

RESEARCH

Open Access



A novel community detection based genetic algorithm for feature selection

Mehrdad Rostami^{1*} , Kamal Berahmand² and Saman Forouzandeh³

*Correspondence:

M.rostami@eng.uok.ac.ir

¹ Department of Computer Engineering, University of Kurdistan, Sanandaj, Iran
Full list of author information is available at the end of the article

Abstract

The feature selection is an essential data preprocessing stage in data mining. The core principle of feature selection seems to be to pick a subset of possible features by excluding features with almost no predictive information as well as highly associated redundant features. In the past several years, a variety of meta-heuristic methods were introduced to eliminate redundant and irrelevant features as much as possible from high-dimensional datasets. Among the main disadvantages of present meta-heuristic based approaches is that they are often neglecting the correlation between a set of selected features. In this article, for the purpose of feature selection, the authors propose a genetic algorithm based on community detection, which functions in three steps. The feature similarities are calculated in the first step. The features are classified by community detection algorithms into clusters throughout the second step. In the third step, features are picked by a genetic algorithm with a new community-based repair operation. Nine benchmark classification problems were analyzed in terms of the performance of the presented approach. Also, the authors have compared the efficiency of the proposed approach with the findings from four available algorithms for feature selection. Comparing the performance of the proposed method with three new feature selection methods based on PSO, ACO, and ABC algorithms on three classifiers showed that the accuracy of the proposed method is on average 0.52% higher than the PSO, 1.20% higher than ACO, and 1.57 higher than the ABC algorithm.

Keywords: Machine learning, Feature selection, Genetic algorithm, Graph theory, Multi-objective

Introduction

Datasets have evolved significantly in recent years with developments in science and technology and now involve numerous features. Methods of pattern detection are therefore engaged in samples with thousands of features. Consequently, reducing their dimensionality is essential for the traceability of data sets [1, 2]. High-dimensional vectors impose significant computational costs and also the risk of overfitting [3–5]. Generally, a minimum of $10 \times D \times C$ training examples is necessary for a classification problem with D dimensions and C classes [6]. Whenever the needed number of training examples cannot be provided, reducing features decreases the size of the needed training examples and hence increases the overall yield shape of the classification algorithm. In the previous years, two methods for dimensional reduction were presented: feature

selection and feature extraction [7–10]. Feature selection seeks for a relevant subset of existing features, while features are designed for a new space of lower dimensionality in the feature extraction method. Both methods for the reduction of dimensionality are designed to improve learning efficiency, minimize computational complexity, develop more generalizable models, and reduce needed storage [11–15].

Feature selection has been an active research area in data mining, pattern recognition, and statistics communities [16–20]. The total search space to find the most relevant and non-redundant features, including all possible subsets, is 2^n , where n is the number of original features [21, 22]. Comprehensive search ensures that the most appropriate features are found, but usually, this is not computationally feasible, even for medium-sized datasets [23, 24]. Since the evaluation of all possible subsets is very costly, a solution must be searched that is both computationally feasible and useful in terms of quality. Many feature selection methods use metaheuristic algorithms to avoid increasing computational complexity [25–27]. These algorithms will be able to optimize the problem of feature selection with appropriate accuracy within an acceptable time.

Techniques of optimization based on the population including ant colony optimization (ACO) [28], genetic algorithm (GA) [21], simulated annealing (SA) [29], taboo search (TS) [30], and particle swarm optimization (PSO) [31] were recently used in feature selection. In fact, hybrid search strategies have been used that merge the wrapper and filter approaches. In [32], the suggestion was made for the use of a hybrid filter wrapper subset selection algorithm based on the PSO for the classification of Support Vector Machines (SVM). In addition, some existing techniques take into account the connection of features in their search strategies. For instance, in [33], an enhanced genetic algorithm was proposed for the optimum selection of a feature subset from a multi-character set. This approach separates the chromosome into many classifications for local management. Various mutation and crossover operators are then used on mentioned categories to eliminate invalid chromosomes. In recent decades, many Evolutionary algorithm-based algorithms such as Genetic Algorithm (GA), Particle Swarm Optimization (PSO), Ant Colony Optimization (ACO), and Artificial Bee Colony (ABC) have been employed to feature section. Among the SI-based algorithm, Genetic has been efficiently utilized in the feature selection problem to redact of high-dimensional dataset. One of the disadvantages of this method is that it does not consider the connections among the features when selecting the final features. As a result, the probability of selecting a subset with redundancy will increase. To overcome these drawbacks, the present paper introduces a community-based genetic algorithm for the selection of features named CGAFS. A community detection method is used in the proposed approach for dividing features into various groups. Hence a new mutation step named “repair operations” is introduced to fix the chromosome by utilizing predetermined feature clusters. A newly produced offspring shall be repaired to eliminate related features in the offspring. In comparison to the previous genetic algorithm-based feature selection that apply filters and wrappers models in the order, the community detection technique is integrated into the GA-based wrapper model in a structural manner. Furthermore, the cluster number and the optimum size of the subset could also be calculated automatically. The proposed GA-based feature selection methods have several novelties compared to the well-known and state-of-the-art GA-based feature selection methods:

- The proposed method uses a novel community detection-based algorithm to identify the feature clusters to group similar features. Grouping similar features prevent the proposed method to select redundant features. Unlike the other clustering methods such as k-means [34] and fuzzy c-means [35], the proposed clustering method identifies the number of clusters automatically, and there is no longer a need to determine the number of clusters in advance.
- The proposed method uses a community detection-based repair operation that considers both the local and global structure of the graph in computing similarity values. In other words, it takes into account implicit and explicit similarities between features, while the other feature selection methods only take into account the direct similarities between features.
- The number of final selected features imposes another challenge on feature selection methods. In other words, the number of relevant features is unknown; thus, the optimal number of selected features is not known either. In this method, unlike many previous works, the optimal number of selected features is determined automatically based on the overall structure of the original features and their inner similarities.
- The proposed method groups similar features into the clusters and then applies a multi-objective fitness function to assign an importance value to each feature subset. In the proposed multi-objective fitness function, two objectives of feature relevance and feature redundancy are considered, simultaneously. Unlike the other multi-objective methods that identify a set of non-dominated solutions in an iterative process [36, 37], the proposed method finds the near-optimal solution in a reasonable time.

The rest of the present article is structured as the following: “[Related Work](#)” section analyses research on the selection of features; in “[Proposed method](#)” section, the proposed selection algorithm is presented; in “[Experimental results](#)” section, the comparison of the proposed algorithm other feature selection algorithms is discussed. Ultimately, in “[Discussion](#)” section, the authors summarize the present study.

Related work

For several practical applications, including text processing, face recognition, image retrieval, medical diagnosis, and bioinformatics, feature selection was developed as a central procedure [38–40]. Feature selection was a promising area of research and development for statistical pattern detection, data mining, and machine learning since the 1970s, and many efforts have been made to evaluate the methods of feature selection, which may be divided into four groups, namely, filters, wrappers, hybrids and embedded depending on the evaluation process [41–44]. Whenever a procedure performs a feature selection independently of any learning algorithm (e.g., an entirely independent preprocessor), afterward it is included in the filter method classification. The statistical analysis is required for the filter approach of the feature set that can only be used to solve the feature selection problem without using a learning model. Conversely, a predetermined learning algorithm is used by the wrapper approach to identify the quality of the selected subsets. However, wrappers can yield stronger results; they are costly to operate and can disintegrate with too many features. The hybrid approach combines the filter and

wrapper technique and seeks to incorporate the filter and wrapper methods. Ultimately, the embedded techniques take advantage of the selection of features in the learning process as well as are highly comparable to a certain learning model [45, 46].

Depending on the availability of training data class labels, feature selection algorithms could also be classified into two parts: supervised feature selection and unsupervised feature selection [47, 48]. The supervised feature selection is employed in the case that class labels of the data are obtainable, differently the unsupervised feature selection seems to be suitable. In general, the supervised feature selection generates better and more efficiency, primarily because of the use of class labels [49–51].

