Two-Steps Wrapper-Based Feature Selection in Classification: A Comparison Between Continuous and Binary Variants of Cuckoo Optimisation Algorithm

Ali Muhammad Usman, Umi Kalsom Yusof, and Syibrah Naim

Abstract Feature selection (FS) is the process of eliminating irrelevant features and improving classification performance while maintaining the standard of the data. Based on evaluation criteria, FS can be either filter or wrapper. Wrappers are computationally expensive due to many feature interactions in the search space. In this study, cuckoo optimisation algorithm (COA) along with its binary (BCOA) are used as wrapper-based FS for the first time to explore the promising regions in the search space, which obtained improved classification accuracy and selected better quality subsets of features within a shorter time. Based on that we developed two different fitness functions. The first one (BCOA-FS) and (BCOA-FS) adopt the standard wrapper-based evaluation with emphasis mainly on the classification performance. Whereas in the second one (BCOA-2S) and (COA2S) combine the first one in another evaluation process with a focus on both the number of features and classification accuracy. The results obtained indicate that COA-FS and BCOA-FS can select fewer features with better accuracy on both categorical and continuous label data, with BCOA-FS better than COA-FS. Similarly, COA-2S performed better than BCOA-FS and COA-2S and is comparable to the existing works. BCOA-2S outperformed the three of the existing studies on the majority of the datasets with almost 10 and 5% on both classification accuracy and number of selected features, respectively.

Keywords Feature selection · Wrapper-based · Cuckoo optimisation algorithm · Binary Cuckoo optimisation algorithm and Classification

A. M. Usman · U. K. Yusof (✉)
School of Computer Sciences, Universiti Sains Malaysia, 11800 Pulau Penang, Malaysia
e-mail: umiyusof@usm.my

A. M. Usman
e-mail: alimuhammad@fcetgombe.edu.ngs

A. M. Usman
Department of Computer Sciences, Federal College of Education (Technical), P.M.B 60, Gombe, Nigeria

S. Naim
Technology Department, Endicott College of International Studies (ECIS), Woosong University, Daejeon, Korea
e-mail: syibrah@wsu.ac.kr

H. Chiroma et al. (eds.), *Machine Learning and Data Mining for Emerging Trend in Cyber Dynamics*, https://doi.org/10.1007/978-3-030-66288-2_6

1 Introduction

In classification, feature selection (FS) is mainly used to minimise the features in a data set while maintaining its standard [23, 68]. The aim is to select the most relevant subsets that are sufficient enough to describe the target class [62]. FS can be supervised [51], non-supervised [12] and semi-supervised [61]. In dealing with the supervised type, the class label of the data set is defined already in contrast with the non-supervised that the class label is unaware. Whereas, semi-supervised combine both supervised and non-supervised (i.e. with both label and unlabelled data) [32].

The supervised FS is classified further as a filter, wrapper and hybrid approach depending on their evaluation criteria [64]. In the filter-based approach, the features are evaluated without considering any classification algorithm, which makes them computationally fast. However, the filter ignores feature dependence or relationship among selected or ranked features, which subsequently affects the classification performance (i.e. either error rate or accuracy) [15].

In the wrapper-based method, a classification algorithm is used to evaluate each subset of features selected, and hence, it achieves better classification accuracy or error rate [26, 32, 64]. The major shortcomings of the wrapper-based approach are computationally expensive and not favourable on high-dimensional data sets. The most common examples of the wrapper-based approach are sequential forward selection (SFS) [60], sequential backward selection (SBS) [36] plus q take away r [16] and genetic algorithm (GA) [37] among others.

In FS, the classification performance of any of the approach (filter or wrapper) is measured in according to feature size (number of selected features), error rate (or accuracy) and computational time. Machine learning algorithms are commonly used to measure or evaluate the goodness of the selected subset of features in terms of accuracy or error rates [64]. Examples of the most widely used ones include support vector machine (SVM) [57], K-nearest neighbour (KNN) [24] and Gaussian Naïve Bayes (GNB) [34] among others.

The wrapper-based approach requires one to determine a classification algorithm and uses its performance as the evaluation standard. It searches for features that are suitable to the machine learning algorithm that increases the accuracy [39, 40]. Classification accuracy, as well as the selected subsets of features, are used to determine the prediction performance in wrapper model [54, 66]. As such, prediction accuracy and less prone to local optima are the critical advantage of wrapper over the filter. Hence, the outcomes are mostly more encouraging than the findings of the filter models.

However, the shortcomings of the wrapper-based approach include a high risk of overfitting data, classifier dependency, highly computationally intensive and not favourable for astronomical dimensional data [41]. Examples of wrapper techniques are a sequential forward selection (SFS), sequential backward elimination (SBE), plus q take away r and beam search. Others include simulated annealing, randomised hill-climbing, genetic algorithm and estimation of distribution algorithm among others [40, 46]—the steps on how the wrapper-based approach is illustrated in Fig. 1.

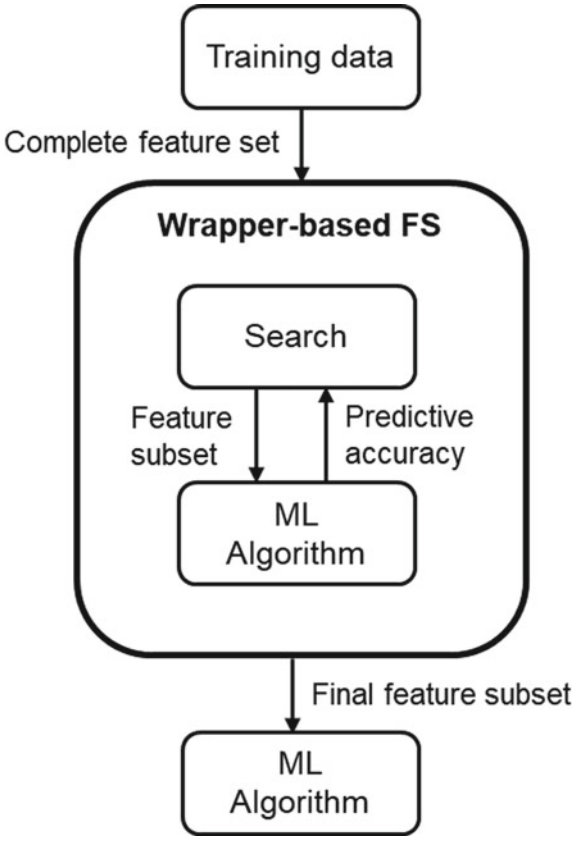

Fig. 1 Wrapper-Based feature selection procedure

Finding the optimal subsets of features with the less computational cost is quite a demanding task because of the search space, and the number of features needed to search in the solutions are too many. Hence, FS is considered as an NP-hard problem [33, 39, 52, 64]. In searching for the best subsets of features, [26, 39] identified three search strategies: complete search, sequential or heuristic search and the random search.

The complete search works by finding all the possible feature subsets while evaluating them one after the other to select the best subset of features with the highest classification performance. There is a guarantee of getting the optimal results based on the laid down criteria in it. However, for a data set with N number of features, there will be 2^N subsets to be generated and evaluated, which is almost impractical for a considerable value of N. More so, [41] argued that 'search is complete does not mean that it must be exhaustive'.

The heuristic or sequential search add or remove features in sequential order; the remaining features that are not selected are considered later for selection in that manner. By doing so, the choice of the features may likely end up with the same pattern as complete search. However, there is no guarantee of finding the target solution [52].

Random search is the most popularly used among all the search strategies [42]. It aims at creating stability between the heuristic and complete search by combining the advantages of both. It begins with a randomly selected subset of features and progress in two ways. Either to follow the heuristic search and insert some randomness or to generate the next subset in a completely random manner [39, 42].

Out of all these search strategies, the random search is the only one that can escape from local optimum in the vast search space due to the involvement of randomness and mostly finish within the shortest time [26, 40, 64].

There are some works on the wrapper-based FS that applies different search strategies on different meta-heuristic algorithms. For example, [9, 10] used an artificial immune system for the FS. Also a particle swarm optimisation (PSO) is reported in [31, 44, 52–54, 63, 65].

Recently, differential evolution (DE) in [27], cuckoo search in [13, 18], grasshopper optimisation algorithm in [43], genetic algorithm (GA) in [3, 19, 30, 59, 66] are reported. In addition to that, genetic programming is used for the wrapper-based FS in [48], and recently a flower pollination algorithm for FS as also in [56]. It clearly shows that the meta-heuristic algorithms are suitable for addressing these problems. Despite the attempt to solve the lingering issues of the wrapper-based FS still, the existing works cannot successfully evolve the best subset of features with improving accuracy on some of the data sets [23].

The cuckoo optimisation algorithm (COA) presented in [49] is among the evolutionary algorithms that show promising results in handling different combinatorial optimisation problem including NP-hard, despite its proven records, especially in dealing with filter-based FS in [55]. Its application, specifically for the wrapper-based FS, is not fully investigated.

This study aimed to find the best subsets of features with lesser feature size and yet maintain the same or even better classification accuracy compared to using full-length features within a short period. Also, investigate the difference between COA and BCOA in wrapper-based FS.

To accomplish this goal, a pair of two FS frameworks are developed based on BCOA and COA. These proposed algorithms were studied and compared with other FS algorithms presented in other works on benchmark problems of varying difficulties.

