is represented by considering the population of universes U as [30]:

$$U = \begin{bmatrix} x_1^1 & x_1^2 & \dots & x_1^d \\ x_{21} & x_2^2 & \dots & x_2^d \\ \vdots & \vdots & \vdots & \vdots \\ x_n^1 & x_n^2 & \dots & x_n^d \end{bmatrix} \tag{1}$$

where d is the dimension of the problem (number of parameters) and N is the number of universes. These universes are sorted based on their inflation rates (fitness function values), and then the object x_i^j (the jth parameter of ith universe) is exchanged by using the roulette wheel mechanism that selects universe U_k as in Eq. (2):

$$x_i^j = \begin{cases} x_k^j & \text{if} \quad r_1 < NI(U_i) \\ x_i^j & \text{otherwise} \end{cases} \tag{2}$$

where $r_1 \in [0, 1]$ is a random number, U_i is the ith universe, and $NI\,(U_i)$ is normalized inflation rate (fitness value) of the U_i.

The less inflation rate indicates that a higher probability of sending objects through white/black hole tunnels. By supposing that each U_i has wormholes (that transport the objects of U_i through space randomly), the exploitation is performed and the diversity of universes is maintained. These wormholes are changing the objects of the universes in random form without consideration of their inflation rates. Also, the wormholes are used to update the universes' objects and improve the rate of inflation by changing the objects of the universe which has the best inflation rate as the following:

$$x_i^j = \begin{cases} \begin{cases} X_j^b + \mathrm{TDR} \times \left((ub_j - lb_j) \times r_4 + lb_j \right) & r_3 < 0.5 \\ X_j^b - \mathrm{TDR} \times \left((ub_j - lb_j) \times r_4 + lb_j \right) & r_3 \geq 0.5 \end{cases} & r_2 < \mathrm{WEP} \\ x_i^j & r_2 \geq \mathrm{WEP} \end{cases} \tag{3}$$

where X_j^b shows the jth parameter in the best solution, lb_j and ub_j indicate the lower and the upper bounds respectively in jth variable, r_2, r_3, and r_4 are random numbers in [0, 1].

The traveling distance rate (TDR) is a coefficient that is applied to determine the distance to allow a wormhole move the object to the best universe, and it is defined as:

$$\mathrm{TDR} = 1 - \frac{t^{1/p}}{T^{1/p}} \tag{4}$$

where t equals the current iteration, T indicates the length of iterations, and p equals 6 as a default value and determines the exploitation accuracy over the iterations.

The wormhole existence probability (WEP) is used to increase linearly over the repetition to emphasize exploitation and the WEP is defined as:

$$\mathrm{WEP} = \mathrm{WEP}_{\min} + t \times \left(\frac{\mathrm{WEP}_{\max} - \mathrm{WEP}_{\min}}{T} \right) \tag{5}$$

where $\mathrm{WEP}_{\min}$ equals 0.2 as a default value, $\mathrm{WEP}_{\max}$ equals 1 as a default value.

The computational complexity of the MVO algorithm is calculated based on the number of iterations and universes as well the mechanism of roulette wheel and sorting. The sorting algorithm has the complexity of $O(n \log n)$ in the best case and $O(n^2)$ in the worst case. And the roulette wheel selection is of $O(n)$ or $O(\log n)$ depending on the implementation. The following equations show the overall computational complexity [30]:

Table 1 Five different chaotic maps used in this study

No	Map name	Equation
1	Tent	$x_{i+1} = \begin{cases} \frac{x_i}{0.7} & x_i < 0.7 \\ \frac{10}{3}(1 - x_i) & x_i \geq 0.7 \end{cases}$
2	Logistic	$x_{i+1} = ax_i(1 - x_i), a = 4$
3	Singer	$x_{i+1} = \mu\left(7.86x_i - 23.31x_i^2 + 28.75x_i^3 - 13.301875x_i^4\right)\mu = 1.07$
4	Sinusoidal	$x_{i+1} = ax_i^2 sin(\pi x_i), a = 2.3$
5	Piecewise	$x_{i+1} = \begin{cases} \frac{x_i}{p} & 0 \leq x_i < p \\ \frac{x_i - p}{0.5 - p} & p \leq x_i < 0.5 \\ \frac{1 - x_i - p}{0.5 - p} & 0.5 \leq x_i < 1 - p \\ \frac{1 - x_i}{p} & 1 - p \leq x_i < 1 \end{cases}$

$$O(\text{MVO}) = O\left(l\left(n^2 + n \times d \times \log\, n\right)\right) \qquad (6)$$

where n is the universes' number, l is the maximum iterations' number, and d is the objects' number.

2.2 Chaotic maps

Chaotic methods have important properties such as ergodicity, stochastically intrinsic, and showing irregular conduct as well as sensitive dependence on the initial conditions [4, 33]. These properties have been translated to various equations which are called "chaotic maps" to be applicable for using in computational applications such as optimization problem. So, using these maps to update random variables in optimization methods is called chaotic optimization algorithm (COA) [21]. This change makes optimization methods inherit the strength of chaos such as the ergodic and non-repetition; so, it can escape from local optima and attain a high-speed searches than random search.

In this paper, we use one-dimensional, non-invertible maps to create a set of chaotic values, to adjust MVO parameters. In Table 1 there are five chaotic maps which are used in the experiments.

3 The proposed chaotic multi-verse optimizer-based feature selection

In this section, the proposed algorithm for feature selection in wrapper mode is illustrated, which the chaotic theory is combined with the standard MVO algorithm. The chaotic theory has an important feature that its ability to change the initial values for the data which makes different in system behaviors. As in the standard MVO algorithm, Eq. (3) is the main equation to improve the inflation rate using wormholes, and the parameter r_4 is an important parameter that affects in updating position in the exploration phase. Therefore, the tuning of this parameter using chaotic maps plays an important role to improve the MVO mechanism to perform exploitation and avoid local optima. The proposed algorithm (called CMVO) adapts this parameter in each iteration of the optimization process. In Fig. 1 and Algorithm 1 the proposed algorithm is illustrated.

The CMVO starts with constructing the chaotic map (using one from Table 1), then generating a random population of size N and dimension d, in which each universe

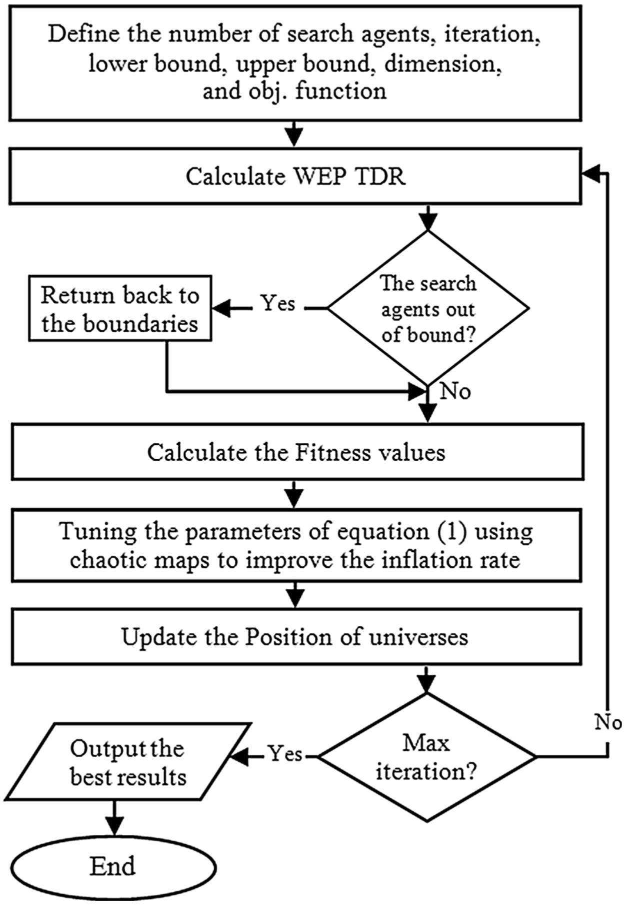
Define the number of search agents, iteration, lower bound, upper bound, dimension, and obj. function
Calculate WEP TDR
The search agents out of bound?
Yes
Return back to the boundaries
No
Calculate the Fitness values
Tuning the parameters of equation (1) using chaotic maps to improve the inflation rate
Update the Position of universes
Max iteration?
No
Yes
Output the best results
End