From another view, filter methods are classified into ranking-based and Subset Selection-Based (SSB) methods. Ranking-based methods first assign a relevance value to each feature using a univariate or a multivariate criterion, and then sort the features and select those of the top high scores. Although the ranking-based methods require low computational resources, all these methods consider only the relevancy of the features and neglect the redundancy with others. Identifying a set of optimal feature subset that results in building a learning model with maximum accuracy is an NP-hard problem. To overcome this issue, the subset selection-based methods seek to find a near-optimal feature set by applying some heuristic or meta-heuristic methods. For example, Relevance redundancy feature selection [52], MIFS [53], Normalized mutual information feature selection [54], MIFS-U [55], MIFS-ND [56], JMIM [57], OSFMI, and MRDC [58] use sequential forward or backward selection as a type of greedy search strategy, and thus they easily trap into a local optimum.

The search space includes all feasible feature subsets to discover the best feature subset, indicating that the search space is as the following:

$$\sum_{s=0}^n \binom{n}{s} = \binom{n}{0} + \binom{n}{1} + \dots + \binom{n}{n} = 2^n \quad (1)$$

where n (quantity of original features) is the dimensionality and s is the size of the current subset of features. Thus, the problem to discover the ideal feature subset seems to be NP-hard. Because the analysis of the whole feature subsets is costly in a computational manner, time-consuming, and also inefficient even in small sizes, solutions are required that are computationally efficient and that provide a reasonable tradeoff among time–space cost and strength of the solution [11, 59, 60]. Most feature selection algorithms also include random or heuristic search techniques to minimize the computation period [59, 61, 62].

One approach to solving complex optimization and NP-Hard problems is meta-heuristics algorithms. Meta-heuristic algorithms are approximate approaches that can find satisfactory solutions over an acceptable time instead of finding the optimal solution [63]. These algorithms are one of the categories of approximate optimization algorithms that have strategies to escape from local optima and can be used in a wide range of optimization problems.

Many feature selection methods use meta-heuristics to avoid increasing computational complexity in the high dimensional dataset. These algorithms use primitive

mechanisms and operations to solve an optimization problem and search for the optimal solution over several iterations [49]. These algorithms often start with a population containing random solutions and try to improve the optimality of these solutions during each iteration step. At the beginning of most of the meta-heuristic algorithms, a number of initial solutions are randomly generated, and then a fitness function is utilized to calculate the optimality of the individual solutions of the generated population. If none of the termination criteria are met, production new generation will begin. This cycle is repeated until one of the termination criteria is met [64, 65].

Meta-heuristic approaches can be classified into two categories: Evolutionary Algorithms (EA) and Swarm Intelligence (SI) [63]. An EA uses mechanisms inspired by biological evolution, such as reproduction, mutation, recombination, and selection. Candidate solutions to the optimization problem play the role of individuals in a population, and the fitness function determines the quality of these solutions. After repetitions of the evolutionary algorithm, the initial population evolves and moves toward global optimization [66]. On the other, SI algorithms usually consist of a simple population of artificial agents locally with the environment. This concept is usually inspired by nature, and each agent performs an easy job, but local interactions and partly random interactions between these agents lead to the emergence of “intelligent” global behavior, which is unknown to individual agents [67].

In [68], a k-Nearest-Neighbors technique, for which a genetic algorithm is utilized for the efficient feature selection to decrease the dataset dimensions and improve the classification accuracy, is employed for diagnosing the stage of patients’ disease. Moreover, in [69] a new two-layer feature selection approach that combines a wrapper and an embedded method in constructing an appropriate subset of predictors is proposed. In the first layer of this technique, the Genetic Algorithm has been adopted as a wrapper to search for the optimal subset of predictors, which aims to reduce the number of predictors and the prediction error. Then a second layer is added to the proposed technique to eliminate any remaining redundant/irrelevant predictors to improve the prediction accuracy. Rathee and Ratnoo [70] proposed a genetic algorithm-based multi-objective method for feature selection. This method combines the idea of non-dominated sorting with a genetic algorithm to arrive at a set of non-dominated solutions. Furthermore, in [71] an ensemble feature selection method based on t-test and genetic algorithm is developed. In this method after t-test-based data preprocessing, a Nested Genetic Algorithm, is utilized to get the optimal subset of features by combining data from two different datasets. Nested-GA consists of two Nested Genetic Algorithms that run on two different kinds of datasets.

In [72], a novel hybrid PSO-based feature selection method for the analysis of Laser-induced breakdown spectroscopy is introduced. In this method, an attempt has been made to use the advantages of coating and filter methods simultaneously. In [73] a PSO-based feature selection with multiple classifiers is proposed to improve for increasing the classification accuracy and reducing computational complexity. In this paper, a new Self-Adaptive Parameter and Strategy are used to deal with the issue of feature selection in a high-dimensional dataset. The reported results showed that the use of these mechanisms greatly increased the search ability of particle optimization algorithms for high-dimensional datasets. Moreover, in [31], a novel graph-based feature selection method

is developed to increase disease diagnosis accuracy. In this method, using the node centrality criterion, a new mechanism for initializing the particles is proposed. Then, by defining a multi-objective fitness function, a subset of the final features that are least similar to each other and most relevant to the target class are selected. Finally, based on the selected features, the disease is diagnosed.

In [48] a novel ACO-based feature selection method is proposed for unsupervised mode. The authors of this paper selected the most non-redundant features that have the least similarity with each other. Moreover, Moradi and Rostami [28], developed a filter-based feature selection approach utilizing the ACO algorithm and graph clustering. This approach represented the feature space as a clustered graph. Then, according to the similarity between the features and by defining a filter criterion, it selects a dissimilar and related subset of the features in [74] proposed an unsupervised ACO-based feature selection method to remove redundant and irrelevant features. This method tries to select an optimal subset of features in a hierarchical process, by considering the similarity between features. In [75] the combination of feature selection and ant colony optimization is proposed to improve the classification accuracy of imbalanced data. In this method, instead of using a single-objective fitness function, a multi-objective ant colony optimization algorithm is used to improve the performance feature selection. The reported results showed acceptable performance of the proposed method in classifying imbalanced and high-dimensional datasets.

In [76] a Multi Hive ABC Programming is developed to select the final feature set in high dimensional datasets. This approach utilized the ability of an automatic programming algorithm to remove irrelevant and redundant features. The authors of [77], developed a multi-objective ABC-based feature selection approach. In this method, two new operators are used to improve its search capability and convergence of the ABC search strategy. In [78], an ABC-based feature selection is proposed by integrating of multi-objective optimization algorithm with a sample reduction strategy. This proposed method has both increased classification accuracy and reduced computational complexity.

Proposed method

For real-world datasets, there are a vast number of irrelevant and redundant features, which may significantly degrade the performance of the model learned and the learning speed of the models. Feature selection is an essential step in data preprocessing in data mining to remove irrelevant and redundant features of a given dataset. Many technologies can easily eliminate irrelevant features from the other feature subset selection methods, but do not handle redundant features. Many often only eliminate redundant features. With the redundant features, the presented algorithm will remove the irrelevant.

The authors consider a hybrid method based on a combination of a community detection approach and the genetic algorithm, in the context of the hybrid approaches to the feature selection problem. Genetic algorithms are methods of optimization focused on the natural selection process. John Holland initially introduced GAs to describe the adaptation mechanisms of the natural systems and to develop new artificial structures on identical principles. This imitates the natural selection method

and begins with artificial individuals (represented by a ‘chromosome’ population). GA attempts to improve the fitters using genetic operators (e.g., crossover and mutation). In addition, it seeks to produce chromosomes in a certain quantitative measure, which are stronger compared to their parents. Hence, GA has recently been widely used as a tool for data mining feature selection.

In theory, it was shown that genetic algorithms could randomly seek the optimal solution for a problem. Simple genetic algorithms, however, have some shortcomings such as premature convergence, poor ability of fine-tuning near local optimum points in applications. On the other side, certain other techniques of optimizing, including the steepest descent method, simulated annealing, and hill-climbing generally include strong local search ability. Moreover, some heuristic algorithms have a strong performance with issue-specific information. Furthermore, some hybrid GAs for feature selection was established by incorporating the optimization methods or heuristic algorithms, as mentioned above, to improve the fine-tuning capabilities and performance of simple GAs. In the present study, the authors suggest a new genetic algorithm of clustering for feature selection issues, in which the connection and repair of this feature are used for the selection of candidate features.