Precisely, this study will examine

1. Whether adopted COA wrapper-based FS algorithms would choose the best subsets of feature, that has least feature size, less computational and accomplish the best error rate compared to full-length features, and would outpace the adopted BCOA wrapper-based single objective algorithms;
2. Whether adopted BCOA wrapper-based FS algorithms would choose the best subsets of feature and can attain the best performance than the adopted COA wrapper-based algorithms above;
3. Whether COA wrapper-based algorithm with two steps evaluation would choose sets of best features subsets and would outpace the two steps BCOA wrapper-based algorithm, and other existing works; and

4. Whether BCAO wrapper-based algorithm with two steps evaluation would choose sets of best features subsets and would outpace all other approaches stated directly above.

The rest of the paper is prearranged as follows: Part 2 is the background containing the details about the adopted COA and BCOA along with related works. The proposed wrapper-based feature selection approaches are presented in Part 3, while Part 4 is the experimental design, data sets used along with benchmark approaches. Then Part 5 is the presentation of the results while Part 6 concludes the entire work and suggests future work areas.

2 Background

2.1 Cuckoo Optimisation Algorithm

The original Cuckoo Optimisation Algorithm (COA) is strictly made for a continuous optimisation problem. At the same time, the binary version (BCOA) can be applied to solve problems that are in binary or discrete form. COA used for FS is very scarce in the literature. The size or dimension of the search space (i.e. the full-length features in every data set) is n. Every habitat in the COA is assigned by using a vector of n decimal numbers. The location of habitat i in dth length is xid normally in the range [0, 1]. To know in case if a feature is selected or otherwise, a verge $0 < \theta < 1$ is mandatory to equate it with the decimal numbers in the habitat position. If eventually, $xid > \theta$, then feature d is chosen else d is not be chosen.

COA developed by [49] is adopted, and the detail of how it works is

1. An array called "habitat" is used for the optimisation problem as show in Eq. 1.

$$habitat = [x_1, x_2, \ldots, x_{Nvar}] \tag{1}$$

2. Five and twenty eggs are used as the lower and upper limits, respectively, for every iteration.
3. They lay their eggs within a maximum range distance from their habitat in Equation.

$$ELR = \alpha \times \frac{number\ of\ current\ cuckoos}{total\ number\ of\ eggs} \times e_{new} \tag{2}$$

An α represent an integer number.

4. P% (those without any profit value) of the laid eggs are killed.
5. A k-means (K = 3 or 5) clustering is used for the grouping.
6. All cuckoos deviate φ radians while flying $\lambda\%$ to the goal, as shown in Eq. 3.

$$\lambda \sim U(0, 1) \qquad \varphi \sim (-\omega, \omega) \tag{3}$$

where $\lambda\ U(0,\ 1)$ means that λ is a uniformly distributed random within range of 0 and 1. ω is limits an aberration from goal habitat.

2.2 Binary Cuckoo Optimisation Algorithm

Binary Cuckoo Optimisation Algorithm (BCOA) is mostly used to solve FS problem; meanwhile, the representation of the habitat is in the form of a binary string, where the position of every habitat is a boolean 1 which signifies that a feature is chosen and 0 otherwise. Assuming X_G and X_C represent the respective goal and current habitat. Then, Eq. 4 computes the X_{NH} next habitat as follows:

$$X_{NH} = X_C + rand(X_G - X_C) \tag{4}$$

A sigmoid function is applied in Eq. 5 to use X_{NH} as binary to record it within [0, 1]. Then Eq. 6 alters the values to either 0 or 1.

$$S = \frac{1}{(1 + e^{-X_{NH}})} \tag{5}$$

$$IF\ (S > rand)\ THEN\ X_{NH} = 1\ AND\ IF\ (S < rand)\ THEN\ X_{NH} = 0 \tag{6}$$

2.3 Related Works

This part reviews some related works on wrapper-based FS. Both the traditional and meta-heuristic ones, as shown in the subsequent parts. However, this study focuses mostly on the evolutionary algorithms; for more details on the swarm intelligence based approaches refer to [7].

2.3.1 Classical Wrapper-Based Feature Selection

As mentioned earlier, wrapper-based FS algorithms are highly computationally cost compared to the filter-based FS algorithms [15, 48]. Perhaps, this is due to the longer evaluation processes involved in the training and testing of the classifier. Furthermore, since the search space of the FS problem is exponential to the number of features. Therefore, searching for the entire search space is impractical. Based on that the existing wrapper-based techniques used stochastic or greedy search [21, 42].

The most common FS techniques that practice the greedy hill-climbing are sequential feature selection (SFS) [1] and sequential backward selection (SBS) [41].

In SFS, it begins with an empty set of features and keeps on adding one feature at a time in an iterative manner until adding another feature will not enhance the existing classification performance then it stops. Unlike, in SBS where it starts with a full set of features and keeps on looping to remove one feature at a time until removal of a feature cannot improve the existing classification accuracy (error rate). Apart from the computational cost incurred on a large number of data sets, another major

drawback of both SFS and SBS is the nesting effect, since any feature that is added or removed cannot be undone. Thus, they both are trapped into the local optima easily [40, 41].

Although, [38] developed a "plus q take away r" technique that will escape the nesting effect, SBS was applied r times in a back-tracking order while SFS is applied q times in forwarding step order. Determining better numbers for q and r is required, to solve this problem of having fixed values for both q and r. Then, [47] enhanced it by introducing a floating-point in both SFS (sequential forward floating selection (SFFS)) and SBS (sequential backwards floating selection (SBFS)) that automatically determine the value of q and r. Although, both SFFS and SBFS proved to be useful in some cases, [67] argued that they could likely trap into local optimal even if the benchmark function is monotonic (neither decrease nor increase) and yet is a small-scale problem.

Inline spectral frequencies (LFS), the number of features to be used for evaluation in every step are limited. As such, the computational efficiency of the sequential forward's methods was enhanced by the LFS and sustained an analogous accuracy of the selected subset of features. But, LSF ranks all features without taking into consideration whether some features are present or not, and this restricts the performance of the LSF algorithm particularly the interaction between features.

2.3.2 Wrapper-Based Feature Selection with Meta-Heuristic Algorithms

As mentioned earlier, meta-heuristic algorithms have become more robust in handling NP-hard problems, including FS. Huang and Wang [30] employed GA for both FS and SVM parameter optimisation on a real-world data set. The results obtained are in favour of the GA in terms of classification accuracy and fewer number of features compared with the grid algorithm reported in work. Also, [59] proposed another GA for FS and SVM for parameter selection in the detection of diabetic retinopathy. A promising result was obtained on 60 images of data sets. An enhanced GA (EGA) was proposed in [19] to reduce text dimensionality. It is incorporated with six filter FS methods to create a hybrid one. Finally, experimental results showed that the hybrid outperformed the single approach as well as the traditional GA. Recently in [3], the highest accuracy of 99.48% was attained on two different Wisconsin breast cancer data sets. GA was used for FS before applying the five different classifiers. The results obtained are better than the others.

Unler and Murat [53] present a discrete PSO for FS in binary classification problems. The proposed approach incorporates an adaptive FS technique which dynamically takes into consideration the relevance and dependence of the features included in the feature subset. The experimental results indicated that the proposed discrete PSO algorithm is competitive in terms of both classification accuracy and computational performance compared with the scatter search and tabu search algorithms on openly available data sets.

Vieira et al. [58] proposed a modified binary PSO (MBPSO) for FS with simultaneous optimisation of SVM parameters to predict the outcome of patients with septic shock. The results indicated that MBPSO performed very well compared with the standard PSO both in terms of accuracy and features selected. However, when compared to GA, the same accuracy was recorded, but the MBPSO select fewer features.

Similarly, [31] developed a supervised PSO-based rough set FS for medical data diagnosis. Two different algorithms PSO-based relative reduct (PSO-RR) and PSO-based quick reduct (PSO-QR) are presented. The results obtained showed that the proposed algorithms performed better in terms of the fewer number of features, classification accuracy and computational time compared to the standard PSO and other methods reported.

Recently, PSO initialisation and updating mechanism are changed to suit better FS problems in [52]. The discretisation is applied before the FS since discretisation is considered an essential task of FS. A potential particle swarm optimisation (PPSO) is proposed which employs a modern illustration that can minimise the search space of the problem and an advanced fitness function to assess candidate solutions better and direct the search process. The results of the experiments on the ten high-dimensional data sets disclosed that PPSO chooses fewer than 5% of the number of features for all data sets. Compared with the two-stage method which uses bare-bone PSO (BBPSO) for FS on the discretised data, PPSO attains a better accuracy on seven data sets. Furthermore, PPSO gains improved classification accuracy than evolve PSO (EPSO) on eight data sets with a reduced feature size on six data sets. Moreover, PPSO also performs better than the three compared approaches and achieves similar to one approach on majority data sets in terms of both learning capacity as well as generalisation ability.

To predict heart disease among patients, [18] used cuckoo search and rough set for FS, and the disease prediction is made using fuzzy. A better result was achieved in four different benchmark data sets.

Recently, [13] used a modified cuckoo search along with rough set to build the fitness function that takes several features into the reduce set and classification into consideration. SVM and KNN are used to evaluate the performance of the proposed approach. The results obtained indicate the superiority of the method used and can significantly improve performance.

Despite the attempt to solve the lingering issues of the wrapper-based FS, still the existing works cannot successfully evolve the best subset of features with improving accuracy on some of the data sets [23]. COA presented in [49] is among the evolutionary algorithms that show promising results in handling different combinatorial optimisation problem, including NP-hard; However, despite its proven records, especially in dealing with filter-based FS in [55].