Fig. 1 The proposed algorithm (CMVO)

(solution) represents a combination of features. The object in universe is represented as binary value using the following formula:

$$x_i^j = \begin{cases} 1 & \text{if} \quad x_i^j > \sigma \\ 0 & \text{otherwise} \end{cases} \tag{7}$$

where $\sigma \in [0, 1]$ is a random value, the fitness function is taken into concentration the accuracy (*K*NN classifier is used) and the number of selected feature. The fitness function must maximize the classification accuracy and minimize the selected features; therefore, it defined as:

$$\text{Fitness function} = \gamma \times \frac{N_C}{N} + \beta \times \left(1 - \frac{d_s}{d}\right) \tag{8}$$

where γ and β are the weighted factors which have value in [0, 1] to balance between the minimization the number of features and maximization the accuracy of classification and d_s is the number of selected features. d is the total number of features. N_C is the correct number of classified instances. $\frac{N_C}{N}$ is the classification accuracy of *k*NN classifier; it is evaluated after splitting the dataset into training and testing sets using fivefold cross-validation method, where the algorithm runs five times, and at each run, the dataset is split into five classes. One of them is chosen to represent the testing set, and the four classes are used to represent the training set. The accuracy at each run is calculated, and then the average of accuracy for five runs is computed, which represents the final output.

After computing the fitness function for each solution (inflation rate), the solutions are sorted (using Quick sorting algorithm) based on their fitness values. Then the white/black holes model (lines 14–20 in Algorithm 1) is performed using the roulette wheel selection mechanism. The wormhole model is computed (lines 21–32), where the value of r_4 has been selected from the chaotic map. Then the global best function and its corresponding global best solution are computed as in lines 10–11. The previous steps are repeated until stop conditions are satisfied (maximum number of iterations).

Algorithm 1 The Chaotic Multi-Verse Optimizer (CMVO) algorithm

```
 1: Input: Dataset with number of features is $d$, $N$ size of population, $max\_iter$ maximum number of iteration
 2: Output: The $Best\_universe$ that represents the best subset of features.
 3: Initialize $WEP$, $TDR$, and $Best\_universe$
 4: Generate random universes (U)
 5: $SU$ =Sorted universes $U$.
 6: Generate Chaotic map $CM$ with dimension of $max\_iter$
 7: $NI$ =Normalize the fitness of the universes
 8: $t = 1$.
 9: while $t < max\_iter$ do
10:     Evaluate the fitness of all universes
11:     Determine the best fitness function and update $Best\_universe$
12:     for each universe indexed by $i$ do
13:         Update $WEP$ and $TDR$
14:         $Black\_hole\_index = i$;
15:         for each object indexed by $j$ do
16:             $r1 = rand$;
17:             if $r1 < NI(U_i)$ then
18:                 $White\_hole\_index = RouletteWheelSelection(-NI)$;
19:                 $U(Black\_hole\_index, j) = SU(White\_hole\_index, j)$;
20:             end if
21:             $r2 = rand$;
22:             if $r2 < WEP$ then
23:                 $r3 = rand$;
24:                 $r4 = CM(t)$;
25:                 if $r3 < 0.5$ then
26:                     $U(i, j) = Best\_universe(j) + TDR * ((ub(j) - lb(j)) * r4 + lb(j))$;
27:                 else
28:                     $U(i, j) = Best\_universe(j) - TDR * ((ub(j) - lb(j)) * r4 + lb(j))$;
29:                 end if
30:             end if
31:         end for
32:     end for
33:     $t = t + 1$
34: end while
```

Table 2 The datasets used in this study

	Dataset	Sample	Feature	Classes
1	Wisconsin	699	9	2
2	Dermatology	358	34	6
3	DNA	3186	180	3
4	Ion	351	34	2
5	Sonar	208	60	2

4 Experiments and discussion

A comparative analysis has been done between the following algorithms, MVO based on tent chaotic map, MVO based on logistic chaotic map, MVO based on Singer chaotic map, MVO based on sinusoidal chaotic map, MVO based on piecewise chaotic map, the standard MVO, PSO, and ABC. In the rest of this paper, when the name of chaotic map is mentioned separately, it refers to the CMVO based on this map.

Each algorithm has been applied 10 times with random positioning of the search agents. The parameters settings for all algorithms are as follows, numbers of search agents are 25, max iteration is 100, problem dimension is d, and search domain is [01]; as well as PSO: Inertial weight = 1, Inertia weight damping ratio = 0.9, Personal learning coefficient = 1.5, and Global learning coefficient = 2.0. For ABC, a number of limit trials = 5, and in MOV and CMVO algorithms the $\text{WEP}_{\min} = 0.2$ and $\text{WEP}_{\max} = 1$.

4.1 Datasets

The datasets are taken from the UCI data repository [34]. Five datasets are used to validate the performance of the proposed algorithm. Fivefold cross-validation is applied to

Table 3 The number of selected features using eight algorithms

Dataset	MVO	Tent	Logistic	Singer	Sinusoidal	Piecewise	PSO	ABC
Wisconsin	6	6	6	6	6	6	6	6
Dermatology	7	14	11	10	9	10	13	9
DNA	86	88	95	92	99	97	102	94
Ion	15	20	21	20	17	25	17	17
Sonar	39	27	38	31	33	35	36	33

Table 4 The accuracy of all algorithms with different classifiers that represent the correct classification rate

Datasets	Classifiers	Sinusoidal	Piecewise	Singer	Logistic	Tent	MVO	PSO	ABC	ALL
Wisconsin	RF	88	97	93	94	88	91	90	91	96
	J48	88	97	93	94	88	91	90	91	93
	Kstar	93	96	96	94	88	91	90	91	91
	LMT	93	97	96	94	88	91	90	91	97
Dermatology	RF	100	92	92	100	78	92	91	97	100
	J48	100	97	94	94	86	92	86	94	97
	Kstar	97	92	94	100	83	86	89	97	86
	LMT	100	97	94	100	83	94	94	94	100
DNA	RF	84	76	72	92	92	90	92	76	92
	J48	76	76	64	86	92	86	90	74	94
	Kstar	42	64	60	78	73	71	68	68	60
	LMT	86	72	74	92	92	90	90	70	90
Ion	RF	100	100	92	100	96	100	96	87	91
	J48	100	88	92	100	91	100	96	91	91
	Kstar	92	75	88	87	91	91	88	78	83
	LMT	96	96	92	96	96	100	96	87	96
Sonar	RF	77	92	73	85	71	87	92	79	87
	J48	77	69	67	92	86	87	85	79	80
	Kstar	69	100	73	85	71	87	85	93	87
	LMT	77	77	67	77	64	73	85	71	80

split a whole dataset into labeled training set (80% sample) and unlabeled testing set (20% sample). Table 2 summarizes these datasets in details.