Application of the hybrid genetic algorithm for the selection of features typically involves chromosome encoding schemes, fitness function estimation, fitter chromosome selection, genetic crossover and mutation operations, and stoppage criterion. The suggested approach provides a candidate solution to the problem of subset selection in the chromosome population. A chromosome is encoded with binary digit series that “1” means “selected” and “0” means “unselected.” Every digit (or gene) correlates to a feature so that the chromosome gene length is equivalent to the total of input features available. The methods for genetic operations are as follows. Initially, the design proposed in the present article uses the roulette wheels’ selection process. Next, an adaptive crossover approach is applied. The single-point crossover operator is utilized where the overall number of features in a specified dataset is less than 20; whereas the overall number of functions is greater than 20, double-point crossover procedures are used.

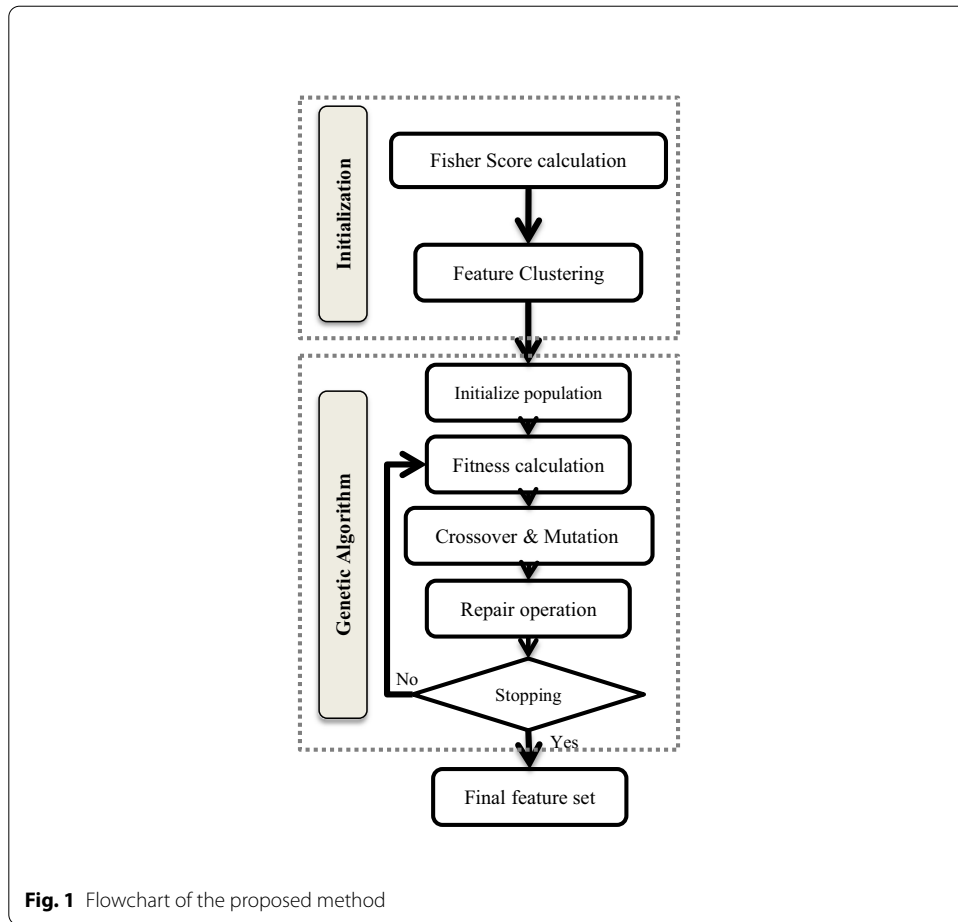
The main steps of the Community Detection-based Genetic Algorithm for Feature selection (CDGAFS) are summarized in Fig. 1. In addition, in its corresponding subscription, every stage of the CDGAFS is defined.

Step 1: Measure the relevance of features:

For measuring the discriminatory power of the features, the discrimination ability of the feature F_i is measured by applying the Fisher score as the following:

$$Score_i = \frac{\sum_{k=1}^C n_i (\bar{x}_i^k - \bar{x}_i)^2}{\sum_{k=1}^C n_i (\sigma_i^k)^2} \quad (2)$$

where, C implies the number of classes of the dataset; n_i is referred to as the number of samples in class i , \bar{x}_i indicates the mean of all the patterns according to the feature F_i , as well as \bar{x}_i^k and σ_i^k imply mean and variance of class k corresponding to the feature F_i . A



larger $Score_i$ value shows that the feature F_i possesses a higher discriminative capability. In most instances, fisher score values of features are near each other. In order to conquer this situation, a non-linear normalization approach named softmax scaling has been applied for scaling the edge weight into the range [0 1] as the following:

$$\widehat{Score}_i = \frac{1}{1 + \exp\left(-\frac{Score_i - \overline{Score}}{\sigma}\right)} \tag{3}$$

where $Score_i$ indicates the fisher score of the feature F_i , \overline{Score} and σ imply the variance and mean of all of the fisher score values, respectively, as well as \widehat{Score}_i shows normalized fisher score value of the feature F_i .

Step 2: Feature clustering:

In general, to apply any feature clustering algorithm, the similarity between the features must be calculated [14, 15]. Due to the fact that graph-based clustering techniques are used in this paper, the feature space is represented as a graph. For this purpose, the mapping of the feature set into its equivalent graph $G = (F, E, w_F)$ was done, where $F = \{F_1, F_2, \dots, F_n\}$ implies a set of original features, $E = \{(F_i, F_j) : F_i, F_j \in F\}$ are the

edges of the graph and w_{ij} is referred to as the similarity among two features F_i and F_j which were connected by the edge (F_i, F_j) . In the present article, the Pearson correlation coefficient measure has been applied to calculate the similarity value among various features of a provided training set. The relationship between the two features F_i and F_j is defined as the following:

$$w_{ij} = \left| \frac{\sum_p (x_i - \bar{x}_i)(x_j - \bar{x}_j)}{\sqrt{\sum_p (x_i - \bar{x}_i)^2} \sqrt{\sum_p (x_j - \bar{x}_j)^2}} \right| \quad (4)$$

where x_i and x_j imply the vectors of features F_i and F_j in a respective manner. The variables \bar{x}_i and \bar{x}_j denote the mean values of vectors x_i and x_j , averaged over p samples. Obviously, the similarity value among a couple of completely similar features will be 1, and on the other hand, this value will be equal to 0 for entirely dissimilar features. Similar to fisher score values, all similarity values are normalized by the softmax scaling method.

It should be noted that in this step the feature selection problem was represented by a fully connected graph. Each edge in the graph was associated with a value which denoted the similarity value between every two nodes. Therefore, to reduce the time complexity and improve the maximum clique identification performance, before using the next step, the edges with associated weights lower than the θ parameter will be removed. The θ parameter can be set to any value in the range [0 1], and thus when its value is small (large), more (fewer) edges will be considered in the next steps.

After the generation of feature graphs, the initial nodes are divided into a number of clusters in such a way that the members of each cluster have the maximum similarity levels with respect to each other. Most of the existing feature clustering methods suffer from one or more of the following shortcomings [1]:

- the need to specify the number of clusters before performing feature clustering;
- the distribution of features in a cluster, which is one of the most important criteria in feature clustering, is not considered;
- all features are considered equally, while certain influential features should have a greater impact on the clustering process

To deal with these issues, community detection is used for feature clustering. The goal of community detection-based feature clustering is to group the most correlated features into the same community (group). In feature clustering, using community detection, the primary features are divided into a number of clusters, each “community” containing a number of features that are similar to each other. In fact, the features of each community are more similar and the features of different communities are less similar.

In this paper for feature clustering using community detection, an iterative search algorithm (ISCD) [79] is applied to cluster the features in this study. The ISCD algorithm can quickly detect communities, even in large graphs, due to the linear computational complexity. As such, it is efficient for feature clustering of high-dimensional data.

Step 3: Initialize Population:

A population set of chromosomes is produced in this step in a random manner. The number of original features n is equal to each chromosome length. Each chromosome gene is given a value of 1 or 0. When a feature is chosen, the respective gene in the chromosome is set to 1; otherwise, the gene value is set to 0. It is noteworthy that the total number of selected features in each chromosome must be $k \times \omega$, where k implies the number of clusters, and ω is a user-specified parameter controlling the size of the final feature subset.