COA has been applied to solve different kinds of problems. Recently, it is used with harmony search for optimum tuning of fuzzy PID controller for LFC of interconnected power systems in [20]. Energy-aware clustering in wireless sensor networks in [35], accelerated COA was proposed in [22] where simulated annealing algorithm

was used in place of the k-means clustering of the standard COA in vehicle routing problems.

Compared to GA, the imperialist competitive algorithm (ICA), CSA and PSO. COA is simpler to implement and can converge rapidly [6, 29]. Its application, specifically for the wrapper-based FS, is not fully investigated.

3 Proposed Wrapper-Based Feature Selection Approaches

Thus, in this part first, both BCOA and COA are adopted and used for wrapper-based FS. The detail of how each of the experiments was carried out can be seen in the subsequent parts.

3.1 BCOA and COA for Feature Selection

Two wrapper-based FS are proposed, namely, BCOA-FS and COA-FS. Throughout the evolutionary training process, Eq. 7 is applied as the fitness evaluation function to estimate and evaluate the best cuckoo habitat i, where the position x_i signifies the subsets of features.

$$Fitness(x_{(i)}) = ErrorRate \tag{7}$$

where *ErrorRate* is calculated based on Eq. 8:

$$ErrorRate = \frac{(FP + FN)}{(TP + TN + FP + FN)} \tag{8}$$

where FP, FN ,TP, and TN, are the respective false positives, false negatives, true positives, and true negatives.

3.2 A Combined Fitness Function for BCOA and COA Feature Selection

The subset of feature selected by both BCOA-FS and COA-FS may probably comprise some redundancy since the fitness function in Eq. 7 does not reduce the features. However, it hypothesises that the same or less accuracy might be realised using a smaller subset of features. To additionally minimise the feature size deprived of affecting the classification error rate, a two-step FS method (BCOA-2S and COA-2S) is introduced, where the entire evolutionary procedure is separated into two steps.

In step 1, both BCOA-FS and COA-FS emphasises on improving accuracy. Whereas, in step 2, the features are involved in the fitness function. Furthermore, step 2 begins with the solutions realised in step 1, which certifies the reductions in the features according to the subsets of the features with the best accuracy.

The proposed two-step fitness function employed in both BCOA-2S and COA-2S is shown in Eq. 9:

$$Fitness_2(x_i) = \begin{cases} Step\ 1, & Error\,Rate \\ Step\ 2, & Error\,Rate\ \beta * \dfrac{M}{n} + (1-\beta) * \frac{M\,Eror\,Rate}{n\,Eror\,Rate} \end{cases} \tag{9}$$

where Error Rate is the classification error rate attained by the selected subset of features. $\beta \in [0, 1]$ is a constant number within the range [63]. M denotes the size of selected features and n is the total feature size. $nErrorRate$ is the error rate obtained by using the total feature size for classification on the training set. In step 2, the fitness function considers both the feature size as well as the error rate. It guarantees that these two components are in a similar array, i.e. [0, 1], and the of feature size is normalised and represented by M/n.

The classification performance is represented by $(MErrorRate)/(nErrorRate)$ rather than Error Rate alone to circumvent the circumstances, whereby Error Rate is too insignificant (for instance, < 0.005), and M/n plays a significant role inside the fitness function. In a situation like this, the feature size considers most compared to the error rate, which might have a subset of features with high error rate compared to using the full-length feature size. Meanwhile Error Rate would be lesser than $nErrorRate$ at the end of the step 1, $(MErrorRate)/(nErrorRate)$ is in the similar array as M/n, i.e. [0, 1].

As soon as they are joined into a single fitness function, β is employed to display the comparative significance of the chosen features and $(1 - \beta)$ displays the outstanding significance of the error rate. The $Errorrate$ is expected to be more significance compared with feature size, thus β is assign to be lesser than $(1 - \beta)$ (i.e. $\beta < 0.5$). The pseudocode of (BCOA-FS and BCOA-2S) along with (COA-FS and COA-2S) can be seen in Algorithm 1 and Algorithm 2, respectively. The main difference between BCOA-FS, COA-FS and BCOA-2S depend on the fitness evaluation function, that is illustrated mostly in the grey lines of algorithms.

The detailed of the proposed wrapper-based BCOA is depicted in Algorithm 1, whereby Eqs. 7 and 8 have been used as the respective fitness functions. The grey colour signifies the areas where the equations and initialisation as per feature selection problems are used in the proposed algorithms.

Algorithm 1 Proposed BCOA-FS and BCOA-2S

1: **Start**
2: **Initialise each habitat with some features from a dataset**
3: **Collect the features in their respective habitats**
4: **Explain ELR for every single cuckoo using Equation 5 and Equation 6**
5: **Allow the cuckoos to lay their eggs in their matching ELR**
6: **Destroy those cuckoos familiar by the multitude birds**
7: **Allow egg to incubate and baby chicken raise**
8: **Estimate the environment of every recently grownup cuckoo**
9: **Limits cuckoos' highest number in location and abolish those that exist in poorer environments**
10: **Group cuckoos and discover finest cluster and choose goal line environment**
11: **Allow the new cuckoo populace to settle at the goal line environment**
12: **Return the optimum solution (selected features)**
13: **Evaluate the fitness function according to Equation 8 in BCOA-FS and Equation 9 in BCOA-2S**
14: **If the stop condition is satisfied to stop, else go to 3**
15: **Stop**

Algorithm 2 Proposed COA-FS and COA-2S

1: **Start**
2: **Initialise each habitat with some features from a dataset**
3: **Collect the features in their respective habitats**
4: **Explain ELR for every single cuckoo using Equation 2 and Equation 3**
5: **Allow the cuckoos to lay their eggs in their matching ELR**
6: **Destroy those cuckoos familiar by the multitude birds**
7: **Allow egg to incubate and baby chicken raise**
8: **Estimate the environment of every recently grownup cuckoo**
9: **Limits cuckoos' highest number in location and abolish those that exist in poorer environments**
10: **Group cuckoos and discover finest cluster and choose goal line environment**
11: **Allow the new cuckoo populace to settle at the goal line environment**
12: **Return the optimum solution (selected features)**
13: **(8) in COA-FS and (9) in COA-2S**
14: **If the stop condition is satisfied to stop, else go to 3**
15: **Stop**

Figure 2 shows that each cuckoo is initialised according to Eq. 1 for each of the data sets. Then, the number of features in each habitat are collected while those eggs detected in the habitat are killed. At this juncture, the fitness function for both BCOA-FS and COA-FS are evaluated using Eqs. 7 and 8, respectively.Whereas, the fitness function for the combined FS (BCOA-2S and COA-2S) are evaluated using

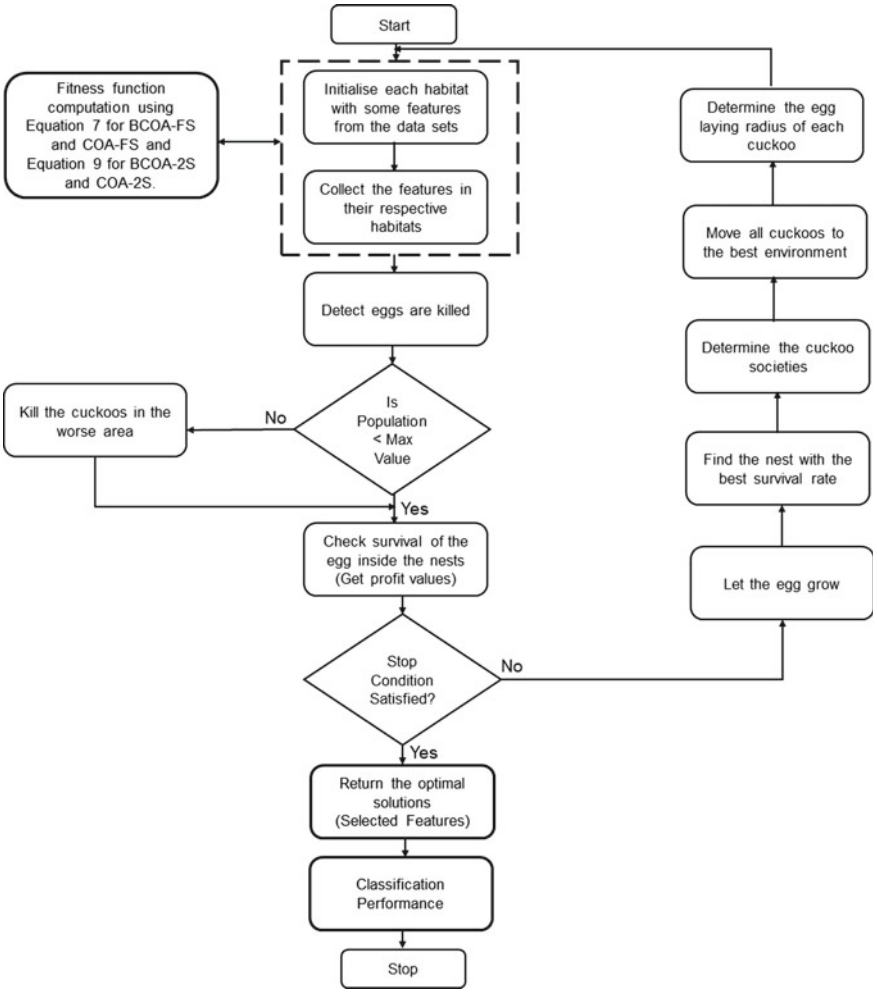

Fig. 2 Flowchart of the (BCOA-FS and BCOA-2S) and (COA-FS and COA-2S)

the fitness function in Eq. 9. The population is compared with maximum value, and if the population is less than the maximum value, then the cuckoo in the worst area would be killed, otherwise it gets profit values (check the survival of the egg inside the nest). Then stop condition evaluated; if yes, it leads the eggs to grow. However, the nest found with the best survival rate among the cuckoo societies is transferred to the best society according to Eqs. 5 and 6 for the BCOA. Whereas, Eqs. 2 and 3 are for the standard COA. Based on Eq. 2, one can find the best ELR and repeat all the steps. Otherwise, return the optimum solution of the highest ranked features. Then, finally, the best, along with the average of them, are selected using a classifier.