4.2 Performance metrics

To compute the performance of the algorithms, four classifiers have been tested and evaluated including Random Forest (RF), J48 decision tree (J48), Kstar, and logistic model tree (LMT).

Also, performance of all algorithms has been evaluated by using different measures of performance, namely accuracy, precision, sensitivity, specificity, NPV, and *F*-measure.

4.2.1 Classification accuracy

The classification accuracy for the experiment is defined as

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{FN} + \text{TN}} \times 100 \tag{9}$$

where TP, TN, FP, and FN are represented the true positive, true negative, false positive, and false negative, respectively.

4.2.2 Sensitivity and specificity

Sensitivity measures the proportion of actual positives which are correctly identified (also called recall).

Table 5 The PSSNF measures of selected features of Wisconsin dataset between CMVO (based on chaotic maps) and the other three algorithms

Algorithms	Classifiers	Precision	Sensitivity	Specificity	NPV	*F*-measure
Sinusoidal	RF	0.92	0.96	0.96	0.92	0.94
	J48	0.92	0.96	0.96	0.92	0.94
	Kstar	0.92	0.96	0.96	0.92	0.94
	LMT	0.92	0.96	0.96	0.92	0.94
Piecewise	RF	0.94	0.93	0.93	0.94	0.94
	J48	0.89	0.86	0.86	0.89	0.88
	Kstar	0.91	0.92	0.92	0.91	0.92
	LMT	0.93	0.93	0.93	0.93	0.93
Singer	RF	0.90	0.95	0.95	0.90	0.92
	J48	0.90	0.95	0.95	0.90	0.92
	Kstrat	0.90	0.95	0.95	0.90	0.92
	LMT	0.90	0.95	0.95	0.90	0.92
Logistic	RF	0.93	0.91	0.91	0.93	0.92
	J48	0.90	0.86	0.86	0.90	0.88
	Kstar	0.93	0.91	0.91	0.93	0.92
	LMT	0.93	0.91	0.91	0.93	0.92
Tent	RF	0.83	0.88	0.88	0.83	0.85
	J48	0.83	0.88	0.88	0.83	0.85
	Kstar	0.83	0.88	0.88	0.83	0.85
	LMT	0.83	0.88	0.88	0.83	0.85
MVO	RF	0.91	0.90	0.90	0.91	0.91
	J48	0.91	0.90	0.90	0.91	0.91
	Kstar	0.93	0.94	0.94	0.93	0.94
	LMT	0.93	0.94	0.94	0.93	0.94
PSO	RF	0.89	0.95	0.95	0.89	0.92
	J48	0.89	0.95	0.95	0.89	0.92
	Kstar	0.89	0.95	0.95	0.89	0.92
	LMT	0.89	0.95	0.95	0.89	0.92
ABC	RF	0.92	0.94	0.94	0.92	0.93
	J48	0.92	0.94	0.94	0.92	0.93
	Kstar	0.92	0.94	0.94	0.92	0.93
	LMT	0.92	0.94	0.94	0.92	0.93

$$\text{Senstivity} = \frac{\text{TP}}{\text{TP} + \text{FN}} \times 100\% \tag{10}$$

Specificity measures the proportion of negatives which are correctly identified.

$$\text{Specificity} = \frac{\text{TN}}{\text{FP} + \text{TN}} \times 100\% \tag{11}$$

4.2.3 Negative predictive value

The negative predictive value (NPV) is the probability that gives a negative result and it is defined as the proportion of subjects with a negative test result who are correctly classified.

$$\text{NPV} = \frac{\text{TN}}{\text{TN} + \text{FN}} \times 100\% \tag{12}$$

A high NPV for a given test means that when the test yields a negative result, it is most likely correct in its assessment.

4.2.4 F-measure

The *F*-measure (also, called *F*-score) is the harmonic mean of both measures recall and precision. It is defined as:

$$F\text{-measure} = \frac{2(\text{precision} + \text{recall})}{\text{precision} \times \text{recall}} \tag{13}$$

Table 6 The PSSNF measures of selected features of Dermatology dataset between CMVO (based on chaotic maps) and the other three algorithms

Algorithms	Classifiers	Precision	Sensitivity	Specificity	NPV	*F*-measure
Sinusoidal	RF	0.93	0.96	0.99	0.99	0.95
	J48	0.93	0.96	0.99	0.99	0.95
	Kstar	0.93	0.96	0.99	0.99	0.95
	LMT	0.93	0.96	0.99	0.99	0.95
Piecewise	RF	0.93	0.96	0.99	0.99	0.95
	J48	0.91	0.93	0.98	0.98	0.92
	Kstar	0.91	0.91	0.98	0.98	0.91
	LMT	0.89	0.90	0.98	0.98	0.89
Singer	RF	0.93	0.96	0.99	0.99	0.95
	J48	0.93	0.96	0.99	0.99	0.95
	Kstar	0.93	0.96	0.99	0.99	0.95
	LMT	0.88	0.88	0.98	0.98	0.88
Logistic	RF	0.86	0.83	0.98	0.98	0.85
	J48	0.89	0.90	0.98	0.98	0.89
	Kstar	0.96	0.98	0.99	0.99	0.97
	LMT	0.89	0.90	0.98	0.98	0.89
Tent	RF	0.76	0.76	0.95	0.95	0.76
	J48	0.82	0.82	0.96	0.96	0.82
	Kstar	0.81	0.83	0.96	0.96	0.82
	LMT	0.83	0.84	0.97	0.97	0.84
MVO	RF	0.96	0.96	0.99	0.99	0.96
	J48	0.95	0.96	0.99	0.99	0.95
	Kstar	0.98	0.98	0.99	0.99	0.98
	LMT	0.93	0.93	0.98	0.98	0.93
PSO	RF	0.91	0.91	0.98	0.98	0.91
	J48	0.93	0.96	0.99	0.99	0.95
	Kstar	0.91	0.91	0.98	0.98	0.91
	LMT	0.88	0.88	0.98	0.98	0.88
ABC	RF	0.97	0.98	0.99	0.99	0.97
	J48	0.97	0.98	0.99	0.99	0.97
	Kstar	0.85	0.94	0.99	0.98	0.89
	LMT	0.97	0.98	0.99	0.99	0.97

Table 7 The PSSNF measures of selected features of DNA dataset between CMVO (based on chaotic maps) and the other three algorithms