Step 4: Calculate Fitness values:

After creating the initial population, the fitness function for all chromosomes must be calculated. For this purpose, in this proposed method, a novel multi-objective fitness function is introduced. In this fitness function, a combination of classification accuracy in the K-Nearest Neighbors (KNN) classification algorithm and the sum of similarities between the selected features is used. The fit of the FS^k feature subset in the iteration t denoted by $J(FS^k(t))$ is measured by Eq. (5).

$$J(FS^k(t)) = \frac{CA(FS^k(t))}{\frac{2}{|FS^k(t)| * (|FS^k(t)| - 1)} \sum_{F_i, F_j \in FS^k} Sim(F_i, F_j)} \quad (5)$$

where, $CA(FS^k(t))$ indicates the classification accuracy for the selected feature subset $FS^k(t)$ on the KNN classifier, $|FS^k(t)|$ represents the subset size the selected features $FS^k(t)$ and $Sim(F_i, F_j)$ indicates the similarity between the attribute F_i and F_j . As can be seen in this Equation, in calculating the suitability of each subset, the classification accuracy for that subset and the total similarity between the features selected in that subset are considered simultaneously. Consequently, a higher set of features is allocated to the feature's subset possessing the most relevance to the objective class and the least redundancy.

Step 5: Perform Crossover & Mutation operation:

New chromosomes are produced by crossover and mutation operators. The single point crossover among the selected chromosomes has been used in this research to produce new populations. In addition, a single parent chromosome may be flipped by randomly flipping one or more bits to create a child. That chromosome gene follows the predefined probability of mutation, whether or not it chooses to be mutated.

Step 6: Perform Repair Operation:

The proposed technique suggests a repair operation on an offspring among all freshly created chromosome to re-adjust the number of features selected from every group. If the number of selected features in one of the clusters is less than ω , one feature is randomly selected, and the corresponding feature is adjusted to be 1. Moreover, where more than one feature has been selected, one of them is randomly retained, and the other is eliminated from the chromosome. The repair process includes the unique and general characteristics of a certain dataset for the offspring generated by the fitter. Two steps are

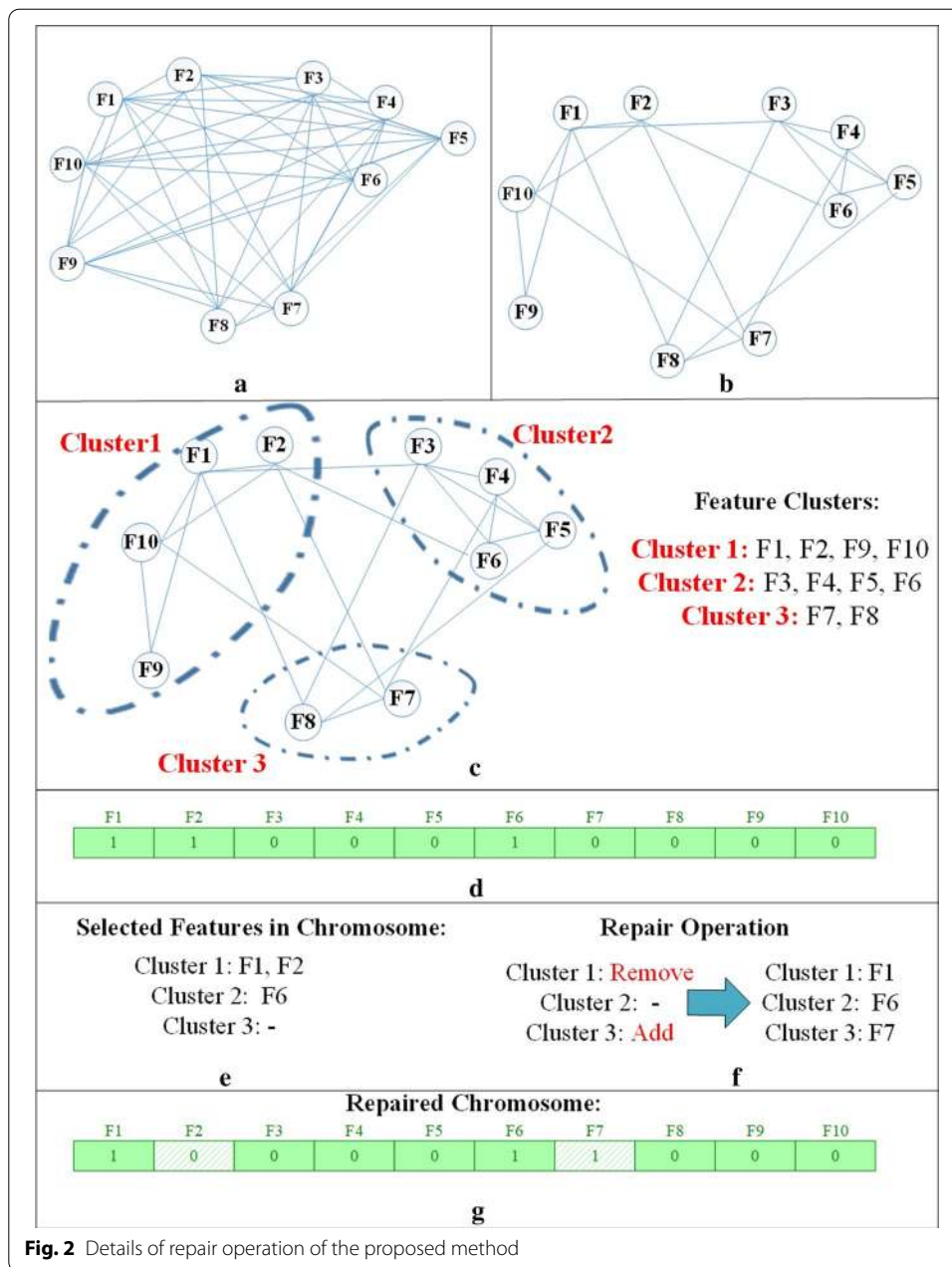


Fig. 2 Details of repair operation of the proposed method

regarded for the repair in CDGAFS: (i) check of the number of features in each cluster; and (ii) the enhancement of the offspring. It is noteworthy that only once will the first stage be done. The details of the repair procedure are shown in Fig. 2. This Figure illustrates the overall schema of the proposed repair operation for an empirical dataset with ten nodes. The complete graph for this dataset is shown in Fig. 2a. After edge removal, the complete graph is converted into a sparse graph. Figure 2b shows the graph from which edges with associated weights lower than the $\theta = 0.6$ parameter are removed. Then the community detection algorithm is applied and all ten features are divided into three clusters that are shown in Fig. 2c. These three stages (i.e. Fig. 2a–c) are performed only once in the proposed method and are considered as a pre-processing of the

genetic-based feature selection method. After these stages, the repair operation can be performed. Figure 2d shows the structure of a candidate chromosome for repair. As can be seen in this figure, in this candidate chromosome, three features have been selected from the initial features. As can be seen in Fig. 2e, from the Cluster 1, features of F1 and F2 are selected, from the Cluster 2, feature of F6 is selected and from the Cluster 3 no feature is selected. Given that the value of the parameter ω is equal to 1, therefore, one feature must be selected from each cluster. Since one of the selected features from the Cluster 1 must be randomly removed. Also, since no feature of Cluster 3 has been selected, one feature from Cluster 3 must be added to selected features in the chromosome, randomly. As shown in Fig. 2f, the feature of F2 is removed from the selected features from Cluster 1, and the feature of F7 is added to the selected features from cluster 3. Also, since, exactly one feature has been selected from Cluster 2, the selected features of this cluster do not change. Finally, the structure of the repaired chromosome can be seen in Fig. 2g.

In the description of the repair operator in the previous section, no explanation was given as to what features of each cluster should be added or removed. Consider the previous example; No attributes were selected from Cluster 2. As a result, all the features of this cluster have an equal chance of being selected. The question that arises here is which feature is better to select. There are two different strategies for selecting and removing features from a cluster.