The time complexity of both BCOA-FS and COA-FS based on the fitness function in Eq. 7 is $O(\frac{1}{m}) + O(\frac{1}{n})$. The term m and n represent the number of selected features and the population size, respectively. The binary search for using BCOA runs in $O(n)$ time, while the COA search in $O(log_2 n)$ time. Thus, the computational complexity of BCOA-FS is $O(\frac{1}{m}) + O(\frac{1}{n}) + O(m)$ and COA-FS is $O(\frac{1}{m}) + O(\frac{1}{n}) + O(log_2 n)$.

Based on the fitness function in Eq. 8, the complexity is $O(\frac{1}{m^2}) + O(\frac{1}{n^2})$. Therefore, the total complexity of the BCOA-2S is $O(\frac{1}{m^2}) + O(\frac{1}{n^2}) + O(n)$ and that of COA-2S is $O(\frac{1}{m^2}) + O(\frac{1}{n^2}) + O(log_2 n)$. Therefore, BCOA-FS and COA-FS can complete its process within a shorter time in most cases compared to its BCOA-2S and COA-2S counterpart.

4 Experimental Design

This part describes the data sets used in conducting the experiments. Parameters settings, as well as benchmark, approaches are used to test the performance of the proposed methods.

4.1 Experimental Datasets

The data sets used in this study are the 26 well-known University of California Irvine (UCI) Machine learning data sets with distinct features. It contains a different number of features ranging from 9 to 500, 14 categorical and 12 continuous data type, 72–5000 instances, 13 binary classes and 13 multi-classes. These different appearances of the data set, especially on the number of features that contain smaller, medium and large features are the motives behind the selection of the data sets as shown in Table 1. The data sets can be found in [17] or can be downloaded freely at https://www.ics.uci.edu.ml/~earn.

Furthermore, most of the data sets have been used recently in the works of [2, 15, 43, 44], which clearly show that the data sets are goods for benchmarking FS problems. The data sets contain both categorical and continuous data that can be useful in demonstrating the comparison between the categorical discrete and the continuous data. Continuous data have infinite values in the form of decimal numbers, while the categorical discrete values are mostly finite values in groups.

4.2 Experimental Parameter Settings

The parameters employed for the experiments were set as follows: initial and maximum population are set to 5 and 20, respectively. Moreover, the proposed algorithms were run 40 independent times on each data set. The parameter settings used for the

Table 1 List of data sets

S/N	Data set	Features	Classes	Instances	Data type
1	Wine	13	3	178	Continuous
2	Australian	14	2	690	Continuous
3	Zoo	17	7	101	Continuous
4	Vehicle	18	4	846	Continuous
5	Lymphography (Lymph)	18	4	148	Categorical
6	Mushroom	24	2	5644	Categorical
7	Spect	22	2	267	Categorical
8	German	24	2	1000	Continuous
9	Leddisplay	24	10	1000	Categorical
10	WBCD/BreastEW	30	2	569	Continuous
11	Ionosphere (Ionosp)	34	2	351	Continuous
12	Dermatology	34	6	366	Categorical
13	Soybean Large	35	19	307	Categorical
14	Chess (KrvskpEW)	36	2	3196	Categorical
15	Connect4	42	3	44473	Categorical
16	LungCancer (Lung)	56	3	32	Continuous
17	Promoter	57	2	106	Categorical
18	Sonar	60	2	208	Continuous
19	Splice	60	3	3190	Categorical
20	Optic	64	10	5620	Categorical
21	Audiology	68	24	226	Categorical
22	Coil2000	85	2	9000	Categorical
23	Hillvalley	100	2	606	Continuous
24	Musk1 (Clean1)	166	2	476	Continuous
25	DNA	180	3	3186	Categorical
26	Madelon	500	2	4400	Continuous

proposed $COA-FS$, $COA-2S$, $BCOA-FS$ and $BCOA-2S$ algorithms are chosen based on the work of [45, 49]. The maximum number of iterations was set to 100.

Also, similar to the work of [63] and [65]. In the experiments, all the rows in each of the data sets were partition into two groups: a training group and a test group. The most partitioning approach is that 2/3 (about 66%) of the rows in the data sets are in the training group and 1/3 (almost 33%) of the rows are in the test group [11]. To simplify the process, we divide 70% of the rows into each data set as the training group and the remaining 30% as the test group. The rows are chosen so that the percentage of rows from various classes are equal in both the training group. The proposed wrapper-based methods need a classifier to estimate the suitability of the

Table 2 Existing wrapper-based approaches

References	Type	Acronym	Year
[8]	Single-objective	GSBS	1994
[9]	Single-objective	BAIS	2009
[63]	Single-objective	ErFS and 2SFS	2012
[27]	Single-objective	ABC-ER, ABC-Fit2C	2018

selected subsets of features. A KNN (with $K = 5$) was used in the experiments, to reduce the wrapper-based computational time [4].

The experiments of GSBS and LFS are carried out using the popularly known Waikato Environment for Knowledge Analysis (WEKA) [28]. The entire settings in LFS along with GSBS are saved to the defaults since they can obtain better results. Also, a 5NN was used in both LFS and GSBS, which generate a unique solution (feature subset) for each data set.

4.3 Benchmark Approaches

Scrutinise the concert of the proposed wrapper-based approaches in this chapter. The results found are related to the previous works, as shown in Table 2. From (Table 2), two traditionally known wrapper-based FS methods, namely linear forward selection (LFS) [25] and greedy stepwise backward selection (GSBS) [8] are used as benchmark methods. Both LFS, together with GSBS, were consequential of SFS and SBS, respectively. LFS [25] limits the number of features that are selected in each step of the forward selection, which can reduce the number of evaluations. As such, the LFS is computationally less expensive compared to the SFS and will get better results. More details about the LFS is in [25].

On the other hand, the greedy stepwise based FS algorithm mostly shifts either forward or backwards in the search space [8]. Provided that the LFS makes a forward selection, a backward search is selected in the greedy stepwise search to create a greedy stepwise backward selection (GSBS). GSBS begins with all the feature size and halts if the removal of any outstanding feature results in a reduction in evaluation measure, i.e. the error rate of the classifier. Also, the work in [63] was used as a benchmark method for both single and multi-objective wrapper-based approach, due to the similarities in the data sets. The detail explanation of the results obtained and the analysis is presented in the subsequent sections.

The details of the results obtained are offered in the subsequent section.

5 Results and Discussions

This part deliberates on the results of the proposed methods, comparison between them and other existing works that their work coincide with the data sets apply in this study.

5.1 Results of the Proposed BCOA-FS and COA-FS

The results of both the categorical and continuous data sets for BCOA-FS and COA-FS are displayed in Tables 4 and 3, respectively. The results showed a comparison between all the proposed wrapper-based methods. From the tables, "BCOA-FS" and "COA-FS" represent the proposed wrapper-based methods that adopt both BCOA and COA, respectively. "All" stands for all features used for each of the data sets. Besides, "Ave Size", "Ave Acc" and "Best Acc" represents average feature size, average accuracy and best accuracy attained by each of the data sets for the 40 independent runs, respectively.

The results proposed BCOA-FS outperformed its COA-FS counterpart on the continuous data sets. Out of the 12 data sets in the table (Table 3), they recorded similar feature size, best accuracy and average accuracy on WineEW, Australian, Zoo and to some extents on Vehicle data sets. However, as the number of features increases, BCOA-FS outperformed COA-FS on the remaining eight data sets. In addition to that a similar performance was slightly noticed between BCOOA-FS and COA-FS on HillValley datasets. On the average, it is clear that BCOA-FS outperformed its COA-FS counterpart on the majority of the data sets, and hence considered the best wrapper-based feature selection.

Alternatively, a comparison between BCOA-FS and COA-FS was made on categorical data sets, as shown in the results Table 4. Similar to the continuous data sets, the categorical data sets also recorded similarities in terms of the mean of selected features, best accuracy and average accuracy on data sets with fewer feature size as such as Lymph, Mushroom, Spect and Leddisplay. However, from Dermatology that has 34 total number of available features, there is a change in performance between BCOA-FS and COA-FS. The results also imply that as the feature size increase the BCOA-FS perform better than the COA-FS in all the data sets except in Coil2000. Perhaps due to a large number of instances in the Coil2000 data set.