Algorithms	Classifiers	Precision	Sensitivity	Specificity	NPV	*F*-measure
Sinusoidal	RF	0.92	0.91	0.96	0.96	0.92
	J48	0.86	0.86	0.93	0.94	0.86
	Kstar	0.75	0.70	0.85	0.86	0.73
	LMT	0.92	0.88	0.94	0.95	0.90
Piecewise	RF	0.92	0.92	0.97	0.97	0.92
	J48	0.92	0.92	0.97	0.97	0.92
	Kstar	0.80	0.75	0.87	0.89	0.77
	LMT	0.92	0.92	0.97	0.97	0.92
Singer	RF	0.79	0.85	0.91	0.89	0.82
	J48	0.83	0.85	0.93	0.92	0.84
	Kstar	0.80	0.76	0.88	0.89	0.78
	LMT	0.83	0.86	0.91	0.91	0.85
Logistic	RF	0.88	0.89	0.96	0.95	0.89
	J48	0.88	0.88	0.95	0.95	0.88
	Kstar	0.74	0.71	0.86	0.87	0.72
	LMT	0.92	0.91	0.96	0.96	0.91
Tent	RF	0.87	0.94	0.96	0.94	0.91
	J48	0.89	0.89	0.96	0.95	0.89
	Kstar	0.75	0.76	0.88	0.87	0.76
	LMT	0.91	0.91	0.96	0.96	0.91
MVO	RF	0.90	0.90	0.95	0.95	0.90
	J48	0.86	0.85	0.92	0.92	0.86
	Kstar	0.66	0.62	0.81	0.83	0.64
	LMT	0.89	0.88	0.94	0.94	0.88
PSO	RF	0.95	0.97	0.98	0.98	0.96
	J48	0.91	0.90	0.96	0.96	0.91
	Kstar	0.80	0.76	0.89	0.90	0.78
	LMT	0.99	0.98	0.99	0.99	0.99
ABC	RF	0.89	0.91	0.96	0.95	0.90
	J48	0.93	0.88	0.94	0.96	0.90
	Kstar	0.67	0.65	0.82	0.82	0.66
	LMT	0.92	0.91	0.96	0.96	0.91

4.3 Results and discussion

The results of the CMVO algorithm based on different five maps compared to other swarm algorithms are given in Tables 3, 4, 5, 6, 7, 8, 9 and 10 and Figs. 2, 3, 4 and 5 (All in tables and figures points to the classification based all features). Table 3 illustrates the number of selected features using different algorithms, and Tables 4, 5, 6, 7, 8 and 9 demonstrate the performance of different classifiers for each dataset based on the features selected.

Tables 4 and 5 as well as Figs. 2 and 3 show the average of classification rate and the average of precision, sensitivity, specificity, *F*-measure, and NPV measures for Wisconsin dataset employing swarm algorithms and CMVO algorithm with different classifiers. (Note in the following we will refer to these measures precision, sensitivity, specificity, NPV, and *F*-measure by PSSNF measures).

The accuracy of piecewise algorithm is the best (97%), followed by the Singer and logistic which have the same accuracy like if all features are used (94%). The MVO and ABC algorithms are (same accuracy 91%) best than the rest of algorithms (sinusoidal is 90%, PSO is 90%, and Tent is 88%). Also, from Table 5 we can conclude that the higher performance is the sinusoidal map (based on term PSSNF, in general, has 94%), the ABC algorithm is the second best with accuracy 93%. Then the rest algorithms have the nearly the same performance expect the Tent algorithm is less than all algorithms.

Table 8 The PSSNF measures of selected features of Ion dataset between CMVO (based on chaotic maps) and the other three algorithms

Algorithms	Classifiers	Precision	Sensitivity	Specificity	NPV	*F*-measure
Sinusoidal	RF	1.00	1.00	1.00	1.00	1.00
	J48	1.00	1.00	1.00	1.00	1.00
	Kstar	0.89	0.94	0.94	0.89	0.91
	LMT	0.94	0.97	0.97	0.94	0.96
Piecewise	RF	1.00	1.00	1.00	1.00	1.00
	J48	0.86	0.88	0.88	0.86	0.87
	Kstar	0.69	0.77	0.77	0.69	0.73
	LMT	0.94	0.97	0.97	0.94	0.96
Singer	RF	0.91	0.91	0.91	0.91	0.91
	J48	0.91	0.91	0.91	0.91	0.91
	Kstar	0.83	0.92	0.92	0.83	0.87
	LMT	0.91	0.91	0.91	0.91	0.91
Logistic	RF	1.00	1.00	1.00	1.00	1.00
	J48	1.00	1.00	1.00	1.00	1.00
	Kstar	0.81	0.92	0.92	0.81	0.86
	LMT	0.94	0.97	0.97	0.94	0.95
Tent	RF	0.94	0.97	0.97	0.94	0.95
	J48	0.88	0.94	0.94	0.88	0.91
	Kstar	0.88	0.94	0.94	0.88	0.91
	LMT	0.94	0.97	0.97	0.94	0.95
MVO	RF	1.00	1.00	1.00	1.00	1.00
	J48	1.00	1.00	1.00	1.00	1.00
	Kstar	0.88	0.94	0.94	0.88	0.91
	LMT	1.00	1.00	1.00	1.00	1.00
PSO	RF	0.94	0.97	0.97	0.94	0.96
	J48	0.94	0.97	0.97	0.94	0.96
	Kstar	0.83	0.92	0.92	0.83	0.87
	LMT	0.94	0.97	0.97	0.94	0.96
ABC	RF	0.84	0.87	0.87	0.84	0.85
	J48	0.88	0.94	0.94	0.88	0.91
	Kstar	0.72	0.79	0.79	0.72	0.75
	LMT	0.84	0.87	0.87	0.84	0.85

The results for the Dermatology dataset are shown in Tables 4 and 6, as well as Figs. 2 and 4. From these results, we can observe that the sinusoidal and logistic algorithms yield a better accuracy of 99% (with nine and eleven features respectively). Also, the ABC algorithm has higher performance than Piecewise, Singer, The standard MVO, PSO, and Tent algorithms(94, 94, 91, 90 and 83% respectively). From Table 6 the ABC algorithm and the standard MVO have the best results (overall measures); however, there is a small difference between them and sinusoidal (96%). Then the Singer is better than all other algorithms, namely piecewise, logistic, tent and PSO algorithms (94, 92, 85 and 93% respectively).

Tables 4 and 7 as well as Figs. 2 and 5 illustrate the performance of algorithms for DNA dataset. The logistic and tent algorithms yield a better accuracy of 87% (with 95 and 88 features respectively). Also, the PSO, MVO algorithms and if all features are selected are the same results (85%), and their performance is higher than the rest of algorithms. From Table 6, the PSO algorithm has the best results (in term PSSNF) 93%, followed by the piecewise (91%). Then the tent (with accuracy 89%) is higher than sinusoidal and logistic (have nearly performance 88%). Then ABC (87%) is better than MVO and Singer (86%). Also, This table illustrates that the best algorithm in the term of *F*-

Table 9 The PSSNF measures of selected features of Sonar dataset between CMVO (based on chaotic maps) and the other three algorithms