Random Repair: In this strategy, when the number of features of a cluster is less than the required number of features that each cluster should have, from the unselected features of that cluster, so many features are randomly selected that the ω condition is satisfied (Select the number of ω features from each cluster).

Scoring Repair: The advantage of the first strategy was the speed of the repair operator. But in this strategy, when it was necessary to add or remove a feature from a cluster, no attention was paid to the suitability of the features and a feature was randomly selected. This may slow down the convergence of the genetic algorithm as well as reduce its performance. To solve this problem, in the scoring strategy, the repair operator is performed in such a way that the probability of selecting or removing the features in the repair process is determined based on the scoring assigned to them. For this purpose, the Fisher Score criterion, that defined in Step 1, is used to calculate the probability of adding or removing any feature in the repair process.

For example, if in the repair process in a particular case, three features F1, F2, and F3 are candidates to be added to the selected features in a cluster, and the normalized Fisher score for these three features is 0.6, 0.3, and 0.1 respectively, the feature of F1 is selected with a probability of 60%, feature of F2 with a probability of 30% and feature of F3 with a probability of 10%. In other words, using this strategy, the appropriateness of the features is also directly affected in the process of adding. Similarly, when removing a feature in the repair process, features with a higher score will be less likely to be removed. For example, suppose that in a particular case, three clusters F1, F2, and F3 are selected from a cluster with a normalized Fisher score of 0.4, 0.4, and 0.2, respectively, and It is necessary to remove a feature from them. In this case, the probability of removing each feature is calculated based on their inverse Fisher score. For the three features F1, F2, and F3, the inverse of the normalized Fisher score is 2.5, 2.5, and 5, respectively. After

this calculation, and according to these values, similar to the case of adding a feature, the probability of removing these three features is 25, 25, and 50 percent, respectively. In other words, with this strategy, features with a lower Fisher Score are more likely to be removed, and features with a higher score are less likely to be removed.

Step 7: Stopping Criterion:

In the case that the number of iterations is higher than the maximum allowable iteration, continue; otherwise, take a step in the fitness calculation.

Step 8: Final Subset Selection:

Eventually, according to its fitness value, the strongest chromosome of the last generation indicates the optimal subset of features for a specific dataset.

Algorithm 1: Community Detection-based Genetic Algorithm for Feature selection (CDGAFS)

Input D_f : Training dataset
 ω_i : Selected feature from each community

Output $F' = \{f'_1, \dots, f'_n\}$

1: **Begin algorithm**

2: Measure the feature relevance with Fisher Score

3: Calculate feature similarities

4: $\{C_1, C_2, \dots, C_k\}$ = Apply community detection algorithm to create feature clusters

5: Generate the initial population of chromosomes

6: **Do**

7: Calculate Fitness values

8: Perform Crossover & Mutation operation

9: Perform Repair Operation

10: **While until (The Stopping Criterion is met)**

11: Select the best chromosome

12: Report F' as final feature set

13: **End algorithm**

Algorithm 1 shows the pseudo-code of the proposed method.

Experimental results

Many tests were carried out for both the classification accuracy and the number of selected features to assess the proposed approach. The findings have been discussed in this section. The experiments were conducted on a 3.58 GHz CPU and 8 GB RAM machine.

In these experiments, one feature selection method was chosen and evaluated in the experimental result for comparing the efficiency of various techniques of feature selection based on each EA-based algorithm. For a fair evaluation, all of the methods examined in this section were selected from among wrapper-based methods. These wrapper-based methods include PSO-based [73], ACO-based [75], and ABC-based [78]. These are state-of-the-art EA-based feature selection methods.

PSO algorithm is an efficient swarm intelligence-based evolutionary algorithm, introduced by Kennedy and Eberhart in 1995 [80]. The PSO algorithm, inspired by the social behavior of birds and fish, has recently been utilized in many studies to solve the feature selection problem.

The ACO Algorithm was proposed by Dorigo et al. as a multi-agent to solve the optimization problems [81]. This algorithm is inspired by the behavior of ants that are able to find the shortest path between the nest and the food source and also adapt to environmental changes. Moreover, ACO has been successfully applied in several studies to feature selection.

Table 1 Characteristics of the used medical datasets

Dataset	Features	Classes	Patterns
SpamBase	57	2	4601
Sonar	60	2	208
Arrhythmia	279	16	351
Madelon	500	2	4400
Isolet	617	26	1559
Colon	2000	2	62

Table 2 User-specified parameters of the GA-based method

Parameter	Values
Crossover rate	0.8
Mutation rate	0.05
Population Size	100
Iteration number	100
ω (Selected feature in each community)	3
θ (Edge removing)	0.2

The ABC algorithm is an optimization algorithm based on swarm intelligence and intelligent behavior of the bee population that simulates the food search behavior of bee groups [82]. In the early version of this algorithm, it performs a kind of local search that is combined with a random search and can be used for hybrid optimization or functional optimization. This SI-based algorithm has been utilized in many studies to search for the optimal feature subset.

Datasets and preprocessing

The efficiency of CDGAFS was provided in this regard on six popular benchmark classification datasets, i.e., SpamBase, Sonar, Arrhythmia, Madelon, Isolet, and Colon. Several of these datasets include characteristics with missing values so that each missing value was substituted with the average of the data present on the corresponding feature to cope with these values in the tests. Furthermore, in many practical situations, a designer is faced with features; the values of these features are in various ranges. The features associated with a broad range of values thus dominate those related to small range values. A non-linear normalization approach named softmax scaling is applied to measure the datasets to solve this problem.

After the normalization process, each dataset was randomly partitioned into three subsets, such as validation set, training set, and testing set. The distribution of the number of instances and features of these datasets is presented in Table 1.

User-specified parameters

Similar to all feature selection methods, the proposed method has a number of parameters, such as population size, number of iterations, etc. These parameters are important for feature selection methods because they directly control the behaviors of the learning

Table 3 Average classification accuracy rate and as standard deviation (shown in parenthesis) over ten runs of the evolutionary-based feature selection methods using KNN, SVM, and AdaBoost classifier

Dataset	Method	Classifier		
		KNN	SVM	AdaBoost
SpamBase	PSO	<i>92.54 (2.83)</i>	<i>92.35 (1.52)</i>	<i>92.43 (2.25)</i>
	ACO	91.81 (1.82)	89.51 (2.81)	90.89 (2.21)
	ABC	90.35 (2.39)	89.22 (1.93)	91.30 (3.33)
	CDGAFS	<u><i>93.99 (2.76)</i></u>	<u><i>93.68 (1.73)</i></u>	<u><i>93.27 (2.82)</i></u>
Sonar	PSO	<i>88.18 (2.43)</i>	<i>87.81 (2.29)</i>	<i>86.93 (3.32)</i>
	ACO	88.06 (2.32)	87.36 (3.32)	85.82 (1.48)
	ABC	87.23 (1.13)	87.17 (2.81)	86.74 (1.78)
	CDGAFS	<u><i>88.71 (3.76)</i></u>	<u><i>88.34 (2.19)</i></u>	<u><i>87.13 (2.71)</i></u>
Arrhythmia	PSO	<i>86.15 (2.82)</i>	<i>86.01 (2.61)</i>	<i>85.91 (2.82)</i>
	ACO	84.13 (2.12)	86.27 (2.62)	85.72 (3.94)
	ABC	85.83 (2.73)	85.71 (1.75)	84.32 (1.39)
	CDGAFS	<u><i>87.21 (2.37)</i></u>	<u><i>87.38 (2.02)</i></u>	<u><i>86.98 (2.59)</i></u>
Madelon	PSO	86.46 (3.14)	86.65 (2.47)	86.12 (1.81)
	ACO	86.19 (2.20)	85.91 (1.32)	86.34 (2.11)
	ABC	<i>87.55 (2.13)</i>	<i>87.19 (1.81)</i>	<i>86.12 (2.33)</i>
	CDGAFS	<u><i>87.88 (1.55)</i></u>	<u><i>87.82 (1.64)</i></u>	<u><i>86.79 (1.62)</i></u>
Isolet	PSO	<i>85.63 (1.39)</i>	85.39 (1.62)	<i>85.42 (2.32)</i>
	ACO	85.26 (1.58)	<i>85.90 (1.81)</i>	85.41 (2.39)
	ABC	84.38 (2.81)	84.95 (2.16)	84.84 (1.48)
	CDGAFS	<u><i>86.11 (2.44)</i></u>	<u><i>86.01 (2.65)</i></u>	<u><i>85.39 (2.62)</i></u>
Colon	PSO	<u><i>96.41 (2.82)</i></u>	<u><i>96.19 (2.16)</i></u>	<u><i>96.32 (1.31)</i></u>
	ACO	94.43 (1.71)	95.73 (1.19)	95.32 (1.82)
	ABC	93.04 (2.56)	92.61 (3.61)	92.49 (3.45)
	CDGAFS	<i>95.41 (2.15)</i>	<i>95.82 (2.65)</i>	<i>95.36 (2.38)</i>