5.2 Results of the Proposed BCOA-2S and COA-2S

The results of both the categorical and continuous data sets for BCOA-2S and COA-2S are also displayed in Tables 4 and 3, respectively. The terms "BCOA-2S" and "COA-2S" represents the proposed combined accuracy and selected features into a

Table 3 Results of the BCOA-FS, COA-FS, BCOA-2S and COA-2S for continuos data sets

Datasets	Approach	Ave-Size	Best-Acc	Ave- Acc	Time(s)	Datasets	Approach	Ave-Size	Best-Acc	Ave- Acc	Time(s)
WineEW	All	13	77.25			Australian	All	14	71.25		
	LFS	7	74.57				LFS	4	71.25		
	GSBS	8	86.21				GSBS	12	70.45		
	BCOA-FS	8	86.21	96.75	6120		BCOA-FS	3.42	89.25	86.25	10767.6
	COA-FS	8	86.21	96.75	6217		COA-FS	3.42	89.25	86.25	10871.8
	COA-2S	7	86.21	97.25	6516		COA-2S	3.32	90.56	87.58	11170.8
	BCOA-2S	7	100	86.21	6417		BCOA-2S	3.32	90.58	87.58	11071.8
Zoo	All	16	86.21			Vehicle	All	18	84.25		
	LFS	8	86.21				LFS	9	84.05		
	GSBS	7	86.21				GSBS	15	78.51		
	BCOA-FS	9	86.21	96.25	4559.4		BCOA-FS	9.1	89.56	85.22	10143.4
	COA-FS	9	86.21	96.25	4665.4		COA-FS	9.1	89.56	85.22	10261.6
	COA-2S	8	86.21	98.56	4964.4		COA-2S	9	91.25	91.25	10560.6
	BCOA-2S	8	86.21	98.56	4865.4		BCOA-2S	9	91.56	91.56	10461.6
Germany	All	24	86.21			WBCD	All	30	95.20		
	LFS	3	86.21				LFS	10	84.11		
	GSBS	17	86.21				GSBS	24	85.62		
	BCOA-FS	12.5	86.21	72.48	17044.2		BCOA-FS	13.41	95.21	91.28	8393.4
	COA-FS	12.1	86.21	74.25	17180.8		COA-FS	13.52	94.75	92.45	8813.07
	COA-2S	11.5	86.21	75.45	17479.8		COA-2S	13.21	98.22	98.22	9804.96
	BCOA-2S	11.2	86.21	75.70	17380.8		BCOA-2S	12.45	100.00	100.00	9448.416
IonosphereEW	All	34	86.21			LungCancer	All	56	70.00		
	LFS	5	86.21				LFS	6	90.00		
	GSBS	30	86.21				GSBS	33	90.00		
	BCOA-FS	12.56	88.82	86.21	7835.4		BCOA-FS	27	90.65	88.65	76995
	COA-FS	12.12	92.46	86.21	8227.17		COA-FS	27.6	90.31	89.56	80844.75
	COA-2S	11.85	94.26	86.21	9038.7		COA-2S	25.2	92.25	92.25	116816
	BCOA-2S	10.65	98.25	86.21	8710.02		BCOA-2S	25	95.50	95.50	112568.2

(continued)

Table 3 (continued)

Datasets	Approach	Ave-Size	Best-Acc	Ave- Acc	Time(s)	Datasets	Approach	Ave-Size	Best-Acc	Ave- Acc	Time(s)
SonarEW	All	60	86.21			Hillvalley	All	100	56.75		
	LFS	5	86.21				LFS	8	57.69		
	GSBS	45	86.21				GSBS	90	49.45		
	BCOA-FS	23.42	90.65	86.21	67260.6		BCOA-FS	45.25	68.52	65.22	72903.6
	COA-FS	24.5	86.45	86.21	70623.6		COA-FS	46.12	64.26	59.62	76548.78
	COA-2S	21.9	96.58	86.21	107956		COA-2S	44.6	70.65	68.56	118503
	BCOA-2S	19.8	98.20	86.21	104030		BCOA-2S	42.4	74.25	70.35	114193.8
Musk1(Clean1)	All	166	86.21			Madelon	All	500	70.90		
	LFS	10	86.21				LFS	7	64.62		
	GSBS	122	86.21				GSBS	489	51.28		
	BCOA-FS	84.25	88.88	86.21	70106.4		BCOA-FS	255.5	80.00	80.00	143303.4
	COA-FS	85.6	88.81	86.21	73611.7		COA-FS	259.2	79.89	77.47	150468.6
	COA-2S	41.25	88.97	86.21	120893		COA-2S	250.6	81.45	79.45	218918.7
	BCOA-2S	50.1	90.50	86.21	116497		BCOA-2S	248.52	83.45	81.65	210958

single fitness function for both BCOA and COA, respectively. All other headings in the table are the same as explained in the previous subsection.

There are 14 categorical data sets and 12 continuous data sets that make a total of the 26 data sets used in this research. Out of all the 14 categorical data sets, BCOA-2S accomplished better results than COA-2S in terms of the average number of selected features, best accuracy and average accuracy on almost all the data sets. Although in Leddisplay data set, it has similar performance and same best accuracy on Mushroom data set.

On the other hand, the results of the continuous data sets also are in favour of BCOA-2S compared to the COA-2S in the majority of the data sets. A similar performance was obtained on some few data sets such as WineEW, Australian, Zoo and Vehicle. However, as the feature size increases, the BCO-2S also performed better than COA-2S. It is in contrast with categorical data sets no matter the feature size, BCOA-2S performed better than its COA-2S counterpart in almost all the data sets regardless of the number of features in the data sets.

5.3 Comparison Between Proposed Methods and Classical Methods

A result of LSB and GSBS was reported to further compare with the proposed methods. The results clearly show that LFS could select fewer number of features than GSBS in the majority of the data sets. However, GSBS achieve the best classification results in most of the data sets. Although on some data sets with a fewer number of features, they recorded similar performance. But as the number of features increases, LFS select the smallest feature and GSBS obtained the best accuracy.

Comparing LSF and GSBS with the proposed wrapper-based FS, one can observe that our proposed approaches outperformed both LSF and GSBS in terms of the number of selected features, best accuracy, and average accuracy in all most all the data sets, both continuous (Table 3) and categorical (Table 4).

5.4 Comparison Between Proposed Methods and Other Existing Methods

To further evaluate the performance of the proposed methods and consequently be fair in assessing the proposed wrapper-based multi-objective. Some related works with similar datasets were used for comparison, as shown in Table 2. The details of the comparison are enumerated below:

1. **Comparison with ErFS and 2SFS**
   The results of the proposed wrapper-based feature selection are compared with the one in work [63], where ErFS and 2SFS represent the BCOA-FS and BCOA-

Table 4 Results of the BCOA-FS, COA-FS, BCOA-2S and COA-2S for categorical data sets

Datasets	Approach	Ave-Size	Best-Acc	Ave- Acc	Time(s)	Datasets	Approach	Ave-Size	Best-Acc	Ave- Acc	Time(s)
Lymph	All	18	87.25			Mushroom	All	24	99.20		
	LFS	9	93.48				LFS	8	89.51		
	GSBS	14	84.21				GSBS	18	91.25		
	BCOA-FS	9.2	93.56	92.48	9072		BCOA-FS	12.5	98.98	95.65	15597
	COA-FS	10.5	92.65	90.45	9525.6		COA-FS	13.5	98.98	94.75	16376.85
	COA-2S	8	95.00	95.00	10311.84		COA-2S	8	100.00	98.60	24017.4
	BCOA-2S	7	96.55	94.65	9936.864		BCOA-2S	7	100.00	99.80	23144.04
Spect	All	22	85.10			Leddisplay	All	24	100.00		
	LFS	10	80.00				LFS	7	100.00		
	GSBS	18	82.50				GSBS	5	100.00		
	BCOA-FS	10.8	88.98	86.65	9424.8		BCOA-FS	12	100.00	100.00	14328
	COA-FS	11	87.75	84.68	9896.04		COA-FS	12	100.00	100.00	15044.4
	COA-2S	9	91.75	88.95	10733.58		COA-2S	9	100.00	100.00	19118.88
	BCOA-2S	8.2	94.75	91.25	10343.27		BCOA-2S	9	100.00	100.00	18423.65
Dermatology	All	34	94.85			Soybean Large	All	35	87.25		
	LFS	12	83.25				LFS	11	85.56		
	GSBS	26	89.65				GSBS	24	84.21		
	BCOA-FS	12.5	90.25	89.65	15102		BCOA-FS	12.3	88.65	84.25	17188.2
	COA-FS	13.6	90.12	89.62	17557		COA-FS	12.9	86.75	85.25	18129.3
	COA-2S	10.2	93.12	92.52	17856		COA-2S	7.1	92.75	90.35	20340
	BCOA-2S	9.5	97.56	95.63	17757		BCOA-2S	13.2	95.65	91.58	20241
Chess (KrvskpEW)	All	36	92.00			Connect4	All	36	79.78		
	LFS	11	86.21				LFS	6	70.73		
	GSBS	24.2	85.65				GSBS	41	71.68		
	BCOA-FS	12.5	87.75	86.25	43833.6		BCOA-FS	15	94.25	93.15	43833.6
	COA-FS	13	86.44	85.98	48733.4		COA-FS	18	92.25	89.65	46733.4
	COA-2S	8	95.50	93.25	54032.4		COA-2S	12	96.50	93.50	54032.4
	BCOA-2S	9	100.00	98.56	53933.4		BCOA-2S	10	100.00	100.00	53933.4

(continued)

Table 4 (continued)