Algorithms	Classifiers	Precision	Sensitivity	Specificity	NPV	*F*-measure
Sinusoidal	RF	0.76	0.78	0.78	0.76	0.77
	J48	0.76	0.78	0.78	0.76	0.77
	Kstar	0.68	0.71	0.71	0.68	0.69
	LMT	0.76	0.78	0.78	0.76	0.77
Piecewise	RF	0.93	0.93	0.93	0.93	0.93
	J48	0.69	0.69	0.69	0.69	0.69
	Kstar	1.00	1.00	1.00	1.00	1.00
	LMT	0.79	0.83	0.83	0.79	0.81
Singer	RF	0.72	0.75	0.75	0.72	0.74
	J48	0.66	0.67	0.67	0.66	0.66
	Kstar	0.73	0.73	0.73	0.73	0.73
	LMT	0.66	0.67	0.67	0.66	0.66
Logistic	RF	0.83	0.89	0.89	0.83	0.86
	J48	0.92	0.94	0.94	0.92	0.93
	Kstar	0.85	0.85	0.85	0.85	0.85
	LMT	0.76	0.78	0.78	0.76	0.77
Tent	RF	0.71	0.73	0.73	0.71	0.72
	J48	0.86	0.89	0.89	0.86	0.87
	Kstar	0.71	0.71	0.71	0.71	0.71
	LMT	0.64	0.65	0.65	0.64	0.64
MVO	RF	0.86	0.90	0.90	0.86	0.88
	J48	0.86	0.90	0.90	0.86	0.88
	Kstar	0.87	0.87	0.87	0.87	0.87
	LMT	0.73	0.73	0.73	0.73	0.73
PSO	RF	0.93	0.93	0.93	0.93	0.93
	J48	0.86	0.88	0.88	0.86	0.87
	Kstar	0.83	0.89	0.89	0.83	0.86
	LMT	0.86	0.88	0.88	0.86	0.87
ABC	RF	0.79	0.79	0.79	0.79	0.79
	J48	0.79	0.79	0.79	0.79	0.79
	Kstar	0.93	0.94	0.94	0.93	0.93
	LMT	0.71	0.71	0.71	0.71	0.71

Table 10 Comparisons using LSD test between CMVO-Logistic and all other algorithms

	Sinusoidal	Piecewise	Singer	Tent	MVO	PSO	ABC	ALL
Mean Difference (I–J)	4.95	4.33	7.83*	10.37*	5.42	5.71*	7.25*	2.62
p value	0.085	0.132	0.007	0.000	0.060	0.048	0.012	0.361

* The mean difference is significant at the 0.05 level

measure is the PSO followed by the CMVO based on most chaotic maps expect singer map that less than ABC algorithm.

Tables 4 and 8 as well as Figs. 2 and 6 give the results for Ion dataset. The results in these tables are established the performance of the MVO algorithm which has the better accuracy (98% with 15 features) followed by the sinusoidal (97% with 17 features). The performance of logistic is better than the rest of the algorithms with accuracy 96%, also the remaining algorithms have close performance expect ABC algorithm that has 86%. From Table 8 and Fig. 6, we can reach to the same results above, in which the best one is MVO the sinusoidal. These algorithms have the better performance in term of *F*-measure and all other performances (Precision, Sensitivity, Specificity, and NPV).

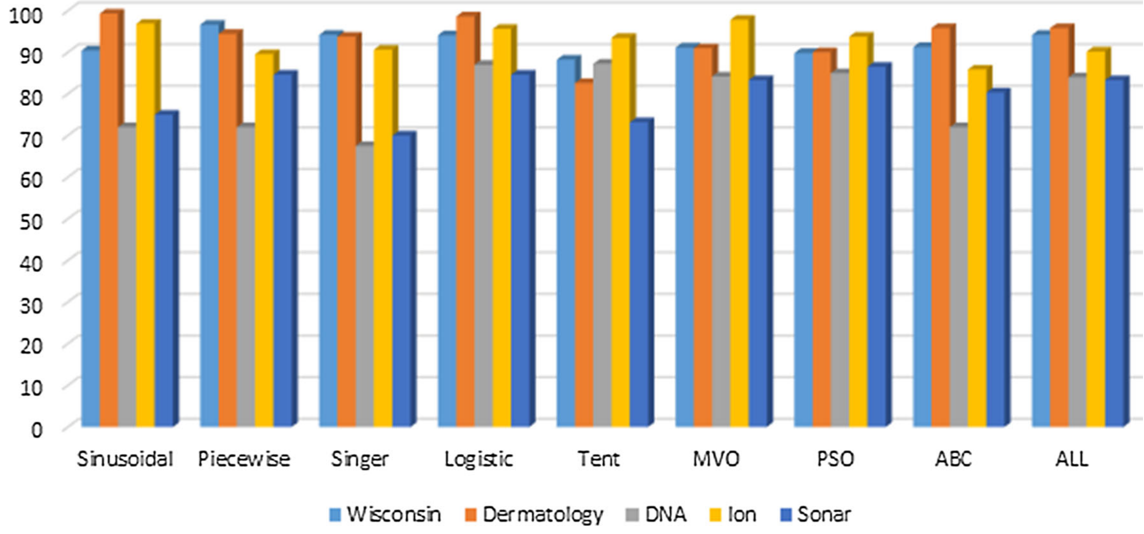

Fig. 2 The average of accuracy over all classifiers for each dataset using different feature selection algorithm

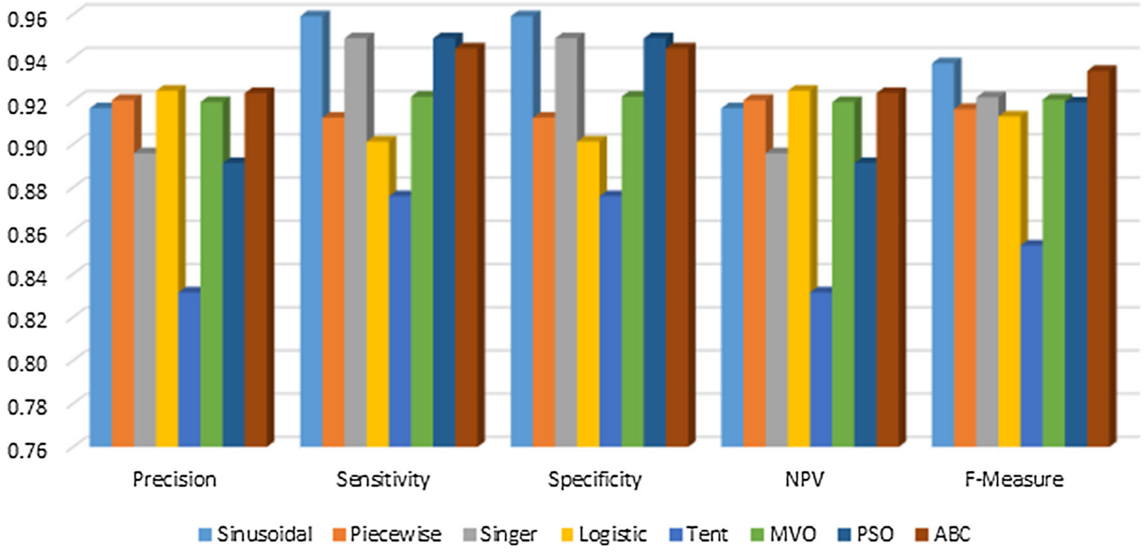

Fig. 3 The average of each PSSNF measure for Wisconsin dataset

The results for the Sonar dataset are given in Tables 4 and 9 as well as Figs. 2 and 7. We can conclude from these results that the PSO algorithm has the higher performance overall algorithms in all terms, then both piecewise and the logistic are in the second rank with accuracy 85%, and over the PSSNF measures, they have close values. Also, the MVO is better than the rest of algorithms (tent, sinusoidal, Singer, and ABC with accuracy 73, 75, 70, and 80%, respectively).