The best result is indicated in italics and underlined, and the second-best is in italics

model and have a considerable impact on the performance of final accuracy. To optimally choose these parameters, it is necessary to repeatedly set parameters and generate a number of predictions with different combinations of values, and then evaluate the prediction accuracy to select the best parameter values. As a result, choosing the best values for the parameters is an optimization problem. One way to optimize the selection of parameter values is to use an exhaustive search algorithm. Given that the accuracy of the learning model must be calculated to evaluate each combination of parameter values, this approach will not be applicable in situations where the construction of the learning model has high computational complexity.

In this paper, to implement different methods and adjust the parameters of each method, the parameter optimization method proposed in [83] is used for choosing the best values for their parameters. In this parameter optimization algorithm, the Bayesian theory-based optimization algorithm is used to solve the problem. Table 2 demonstrates the common parameters for all datasets.

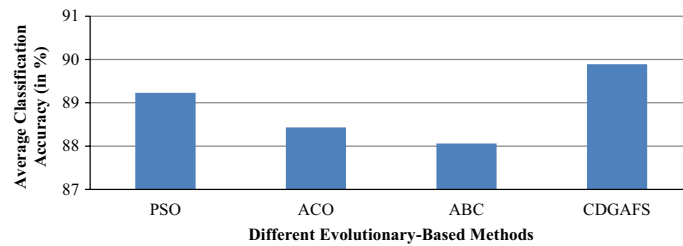


Fig. 3 Average classification accuracy over all datasets on the KNN classifier

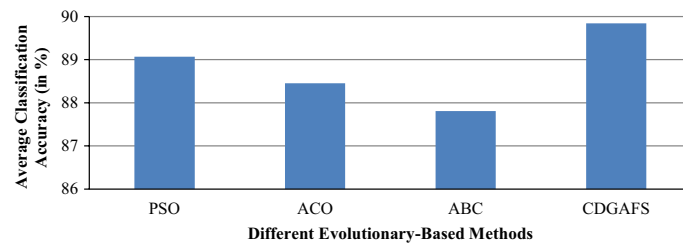


Fig. 4 Average classification accuracy over all datasets on the SVM classifier

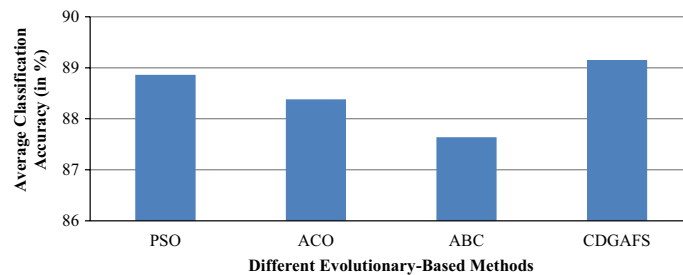


Fig. 5 Average classification accuracy over all datasets on the AdaBoost classifier

The utilized classifier

For assessing the generalizability of the presented approaches in various classifiers, in these tests, 3 classifiers, such as K-Nearest Neighbors (KNN), Support Vector Machine (SVM), AdaBoost (AB), and are utilized.

In pattern recognition, the KNN classifier is a non-parametric approach presented for regression and classification. In both cases, the input contains the nearest examples of training in the feature space. Support vector machine SVM is among Vapnik's supervised learning algorithms. The purpose of SVM is the maximization of the margin among data samples, and excellent performance for classification and regression problems has been shown recently. AdaBoost (AB) ("Adaptive Boosting") is a meta-algorithm for machine learning formulated by Yoav Freund and Robert Schapire. The AdaBoost classifier is a meta-estimator starting with the fitting of a classifier and fitting of additional copies on the identical dataset, afterward the weights of improperly grouped examples are modified to concentrate on severe cases more in subsequent classifiers. Weka (Waikato Environment for knowledge analysis) is the experimental workbench [84], a set of data

Table 4 Average number of selected features of the different wrapper evolutionary-based methods

Dataset	Number of the original feature	Method	Number of selected features	The ratio of the selected features to the original features (in %)
SpamBase	57	PSO	8.92	15.65
		ACO	8.87	15.56
		ABC	8.91	15.63
		CDGAFS	<u>8.17</u>	<u>14.33</u>
Sonar	60	PSO	7.31	12.18
		ACO	7.13	11.88
		ABC	7.83	13.05
		CDGAFS	<u>6.31</u>	<u>10.52</u>
Arrhythmia	279	PSO	20.12	7.21
		ACO	22.82	8.18
		ABC	20.93	7.50
		CDGAFS	<u>19.70</u>	<u>7.06</u>
Madelon	500	PSO	<u>74.35</u>	<u>14.87</u>
		ACO	69.91	13.98
		ABC	75.12	15.02
		CDGAFS	<u>75.03</u>	<u>15.01</u>
Isolet	617	PSO	141.63	22.95
		ACO	138.56	22.46
		ABC	171.73	27.83
		CDGAFS	<u>135.26</u>	<u>21.92</u>
Colon	2000	PSO	<u>12.73</u>	<u>0.64</u>
		ACO	15.71	0.79
		ABC	13.22	0.66
		CDGAFS	<u>12.98</u>	<u>0.65</u>

Minimum number of selected features is indicated in italics and underlined and the second best is in italics

mining methods. In the present study, KNN, AdaBoostM1, and Sequential Minimal Optimization (SMO) as the WEKA implementation of KNN, AB, and SVM have been applied.

Results

In these experiments, the feature subset size and classification accuracy are used as the performance evaluation criteria. In the experiments, first, the comparison of the performances of different wrapper SI-based feature selection approaches is done with various classifiers. Table 3 presents the mean classification accuracy (%) over 10 independent runs of the various SI-based wrapper feature selection techniques by employing KNN, SVM, and AB classifiers. Each entry of Table 3 implies the mean value and standard deviation (given in parenthesis) of 10 independent runs. The optimal result is demonstrated in an underlined and italics, and the second-best is in italics. Table 3 shows that, in the majority of cases, the performance of the proposed CDGAFS approach is better compared to the other evolutionary-based feature selection method. For instance, in the SpamBase dataset on the KNN classifier, the proposed method obtained a 93.99%

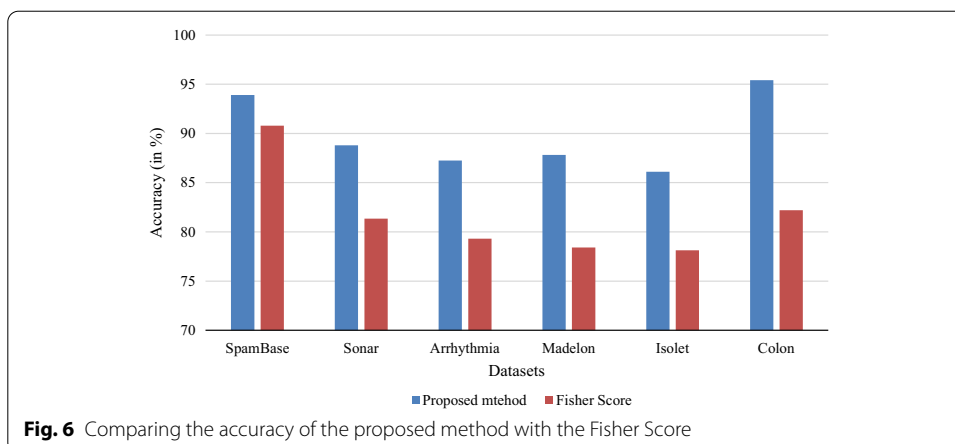


Table 5 Average execution time (in second) of wrapper feature selection methods over ten independent runs