Datasets	Approach	Ave-Size	Best-Acc	Ave- Acc	Time(s)	Datasets	Approach	Ave-Size	Best-Acc	Ave- Acc	Time(s)
Promoter	All	57	90.00			Splice	All	60	79.65		
	LFS	6	90.00				LFS	28	78.83		
	GSBS	50	90.00				GSBS	47	68.65		
	BCOA-FS	20.5	92.11	92.11	77416.2		BCOA-FS	28	80.65	79.95	92320.2
	COA-FS	25	91.89	89.68	77796.4		COA-FS	30.5	80.05	78.45	93060.4
	COA-2S	13.6	91.25	90.65	108095.4		COA-2S	15.6	84.65	83.20	123359.4
	BCOA-2S	12.5	94.50	91.25	107996.4		BCOA-2S	12.4	89.88	84.56	123260.4
Optic	All	64	98.88			Audiology	All	68	65.56		
	LFS	36	98.69				LFS	9	64.63		
	GSBS	38	98.75				GSBS	24	64.63		
	BCOA-FS	30	100.00	98.98	74176.2		BCOA-FS	23	76.62	70.46	73080
	COA-FS	32	100.00	98.98	75937		COA-FS	22.4	75.48	69.98	75020.8
	COA-2S	26.5	100	100.00	106236		COA-2S	18.54	79.68	73.71	105319.8
	BCOA-2S	20.45	100	100.00	106137		BCOA-2S	20.9	82.56	80.54	105220.8
Coil2000	All	85	94.55			DNA	All	180	83.01		
	LFS	10	93.58				LFS	34	85.08		
	GSBS	31	93.98				GSBS	173	82.63		
	BCOA-FS	36	92.65		74725.2		BCOA-FS	88.5	90.55	89.25	99032.4
	COA-FS	37.5	93.21	92.01	78721.6		COA-FS	90.5	89.25	78.65	94025
	COA-2S	20.6	93.65	92.52	109020.6		COA-2S	48.56	91.25	90.65	144324
	BCOA-2S	19.6	97.75	97.75	108921.6		BCOA-2S	52.55	92.01	90.93	144225

2S used in this research. The significant difference between the two is the use of the EC algorithm. An outstanding EA, COA, in particular, was used in this research. Whereas, the exiting works used the most common SI based algorithm (PSO). The results indicated our proposed COA and BCOA which outperformed the existing practices of PSO used in [63]. The result is not surprising because COA is reported to be more robust and can attain better results as claimed in the work of [5, 49].

From Table 5, it is clear that all the comparisons were made on the continuous data sets. Out of the 10 data sets used, it shows that in almost all cases, our proposed methods performed better than the existing one. However, in Zoo and Ionosphere data sets, for example, the existing methods performed better in terms of average accuracy. Nevertheless, the best accuracy and the number of selected features clearly show that our proposed methods performed well.

2. **Comparison with ABC-ER and ABC-Fit2C**
   The comparison between the proposed methods with ABC-ER and ABC-Fit2C in [27] is shown in Table 6. The comparison shows that our proposed methods performed better than all the seven data sets on both accuracy and number of selected features. However, even though ABC-Fit2C chooses slightly fewer features on German and Vehicle data sets than the proposed methods, but still, the proposed methods attained an improved classification accuracy compared to the ABC-Fit2C and ABC-ER. Therefore, the results displayed in Table 6 indicated that the proposed methods can effectively evolve a fewer number of features and yet achieve a better classification performance.
3. **Comparison with BAIS**
   The results obtained by the proposed methods with Bayesian and artificial immune system (BAIS) in work [10] is displayed in Table 7. The results show the superiority of the proposed methods on all the five data sets. The proposed methods outperformed the BAIS with nearly 10% of the classification accuracy on Ionosphere and Sonar data sets. Whereas, around 2–3% of improvement was realised on the proposed methods compared to the BAIS on the Mushroom, WineEW and WBCD data sets. Moreover, fewer subsets of features were selected in the proposed method than the BAIS. Therefore, both in terms of selected features and the classification accuracy, the proposed methods outperformed the BAIS in all aspects.

5.5 Comparisons Between BCOA and COA

Comparing the performance of COA and BCOA for adopted or combined objectives as shown in tables (Tables 4 and 3) for both categorical and continuous data sets, one can observe that BCOA outperformed COA in terms of number of selected features, accuracy and best accuracy for all the proposed methods.

Table 5 Comparison of (BCOA-FS, COA-FS, BCOA-2S and COA-2S) with ErFS and 2SFS

Datasets	Approach	Ave-Size	Best-Acc	Ave- Acc	Datasets	Approach	Ave-Size	Best-Acc	Ave- Acc
WineEW	BCOA-FS	8	86.21	96.75	Australian	BCOA-FS	3.42	89.25	86.25
	ErFS	8	95.96			ErFS	3.88	87.44	85.44
	COA-FS	8	86.21	96.75		COA-FS	3.42	89.25	86.25
	COA-2S	7	86.21	97.25		COA-2S	3.32	90.56	87.58
	2SFS	8	95.96			2SFS	3.42	87.44	84.24
	BCOA-2S	7	100.00	86.21		BCOA-2S	3.32	90.58	87.58
Zoo	BCOA-FS	9	86.21	96.25	Vehicle	BCOA-FS	9.1	89.56	85.22
	ErFS	9.18	97.14	95.50		ErFS	9.52	87.01	87.01
	COA-FS	9	86.21	96.25		COA-FS	9.1	89.56	85.22
	COA-2S	8	98.56	86.21		COA-2S	9	91.25	91.25
	2SFS	9.18	97.14	95.50		2SFS	8.65	87.01	84.95
	BCOA-2S	8	98.56	86.21		BCOA-2S	9	91.56	91.56
German	BCOA-FS	12.5	86.21	72.48	WBCD	BCOA-FS	13.41	95.21	91.28
	ErFS	12.58	72.00	69.41		ErFS	13.42	94.74	93.39
	COA-FS	12.1	86.21	74.25		COA-FS	13.52	94.75	92.45
	COA-2S	11.5	86.21	75.45		COA-2S	13.21	98.22	98.22
	2SFS	11.92	72.00	72.00		2SFS	5	94.74	94.74
	BCOA-2S	11.2	86.21	75.70		BCOA-2S	12.45	100.00	100.00
Ionosphere	BCOA-FS	12.56	88.82	86.21	LungCancer	BCOA-FS	27	90.65	88.65
	ErFS	12.58	93.33	88.40		ErFS	27.35	80.00	72.00
	COA-FS	12.12	92.46	86.21		COA-FS	27.6	90.31	89.56
	COA-2S	11.85	94.26	86.21		COA-2S	25.2	92.25	92.25
	2SFS	12.05	93.33	91.43		2SFS	27.38	90.00	80.00
	BCOA-2S	10.65	98.25	86.21		BCOA-2S	25	95.50	95.50

(continued)

Table 5 (continued)

Datasets	Approach	Ave-Size	Best-Acc	Ave- Acc	Datasets	Approach	Ave-Size	Best-Acc	Ave- Acc
Hillvalyey	BCOA-FS	45.25	68.52	65.22	Musk1(Clean1)	ErFS	86.48	88.81	84.58
	ErFS	47.32	61.81	57.54		BCOA-FS	84.25	88.88	86.21
	COA-FS	46.12	64.26	59.62		COA-FS	85.6	88.81	86.21
	COA-2S	44.6	70.65	68.56		COA-2S	41.25	88.97	86.21
	2SFS	47.04	61.81	57.57		2SFS	85.58	88.81	88.88
	BCOA-2S	42.4	74.25	70.35		BCOA-2S	50.1	90.50	86.21
Madelon	BCOA-FS	255.5	80.00	80.00					
	ErFS	258.1	79.49	76.55					
	COA-FS	259.2	79.89	77.47					
	COA-2S	250.6	81.45	79.45					
	2SFS	256.48	79.36	76.52					
	BCOA-2S	248.52	83.45	81.65					

Table 6 Comparison of (BCOA-FS, COA-FS, BCOA-2S and COA-2S) with ABC-ER and ABC-Fit2C

Datasets	Approach	Ave-Size	Best-Acc	Ave- Acc	Datasets	Approach	Ave-Size	Best-Acc	Ave- Acc
Optical	BCOA-FS	30	100.00	98.98	Vehicle	BCOA-FS	9.1	89.56	85.22
	ABC-ER	41.13	98.10	–		ABC-ER	9.86	79.53	–
	COA-FS	32	100.00	98.98		COA-FS	9.1	89.56	85.22
	COA-2S	26.5	100	100.00		COA-2S	9	91.25	91.25
	ABC-Fit2C	37.43	98.22	–		ABC-Fit2C	7.73	77.88	–
	BCOA-2S	20.45	100	100.00		BCOA-2S	9	91.56	91.56
German	BCOA-FS	12.5	86.21	72.48	Musk1(Clean1)	BCOA-FS	86.48	88.81	84.58
	ABC-ER	10.76	70.17	–		ABC-ER	83.03	83.11	–
	COA-FS	12.1	86.21	74.25		COA-FS	85.6	88.81	86.21
	COA-2S	11.5	86.21	75.45		COA-2S	41.25	88.97	86.21
	ABC-Fit2C	9.13	70.01	–		ABC-Fit2C	80.56	82.23	–
	BCOA-2S	11.2	86.21	75.70		BCOA-2S	50.1	90.50	86.21
Ionosphere	BCOA-FS	12.56	88.82	86.21	Hillvalyey	BCOA-FS	45.25	68.52	65.22
	ABC-ER	12	92.12	–		ABC-ER	47.63	54.13	–
	COA-FS	12.12	92.46	86.21		COA-FS	46.12	64.26	59.62
	COA-2S	11.85	94.26	86.21		COA-2S	44.6	70.65	68.56
	ABC-Fit2C	11.53	91.74	–		ABC-Fit2C	44.96	54.92	–
	BCOA-2S	10.65	98.25	86.21		BCOA-2S	42.4	74.25	70.35
Madelon	BCOA-FS	255.5	80.00	80.00					
	ABC-ER	252.46	72.91						
	COA-FS	259.2	79.89	77.47					
	COA-2S	250.6	81.45	79.45					
	ABC-Fit2C	248.03	72.20	–					
	BCOA-2S	248.52	83.45	81.65					