Finally, from all previous results, to determine the based feature selection algorithm we compute the mean accuracy over all datasets as in Fig. 8. From this figure, we can conclude that the best algorithm is the logistic (with accuracy 92%), followed by the piecewise. The sinusoidal, the standard MVO, and PSO algorithms are have the close value of accuracy 89%; however, this accuracy value is less than if all features are used. Also, the piecewise and sinusoidal algorithms have the same accuracy 87%, followed by the ABC and tent algorithms (with accuracy 85%) that are better than Singer algorithm (83%).

Moreover, for the purpose of illustration, Fig. 9 illustrates the boxplots representing the accuracy averages of all algorithms using all datasets. It is evident from Fig. 9 that the logistic algorithm is located at the upper side of the figure, which refers to the high accuracy scores of its results than the results of other algorithms.

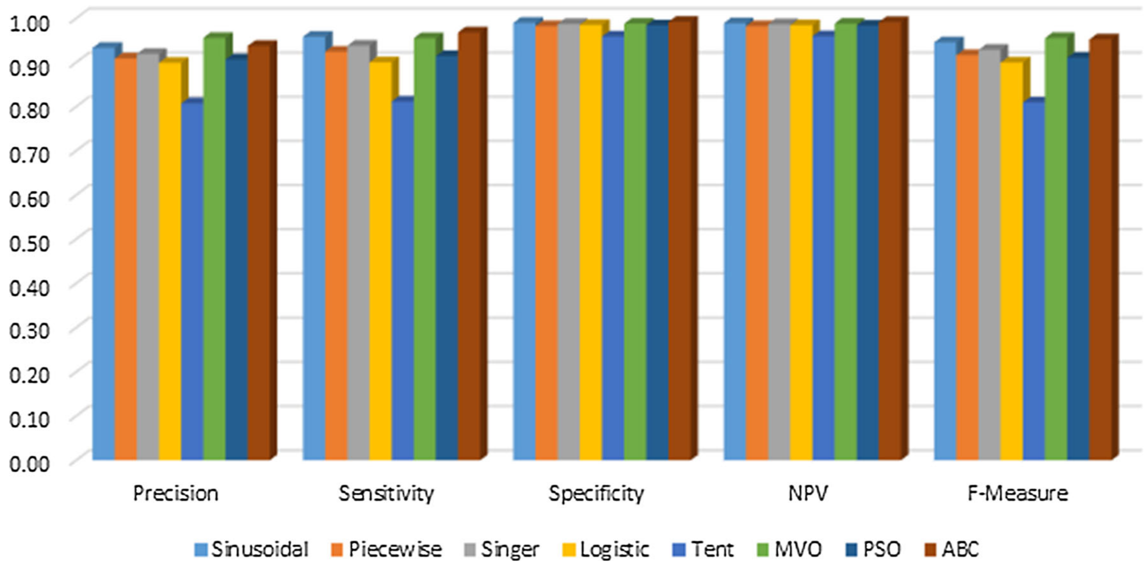

Fig. 4 The average of each PSSNF measure for Dermatology dataset

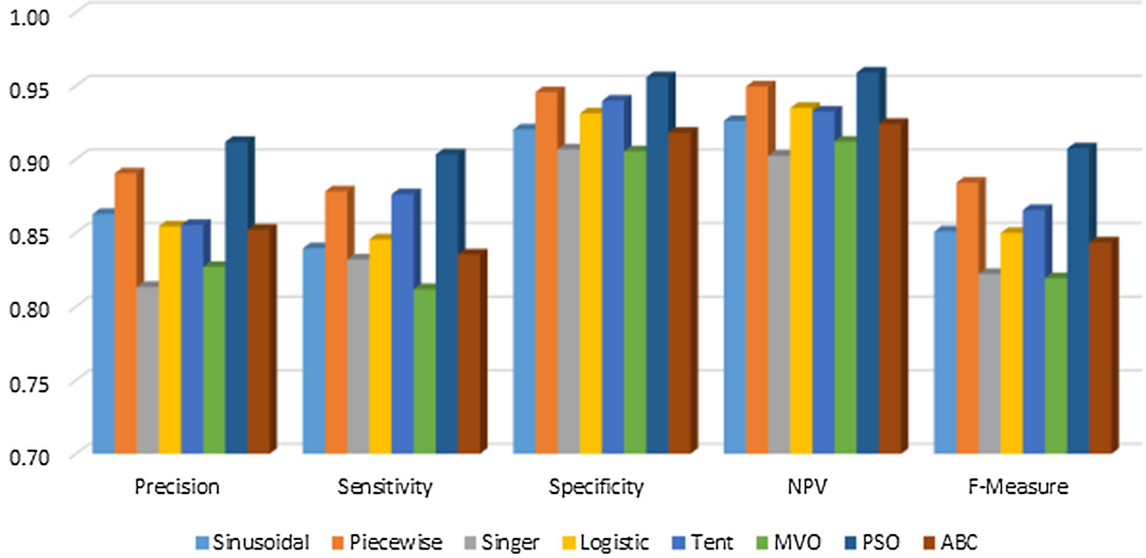

Fig. 5 The average of each PSSNF measure for DNA dataset

4.4 ANOVA analysis

To further analysis the previous results, the analysis of variance (ANOVA) test is used. So far, we have examined the means of a set of measures using ten runs to mean of the accuracy showing that some differences can be found among them. We used the ANOVA test with post hoc LSD test to further statistically compare all the algorithms over their mean accuracies on all datasets. Here, the null hypothesis is that all algorithms are equivalent in term of accuracy rate. The ANOVA gives a statistical value called *p* value, if this value is smaller than the significance level ($\alpha = 0.05$) we can say that they are significantly different and we reject the null hypothesis. The LSD test is used after the null hypothesis is rejected and this test is used to determine the difference among the algorithms.

The *p* value, in our experiment, was 0.028, and this value was less than α; therefore, we rejected the null hypothesis and used the LSD test in order to determine whether there is exists a significant difference between the logistic map and other algorithms. Table 10 shows the LSD result, and we can observe from this table that these results are consistent with Fig. 8. Also, there is no significant difference between the accuracy of logistic, the sinusoidal, and piecewise, also, no significant difference with classification using all features.

However, there is a significant difference between algorithm logistic and four algorithms, namely tent, singer,