Dataset	PSO	ACO	ABC	CDGAFS
SpamBase	6.72	8.42	8.94	6.64
Sonar	4.24	7.78	8.32	4.12
Arrhythmia	22.83	27.31	30.18	21.61
Madelon	89.58	98.12	108.62	88.72
Isolet	48.92	52.34	58.92	47.19
Colon	60.82	79.12	61.41	54.95
Average	38.85	45.51	46.06	37.20

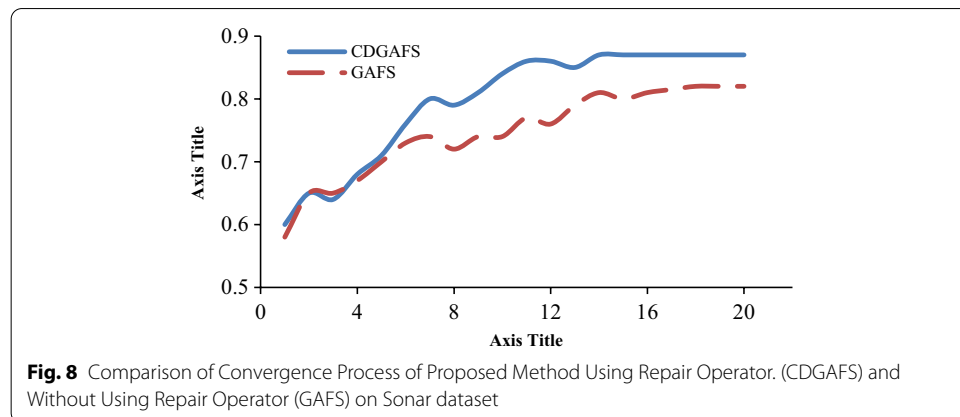
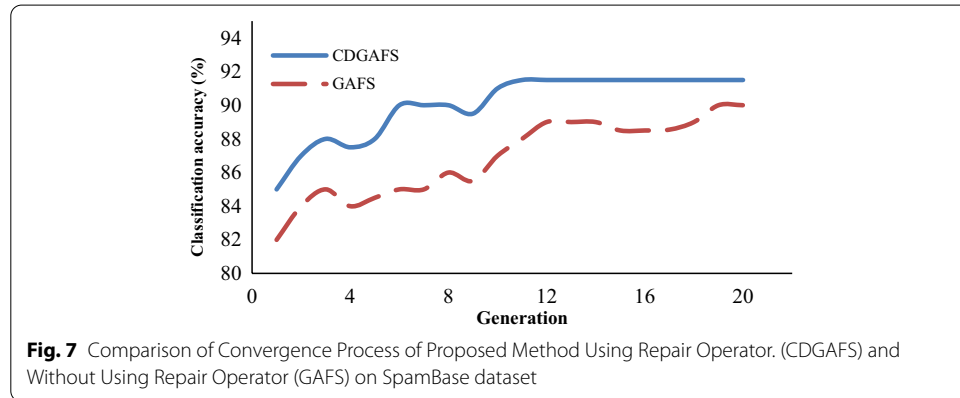
classification accuracy. In contrast, for PSO-based [73], ACO-based [75], and ABC-based [78] methods, these values were reported 92.54%, 91.81%, and 90.35%, correspondingly.

Moreover, Figs. 3, 4, 5 show the mean classification accuracy over all datasets on the KNN, SVM, and AdaBoost classifiers, respectively. As can be seen in these figures, on all classifiers, the suggested approach had the highest average classification accuracy. The findings presented in Fig. 3 indicate that the presented technique obtained 89.89% mean classification accuracy and obtained the first rank with a 0.66% margin in comparison with the PSO-based approach, which achieved the second-best average classification accuracy. Moreover, the results presented in Fig. 4 show the discrepancies among the achieved classification accuracy of the suggested technique, and the second-best ones (PSO-based) and third-best ones (ACO-based) on SVM classifier were reported 0.77 (i.e., 89.84–89.07) and 1.39 (89.84–88.45) percent. Furthermore, based on the result of Fig. 5, on the AB classifier, the proposed CDGAFS method gained the first rank with an average classification accuracy of 89.15%, and the ACO-based and PSO-based feature selection techniques were ranked second and third with an average classification accuracy of 88.86% and 88.38%, respectively.

Table 4 records the number of selected features of the four wrappers evolutionary-based feature selection approaches for each dataset. It is evident that in a general manner, all the four approaches obtain a considerable decrease of dimensionality by choosing a small part of the original features. Among various methods, in SpamBase,

Table 6 Average ranks of the different EA-based feature selection methods on SVM, NB, and AB classifier

Dataset	Wrapper-based feature selection method			
	PSO	ACO	ABC	CDGAFS
KNN	2	3.33	3.5	1.17
SVM	2.33	2.83	3.67	1.17
AdaBoost	1.83	2.83	3.5	1.67



Sonar, Arrhythmia, Isolet datasets, the proposed technique shows the best performance compared to the other evolutionary-based approaches, selecting only 14.33, 10.52, 7.06, and 21.92%, respectively. Moreover, in the Madelon and Colon datasets, the PSO-base method selected an average of 14.87 and 0.64% of features, respectively. In Madelon and Colon datasets, the proposed feature selection method was ranked second with a mean classification accuracy of 15.01% and 0.65%, respectively.

As described in “Proposed method” section, in the Repair Operator step, the suitability of the features is calculated based on the Fisher score criterion for adding or removing a feature. In fact, in the proposed method, it is necessary to calculate the importance of each attribute based on the Fisher score criterion before starting the search strategy of

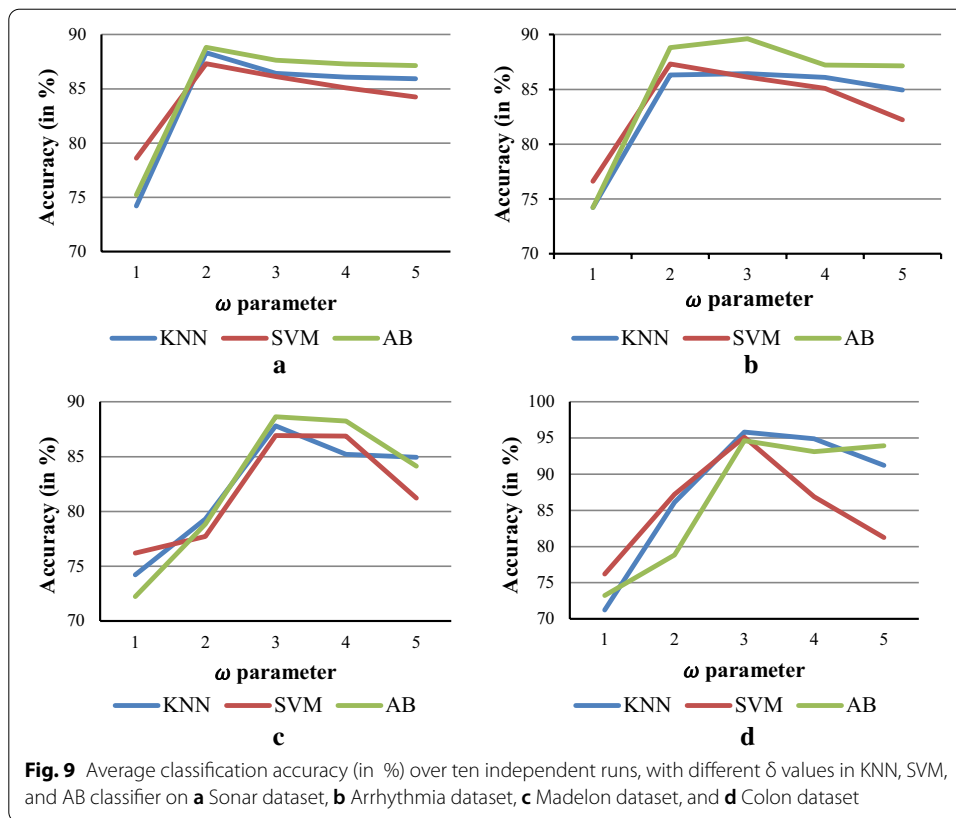
the genetic algorithm. Figure 6 compares the performance of the proposed method with the standard Fisher score feature selection method. In fact, in this Figure, the increase in the accuracy of the proposed method compared to the Fisher method is investigated. As the results of Fig. 6 shows, in all datasets, the accuracy of the proposed method is much higher than the Fisher score method. For example, the accuracy of the proposed method in the Sonar dataset is 3.11% and in the Colon dataset is 13.22% higher than the Fisher Score method. Also, the results of this experiment show that in datasets with higher dimensions, the margin accuracy between the proposed method and Fisher score has increased. The reason for this is that in these datasets with higher dimensions, it is more important to consider the relationships between features, and the Fisher score method will not be able to select an optimal subset because it does not consider the relationships between features.