Table 7 Comparison of (BCOA-FS, COA-FS, BCOA-2S and COA-2S) with BAIS

Datasets	Approach	Ave-Size	Best-Acc	Ave- Acc	Datasets	Approach	Ave-Size	Best-Acc	Ave-Acc
WineEW	BCOA-FS	8	86.21	96.75	Mushroom	BCOA-FS	12.5	98.98	95.65
	COA-FS	8	86.21	96.75		COA-FS	13.5	98.98	94.75
	COA-2S	7	86.21	97.25		COA-2S	8	100.00	98.60
	BAIS	7.8	98.40	–		BCOA-2S	7	100.00	99.80
	BCOA-2S	7	100.00	86.21		BAIS	11.5	98.10	–
Ionosphere	BCOA-FS	12.56	88.82	86.21	WBCD	BCOA-FS	13.41	95.21	91.28
	COA-FS	12.12	92.46	86.21		COA-FS	13.52	94.75	92.45
	COA-2S	11.85	94.26	86.21		COA-2S	13.21	98.22	98.22
	BAIS	13.4	91.20	–		BAIS	14.3	97.20	–
	BCOA-2S	10.65	98.25	86.21		BCOA-2S	12.45	100.00	100.00
Sonar	BCOA-FS	23.42	86.21	90.65					
	COA-FS	24.5	86.21	86.45					
	COA-2S	21.9	86.21	96.58					
	BAIS	23.6	77.30	–					
	BCOA-2S	19.8	86.21	98.20					

Even though BCOA is a discrete binary version of COA, however, it can be seen that it outperformed COA not only on the categorical or discrete data sets but also on the continuous data sets. Continuous or discrete data sets refer to the data sets that have their class label either as categorical or continuous.

Analysis of the computational time also shows that BCOA can complete its evolutionary process within the shortest time than the COA on the majority of the data sets. BCOA is faster than COA in around 10–5% majority of some of the data sets regardless of the continuous or categorical data sets. Meanwhile, this motivates the use of BCOA alone in the multi-objective wrapper-based feature selection. Moreover, this will avoid repetition of similar explanation of BCOA of being the best compared to its COA counterpart.

5.6 Further Discussions

The results show that both BCOA-FS and COA-FS can successfully evolve a set of features with better classification performance within a short period. However, as the number of features increase, BCOA-FS perform better than COA-FS, especially on the categorical datasets. Whereas, the COA-FS performed better mostly on the continuous class label dataset. It demonstrates that the continuous version works well on the continuous label datasets. In contrast, the binary version works well on the majority of the datasets and mostly performed better on the categorical datasets. Correspondingly, both BCOA-2S and COA-2S can successfully select the best features with better classification performance than the COA-FS and BCOA-2S on the majority of the datasets. Also, BCOA-2S outperformed COS-2S in most cases due to the use of the two-step evaluation process.

The proposed approaches used a β value of [0,1] in the evolution process. However, choosing the most appropriate value is quite a challenge. Because most of the selected features, along with their classification performance, are combined into a single fitness function. Nowadays, FS is considered as a multi-objective optimisation problem and treating the FS in that regards will solve the task much better and obtain the set of nondominated solutions.

6 Conclusions and Future Work

This paper disclosed the first study on wrapper-based feature selection using COA and BCOA. Four wrapper-based feature selections are presented. Both BCOA and COA were adopted and used as a wrapper based in the evolutionary process. Then a two-step fitness function was proposed, whereby the new classification performance obtained in the first step is combined with the number of selected features in the second step. The results obtained showed that the proposed methods performed well compared to the previous work. However, combining the two aims of the feature

selection into a single fitness function cannot solve the problem better, and there will be some redundancy still among the number of selected features. Hence there is need for multi-objective feature selection that treats both numbers of selected features and classification performance simultaneously.

On the other hand, COA, especially its binary version, has performed well for FS because (1. COA representation is suitable for FS problems. The habitats in COA is N_{var}-dimensional array representing the current living position of cuckoos, which looks like the way candidate solutions are represented in the FS problem. In this case, the size of the dimensionality is the number of features. The values in any dimension/habitat display whether a feature is chosen or otherwise. (2. The search space in FS problems is too large and mostly get stuck in local optima in most of the existing methods. As such, there is a need for a global search technique. These ECs are well-known for solving problems that do not have a solution; they are robust to dynamic changes and have broad applicability [14, 49, 50]. COA is an EC; precisely an evolutionary algorithm based that has effective and efficient search operators that can search for large space to discover the optimum otherwise nearby optimum solution [14].

Acknowledgements The authors want to thank Universiti Sains Malaysia (USM) for supporting and backing the research via its Research University Grant (RUI) (1001/PKOMP/8014084) along with Woosong University, Korea.

References

1. Aha, D.W., Bankert, R.L.: A comparative evaluation of sequential feature selection algorithms. In: Learning from Data, pp. 199–206. Springer (1996)
2. Ahmad, S. et al.: Feature selection using salp swarm algorithm with chaos. In: ICFNDS '18 Proceedings of the 2nd International Conference on Intelligent Systems, Metaheuristics & Swarm Intelligence, pp. 65–69. ACM (2018)
3. Alickovic, E., Subasi, A.: Breast cancer diagnosis using GA feature selection and Rotation Forest. Neural Comput. Appl. **28.4**, 753–763 (2017). ISSN: 0941-0643
4. Alpaydin, E.: Introduction to Machine Learning. MIT Press (2014). ISBN 0262325756
5. Amiri, E., Mahmoudi, S.: Efficient protocol for data clustering by fuzzy cuckoo optimization algorithm. Appl. Soft Comput. **41**, 15–21 (2016). ISSN: 1568-4946
6. Anemangely, M. et al.: Machine learning technique for the prediction of shear wave velocity using petrophysical logs'. J. Petrol. Sci. Eng. **174**, 306–327 (2019). ISSN: 0920-4105
7. Brezocnik, L., Fister, I., Podgorelec, V.: Swarm intelligence algorithms for feature selection: a review. Appl. Sci. **8**(9), 1521 (2018)
8. Caruana, R., Freitag, D.: Greedy attribute selection. In: Machine Learning Proceedings 1994, pp. 28–36. Elsevier (1994)
9. Castro, P.A.D., Von Zuben, F.J.: Feature subset selection by means of a Bayesian artificial immune system. In: 2008 Eighth International Conference on Hybrid Intelligent Systems, pp. 561–566. IEEE (2008). ISBN: 0769533264
10. Castro, P.A.D., Von Zuben, F.J.: Multi-objective feature selection using a Bayesian artificial immune system. Int. J. Intell. Comput. Cybern. **3.2**, 235–256 (2010). ISBN: 1756-378X
11. Dobbin, K.K., Simon, R.M.: Optimally splitting cases for training and testing high dimensional classifiers. In: BMC Medical Genomics, vol. 4.1, pp. 1–8 (2011). ISSN: 1755-8794

12. Dy, J.G., Brodley, C.E.: Feature selection for unsupervised learning. J. Mach. Learn. Res. **5**(8), 845–889 (2004)
13. El Aziz, M.A., Hassanien, A.E.: Modified cuckoo search algorithm with rough sets for feature selection. Neural Comput. Appl. **29.4**, 925–934 (2018). ISSN: 0941-0643
14. Elyasigomari, V. et al.: Cancer classification using a novel gene selection approach by means of shuffling based on data clustering with optimization. Appl. Soft Comput. **35**, 43– 51 (2015). ISSN: 1568-4946
15. Faris, H. et al.: An efficient binary Salp Swarm Algorithm with crossover scheme for feature selection problems. Knowl.-Based Syst. **154**, 43–67 (2018). ISSN: 0950-7051
16. Ferri, F.J. et al.: Comparative study of techniques for large-scale feature selection. In: Machine Intelligence and Pattern Recognition, vol. 16, pp. 403–413. Elsevier (1994). ISBN: 0923-0459
17. Frank, A., Asuncion, A.: UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. In: School of Information and Computer Science, vol. 213, pp. 21–22. University of California, Irvine, CA (2010)
18. Gadekallu, T.R., Khare, N.: Cuckoo search optimized reduction and fuzzy logic classifier for heart disease and diabetes prediction. Int. J. Fuzzy Syst. Appl. (IJFSA) **6**(2), 25–42 (2017)
19. Ghareb, A.S., Bakar, A.A., Hamdan, A.R.: Hybrid feature selection based on enhanced genetic algorithm for text categorization. In: Expert Systems with Applications vol. 49, pp. 31–47 (2016). ISSN: 0957-4174
20. Gheisarnejad, M.: An effective hybrid harmony search and cuckoo optimization algorithm based fuzzy PID controller for load frequency control. Appl. Soft Comput. **65**, 121– 138 (2018). ISSN: 1568-4946
21. Gheyas, I.A., Smith, L.S. (2010). Feature subset selection in large dimensionality domains. Pattern Recogn. **43.1**, 5–13 (2010). ISSN: 0031-3203
22. Goli, A., Aazami, A., Jabbarzadeh, A.: Accelerated cuckoo optimization algorithm for capacitated vehicle routing problem in competitive conditions. Int. J. Artif. Intell. **16**(1), 88–112 (2018)
23. Gonzalez, J. et al.: A new multi-objective wrapper method for feature selection–Accuracy and stability analysis for BCI. Neurocomputing **333**, 407–418 (2019). ISSN: 0925-2312
24. Guo, G. et al.: An kNN model-based approach and its application in text categorization. In: International Conference on Intelligent Text Processing and Computational Linguistics, pp. 559–570. Springer (2004)
25. Gutlein, M. et al.: Large-scale attribute selection using wrappers. In: 2009 IEEE Symposium on Computational Intelligence and Data Mining, pp. 332–339. IEEE (2009). ISBN: 1424427657
26. Hancer, E., Xue, B., Zhang, M.: Differential evolution for filter feature selection based on information theory and feature ranking. Knowl.-Based Syst. **140**, 103– 119 (2018). ISSN: 0950-7051
27. Hancer, E. et al.: Pareto front feature selection based on artificial bee colony optimization. Inf. Sci. **422** , 462–479 (2018). ISSN: 0020-0255
28. Hastie, T., et al.: The elements of statistical learning: data mining, inference and prediction. Math. Intell. **27**(2), 83–85 (2005)
29. Hosseini-Moghari, S.-M. et al.: Optimum operation of reservoir using two evolutionary algorithms: imperialist competitive algorithm (ICA) and cuckoo optimization algorithm (COA). Water Res. Manag. **29.10**, 3749–3769 (2015). ISSN: 0920-4741
30. Huang, C.-L., Wang, C.-J.: A GA-based feature selection and parameters optimizationfor support vector machines. Expert Syst. Appl. **31.2**, 231–240 (2006). ISSN: 0957-4174
31. Hannah Inbarani, H., Azar, A.T., Jothi, G.: Supervised hybrid feature selection based on PSO and rough sets for medical diagnosis. Comput. Methods Prog. Biomed. **113.1**, 175–185 (2014). ISSN: 0169-2607
32. Hannah Inbarani, H., Bagyamathi, M., Azar, A.T.: A novel hybrid feature selection method based on rough set and improved harmony search". In: Neural Comput. Appl. **26.8**, 1859–1880 (2015). ISSN: 0941-0643
33. Jiménez, F. et al.: Multi-objective evolutionary feature selection for online sales forecasting. Neurocomputing **234**, 75–92 (2017). ISSN: 0925-2312