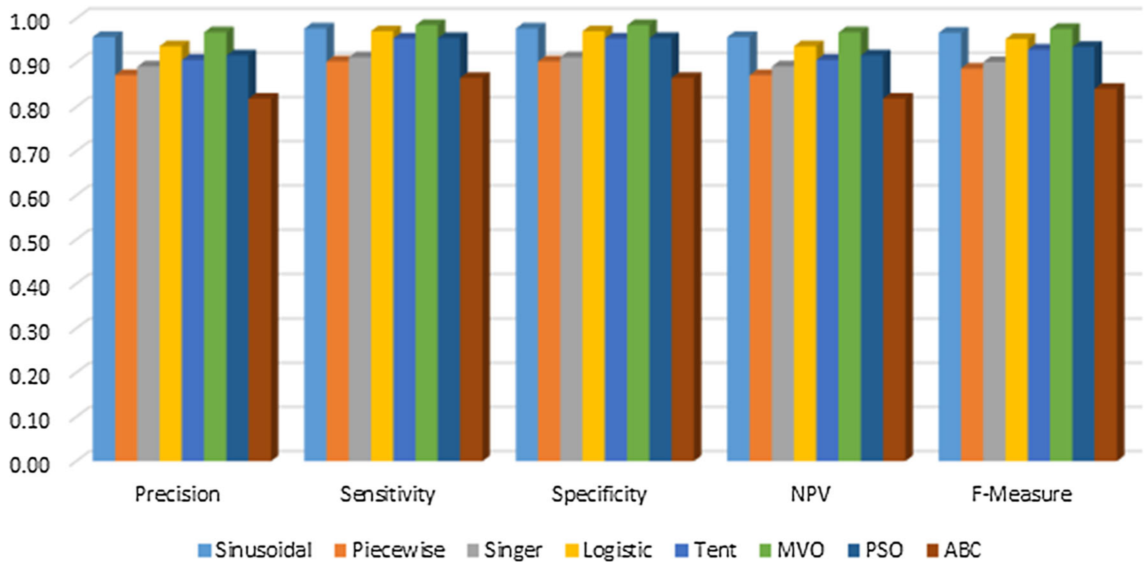

Fig. 6 The average of each PSSNF measure for Ion dataset

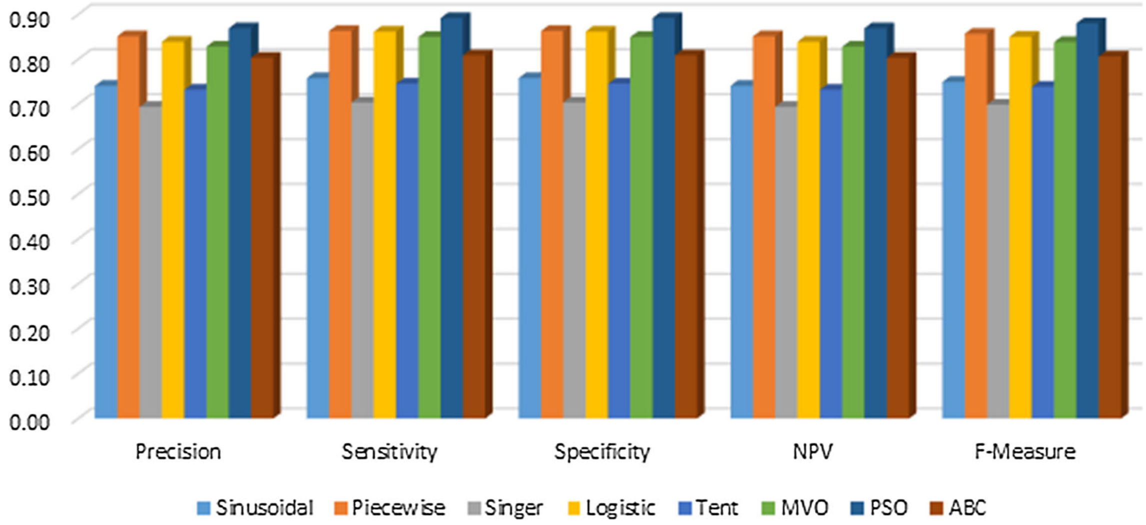

Fig. 7 The average of each PSSNF measure for Sonar dataset

PSO, and ABC. We can conclude that the logistic is a much better feature selection algorithm.

5 Conclusions and future works

In this paper, a novel optimization algorithm based on chaotic and multi-verse optimizer (CMVO) is proposed using five chaotic maps for global optimization of feature selection. The characteristics of the chaotic systems such as regularity and semi-stochastic are used to improve the performance of the MVO algorithm. The CMVO algorithm based on five chaotic maps is tested using five benchmark datasets collected from UCI repository, in which the performance of CMVO algorithm is compared with standard MVO and two other swarm algorithms namely, PSO and ABC. The experimental results showed that tuned MVO with chaotic maps increases the performance of classification rate with minimizing the number of selected features. From the results, we can conclude that the CMVO-based logistic map is better feature selection algorithm compared to all other algorithms; also, in general, chaotic maps increase the performance of MVO, for example, CMVO based on sinusoidal and piecewise are better than ABC and tent algorithms. In the worst results of CMVO-based Singer map (with 83%), its performance nearly equal