34. Kelemen, A. et al.: Naive Bayesian classifier for microarray data. In: Proceedings of the International Joint Conference on NeuralNetworks, vol. 2003, pp. 1769–1773. IEEE (2003). ISBN: 0780378989
35. Khabiri, M., Ghaffari, A.: Energy-Aware clustering-based routing inwireless sensor networks using cuckoo optimization algorithm. Wirel. Pers. Commun. **98.3**, 2473–2495 (2018). ISSN: 0929-6212
36. Kittler, J.: Feature selection and extraction. Handbook of Pattern recognition and image processing **1**(1), 1–37 (1986)
37. Koza, J.R. et al.: Genetic programming 1998: Proceedings of the Third Annual Conference. In: IEEE Transactions on Evolutionary Computation, vol. 3.2, pp. 159–161 (1999). ISSN: 1089-778X
38. Kuncheva, L.I., Jain, L.C.: Nearest neighbor classifier: simultaneous editing and feature selection. Pattern Recogn. Lett. **20.11**, 1149–1156 (1999). ISSN: 0167-8655
39. Liu, H., Motoda, H.: Feature extraction, construction and selection: a data mining perspective. Springer Science and Business Media (1998). ISBN: 0792381963
40. Liu, H., Motoda, H.: Feature selection for knowledge discovery and data mining. Springer Science and Business Media (2012). ISBN: 1461556899
41. Liu, H., Yu, L.: Toward integrating feature selection algorithms for classification and clustering. IEEE Trans. Knowl. Data Eng. 17.4 (2005), pp. 491–502. issn: 1041-4347
42. Liu, X.-Y. et al.: A hybrid genetic algorithm with wrapper-embedded approaches for feature selection. In: IEEE Access **6**, 22863–22874 (2018). ISSN: 2169-3536
43. Mafarja, M. et al.: Evolutionary population dynamics and grasshopper optimization approaches for feature selection problems. Knowl.-Based Syst. **145**, 25–45 (2018). ISSN: 0950-7051
44. Mafarja, M. et al.: Feature selection using binary particle swarm optimization with time varying inertiaweight strategies. In: ICFNDS'18 Proceedings of the 2nd International Conference on Future Networks and Distributed Systems, pp. 1–9. ACM (2018)
45. Mahmoudi, S., Rajabioun, R., Lotfi, S.: Binary cuckoo optimization algorithm. In: 1st National Conference on New Approaches in Computer Engineering and Information Retrieval Young Researchers And Elite Club of the Islamic Azad University, Roudsar-Amlash Branch, pp. 1–7 (2013)
46. Peng, H., Long, F., Ding, C.: Feature selection based on mutual information criteria of max-dependency, maxrelevance, and min-redundancy. IEEE Trans. Pattern Anal. Machine Intell. **27.8**, 1226–1238 (2005). ISSN: 0162-8828
47. Pudil, P., Novovicov, J., Kittler, J.: (1994). Floating search methods in feature selection. Pattern Recogn. Lett. **15.11**, 1119–1125 (1994). ISSN: 0167-8655
48. Purohit, A., Chaudhari, N.S., Tiwari, A.: Construction of classifier with feature selection based on genetic programming. In: 2010 IEEE Congress on Evolutionary Computation (CEC), pp. 1–5. IEEE (2010). ISBN: 1424469112
49. Rajabioun, R.: Cuckoo optimization algorithm. Appl. Soft Comput. **11.8**, 5508–5518 (2011). ISSN: 1568-4946
50. Sivanandam, S.N., Deepa, S.N.: Genetic algorithm optimization problems. Introduction to Genetic Algorithms, pp. 165–209. Springer (2008)
51. Song, L. et al.: Supervised feature selection via dependence estimation. In: Proceedings of the 24th International Conference on Machine Learning, pp. 823–830. ACM (2007). isbn: 1595937935
52. Tran, B., Xue, B., Zhang, M.: A newrepresentation in PSO for discretization-based feature selection. In: IEEE Transactions on Cybernetics **48.6**, 1733–1746 (2018). ISSN: 2168-2267
53. Unler, A., Murat, A.: A discrete particle swarm optimization method for feature selection in binary classification problems. Eur. J. Oper. Res. **206.3**, 528–539 (2010). ISSN: 0377-2217
54. Unler, A., Murat, A., Chinnam, R.B.: mr 2 PSO: a maximum relevance minimum redundancy feature selection method based on swarm intelligence for support vector machine classification. Inf. Sci. **181.20**, 4625–4641 (2011). ISSN: 0020-0255
55. Usman, A.M., Yusof, U.K., Naim, S.: Cuckoo inspired algorithms for feature selection in heart disease prediction. Int. J. Adv. Intell. Inf. **4.2**, 95–106 (2018). ISSN: 2548-3161

56. Usman, A.M. et al.: Comparative evaluation of nature-based optimization algorithms for feature selection on some medical datasets. I-manag. J. Image Process. **5.4**, 9 (2018). ISSN: 2349-4530
57. Vapnik, V.N.: An overview of statistical learning theory. IEEE Trans. Neural Netw. **10.5**, 988–999 (1999). ISSN: 1045-9227
58. Vieira, S.M. et al.: Modified binary PSO for feature selection using SVM applied to mortality prediction of septic patients. Appl. Soft Comput. **13.8**, 3494–3504 (2013). ISSN: 1568- 4946
59. Welikala, R.A. et al.: Genetic algorithm based feature selection combined with dual classification for the automated detection of proliferative diabetic retinopathy. Comput. Med. Imaging Graph. **43**, 64–77 (2015). ISSN: 0895–6111
60. Whitney, A.W.: A direct method of nonparametric measurement selection. IEEE Trans. Comput. **100.9**, 1100–1103 (1971). ISSN: 0018-9340
61. Xu, Z. et al.: Discriminative semi-supervised feature selection via manifold regularization. IEEE Trans. Neural Netw. **21.7**, 1033–1047 (2010). ISSN: 1045-9227
62. Xue, B., Zhang, M., Browne, W.N.: Particle swarm optimisation for feature selection in classification: novel initialisation and updating mechanisms. Appl. Soft Comput. **18**, 261–276 (2014). ISSN: 1568-4946
63. Xue, B., Zhang, M., Browne, W.N.: Particle swarmoptimization for feature selection in classification: a multi-objective approach. IEEE Trans. Cybern. **43.6**, 1656–1671 (2012). ISSN: 2168-2267
64. Xue, B. et al.: A survey on evolutionary computation approaches to feature selection. IEEE Trans. Evol. Comput. **20.4**, 606–626 (2016). ISSN: 1089-778X
65. Xue, B. et al.: Multi-objective evolutionary algorithms for filter based feature selection in classification. Int. J. Artif. Intell. Tools **22.4**, 1–31 (2013). ISSN: 0218-2130
66. Xue, X., Yao, M., Wu, Z.: A novel ensemblebased wrapper method for feature selection using extreme learning machine and genetic algorithm. Knowl. Inf. Syst. **57.389**, 389–412 (2017). ISSN: 0219-1377
67. Yusta, S.C.: Different metaheuristic strategies to solve the feature selection problem. Pattern Recogn. Lett. **30.5**, 525–534 (2009). ISSN: 0167-8655
68. Zhao, H., Sinha, A.P., Ge, W.: Effects of feature construction on classification performance: an empirical study in bank failure prediction. Expert Syst. Appl. **36.2**, 2633–2644. (2009) ISSN: 0957-4174