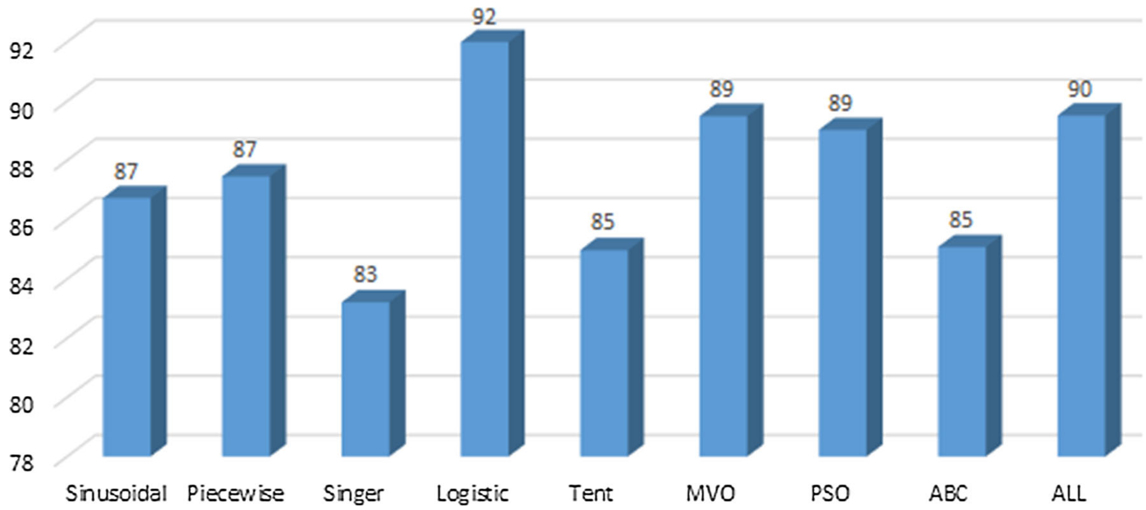

Fig. 8 The accuracy over all algorithms

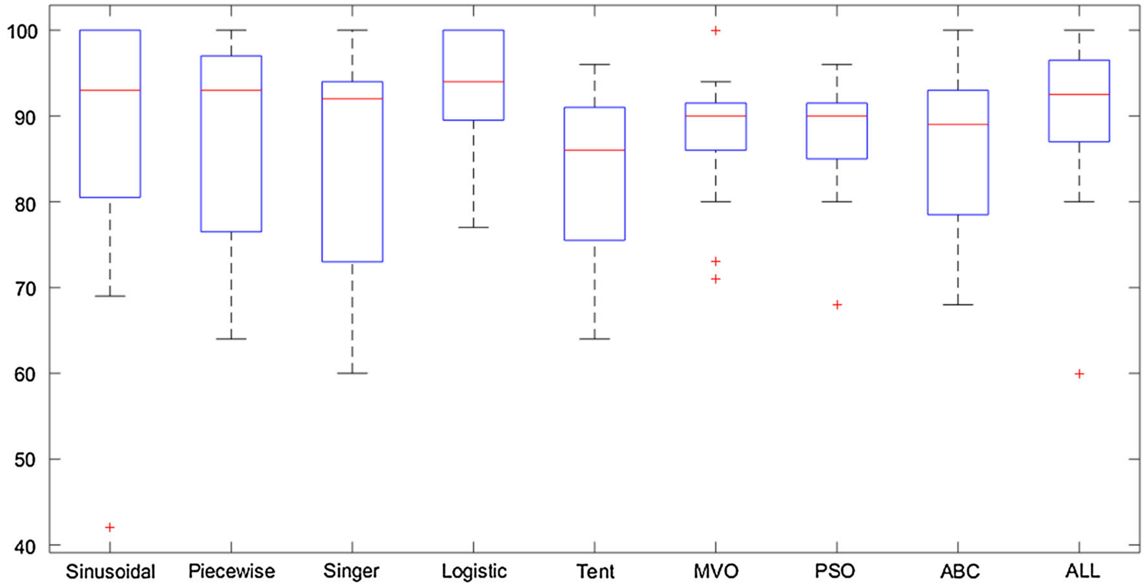

Fig. 9 The boxplots of accuracy of all algorithms

to ABC algorithm. In future, we will improve the MVO algorithm by applying more chaotic maps to different applications.

Compliance with ethical standards

Conflict of interest The authors state that there are no conflicts of interest, and this study was carried out without any funding sources.

References

1. Hancer E, Xue B, Karaboga D, Zhang M (2015) A binary ABC algorithm based on advanced similarity scheme for feature selection. Appl Soft Comput 36:334–348
2. Esmel ME (2011) A novel image retrieval model based on the most relevant features. Knowl Based Syst 24(1):23–32
3. Yousef M, Saçar Demirci MD, Khalifa W, Allmer J (2016) Feature selection has a large impact on one-class classification accuracy for micrornas in plants. Adv Bioinform 2016:5670851. doi:10.1155/2016/5670851
4. Zawbaa HM, Emary E, Grosan C (2016) Feature selection via chaotic antlion optimization. PloS One 11(3):e0150652
5. Espinosa HEP, Ayala-Solares JR (2016) The power of natural inspiration in control systems. Nat Inspir Comput Control Syst 40:1–10
6. Eberhart R, Kennedy J (1995) A new optimizer using particle swarm theory. In: Proceedings of the Sixth International Symposium on Micro Machine and Human Science, MHS'95., vol 1. New York, IEEE, pp 39–43
7. Basturk B, Karaboga D (2006) An artificial bee colony (ABC) algorithm for numeric function optimization. In: IEEE swarm intelligence symposium, vol 8. pp 687–697

8. Mirjalili S, Mirjalili SM, Lewis A (2014) Grey wolf optimizer. Adv Eng Softw 69:46–61
9. Moradi P, Rostami M (2015) Integration of graph clustering with ant colony optimization for feature selection. Knowl Based Syst 84:144–161
10. Mirjalili S (2015) The ant lion optimizer. Adv Eng Softw 83:80–98
11. Zhang Y, Gong D, Hu Y, Zhang W (2015) Feature selection algorithm based on bare bones particle swarm optimization. Neurocomputing 148:150–157
12. El Aziz MA, Hassanien AE (2016) Modified cuckoo search algorithm with rough sets for feature selection. Neural Comput Appl. doi:10.1007/s00521-016-2473-7
13. Emary E, Zawbaa HM, Hassanien AE (2016) Binary ant lion approaches for feature selection. Neurocomputing 213:54–65
14. Anter AM, Hassanien AE, ElSoud MA, Kim T-H (2015) Feature selection approach based on social spider algorithm: case study on abdominal ct liver tumor. In: 2015 Seventh International Conference on Advanced Communication and Networking (ACN). IEEE, pp 89–94
15. Yamany W, Emary E, Hassanien AE (2015) New rough set attribute reduction algorithm based on grey wolf optimization. In: 1st International Conference on Advanced Intelligent System and Informatics (AISI2015), Springer, Egypt, pp 241–251
16. Nakamura RYM, Pereira LAM, Costa KA, Rodrigues D, Papa JP, Yang X-S (2012) BBA—a binary bat algorithm for feature selection. In: 2012 25th SIBGRAPI Conference on Graphics, Patterns and Images. IEEE, pp 291–297
17. Jiang S, Yang S (2017) A steady-state and generational evolutionary algorithm for dynamic multiobjective optimization. IEEE Trans Evolut Comput 21(1):65–82
18. El Aziz MA, Ewees AA, Hassanien AE (2017) Whale Optimization Algorithm and Moth-Flame Optimization for multilevel thresholding image segmentation. Exp Syst Appl 83:242–256
19. Sindhu R, Ngadiran R, Yacob YM, Zahri NAH, Hariharan M (2017) Sine–cosine algorithm for feature selection with elitism strategy and new updating mechanism. Neural Comput Appl. doi:10.1007/s00521-017-2837-7
20. Zhou Z, Zhu S, Zhang D (2015) A novel K-harmonic means clustering based on enhanced firefly algorithm. In: International Conference on Intelligent Science and Big Data Engineering. Springer International Publishing, pp 140–149
21. Wang G-G, Guo L, Gandomi AH, Hao G-S, Wang H (2014) Chaotic krill herd algorithm. Inf Sci 274:17–34
22. Mitić M, Vuković N, Petrović M, Miljković Z (2015) Chaotic fruit fly optimization algorithm. Knowl Based Syst 89:446–458
23. Yu F, Li W, Tao J, Deng K, Ma L, He F (2017) Estimation of distribution algorithm combined with chaotic sequence for dynamic optimisation problems. Int J Comput Sci Math 8(1):12–19
24. Adarsh BR, Raghunathan T, Jayabarathi T, Yang X-S (2016) Economic dispatch using chaotic bat algorithm. Energy 96:666–675
25. Gandomi AH, Yang X-S, Talatahari S, Alavi AH (2013) Firefly algorithm with chaos. Commun Nonlinear Sci Numer Simul 18(1):89–98
26. Saremi S, Mirjalili S, Lewis A (2014) Biogeography-based optimisation with chaos. Neural Comput Appl 25(5):1077–1097
27. Gandomi AH, Yang X-S (2014) Chaotic bat algorithm. J Comput Sci 5(2):224–232
28. Chuang L-Y, Yang C-H, Li J-C (2011) Chaotic maps based on binary particle swarm optimization for feature selection. Appl Soft Comput 11(1):239–248
29. Li M, Du W, Yuan L (2010) Feature selection of face recognition based on improved chaos genetic algorithm. In: 2010 Third International Symposium on Electronic Commerce and Security (ISECS). IEEE, pp 74–78
30. Mirjalili S, Mirjalili SM, Hatamlou A (2016) Multi-verse optimizer: a nature-inspired algorithm for global optimization. Neural Comput Appl 27(2):495–513
31. Ellis GFR (2011) Does the multiverse really exist? Sci Am 305(2):38–43
32. Ning S-L, Wen-Biao Liu (2016) Black hole phase transition in massive gravity. Int J Theor Phys 55(7):3251–3259
33. Ren B, Zhong W (2011) Multi-objective optimization using chaos based PSO. Inf Technol J 10(10):1908–1916
34. Bache K, Lichman M (2013) UCI machine learning repository. University of California, Irvine, School of Information and Computer Sciences. http://archive.ics.uci.edu/ml. Accessed 3 Jan 2017