Also, several experiments were conducted to compare the execution time of different wrapper EA-based feature selection methods. In these experiments, corresponding execution times (in second) for each method, were reported in Table 5. Due to the fact that the feature selection process and the final classification process are independent, only the execution time for feature selection is reported in the data in this Table. The reported results revealed that the proposed CDGAFS feature selection method has the lowest average execution time overall dataset among all other methods. After the proposed method, PSO-based and ACO-based methods ranked second and third, respectively.

The performance of CDGAFS for feature selection can be observed in Tables 3, 4, 5; however, the influence of repair operation upon the feature selection process is unclear. Several tests have been conducted to explain exactly how the repair process plays a significant role in CDGAFS for feature selection tasks. Figures 7 and 8 indicate the classification accuracy of GA-based feature selection algorithms in Sonar and SpamBase datasets as well as demonstrate that CDGAFS has been able to find salient features in feature space easily and rapidly. The successful function of CDGAFS repair can be observed clearly in these figures. In these figures, CDGAFS and GAFS denote the GA-based feature selection with proposed repair operation and GA-based feature selection without repair operation, respectively.

Sensitivity analysis of the parameters

The proposed feature selection method has two parameters of θ and ω , where their corresponding optimal values should be specified by the programmer. The θ parameter is a threshold that is applied to the weighted graph of original features to remove the edges with values less than θ . After this action, the size of the initial graph is reduced considerably. The parameter ω is a that controls the number of selected features from each community. In fact, this parameter is used to control redundancy and its corresponding value is very important to determine the number of selected features and accuracy of the classifier. This parameter can be set to any value in the range $[1M]$, where M is the minimum number of features in the communities. On one hand, if this parameter is tuned to a number close to M , the final future subset size will be too large and similar features may be chosen. On the other hand, when ω is adjusted to a number close to 1, a small set of features is selected. Therefore, these selected



features cannot fully represent the initial features and the microarray data classification accuracy will be reduced.

These parameters are critical to the developed feature selection method because they straightly affect the accuracy of the prediction algorithm, and therefore the final accuracy of the classification depends to a large extent on the precise selection of these parameters. To fine-tune these parameters, you need to adjust the parameters repeatedly and create a number of predictions with a different integration of values, and then measure the classification performance to choose the optimal values. Since optimal adjusting of these parameters can be considered as an optimization problem. One strategy for optimal adjusting is to employ an exhaustive search strategy. This method will not be practical in cases where building a prediction algorithm has a high execution time.

To search for the appropriate value for the ω parameter, different experiments were designed to denote how the classification accuracy changes with different values of that parameter. Figure 9a–d reveals the ω parameter sensitivity analysis for Sonar, Arrhythmia, Madelon, and Colon datasets, correspondingly. The experiment evaluates the classification performance on the KNN, SVM, and AB classifiers for different ω values. The results shown that in all datasets when the ω is adjusted to 2 or 3, the CDGAFS method achieves the best classification accuracy.

Moreover, the effect of the θ parameter on the classification accuracy and the search for its optimal value on different datasets has been investigated in Fig. 10. Similar to

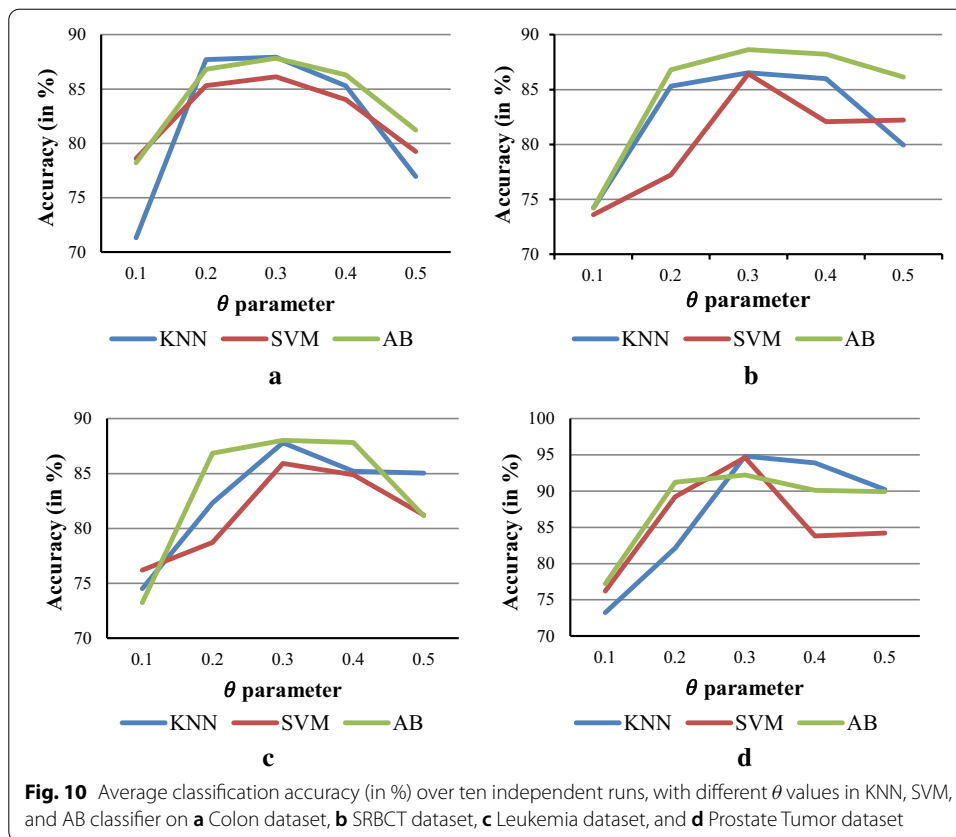


Table 7 The results of the Friedman statistics test

	Classifier		
	KNN	SVM	AB
Chi Square	13.400	11.800	8.491
df	3	3	3
Asymp.Sig (p-value)	0.003847	0.008101	0.036874

the ω Sensitivity analysis, in Fig. 10a–d the θ parameter sensitivity analysis for Sonar, Arrhythmia, Madelon, and Colon datasets are shown, respectively. In these experiments, the value of the θ parameter was changed from 0.1 to 0.6. The results reveal that in all cases when the parameter θ is adjusted to 0.3, the developed feature selection method achieves the best performance.

Complexity analysis

In this subsection, the computational complexity of the proposed method is calculated. In the first step of the proposed method, the fisher score of all features is measured. The computational complexity of the fisher score calculation is $O(ncp)$, where n is the

number of the original features and p denotes the number of patterns and c is the number of classes in the dataset.

The first step of the method aims at converting the feature space into a graph and requires $O(n^2p)$ time steps where n is the number of the original features and p denotes the number of patterns. Moreover, in the next phase, a community detection algorithm is applied to find the feature clusters. The complexity of the community detection algorithm is $O(n \log n)$. Then a specific genetic algorithm-based search technique is utilized to choose the final feature set. The search algorithm will be repeated for a number of iterative cycles (i.e., I). Thus, the time complexity of this part is $O(IPkf_k)$, where P is the number of the chromosomes in the population, k is the number of the clusters and f_k denotes the time complexity to calculate the fitness function. The time complexity of the KNN classifier is $O(Pn)$. Therefore, the computational complexity of this phase is equal to $O(IP^2nk)$. Consequently, the final computational complexity of the proposed method is $O(n^2p + n \log n + IP^2nk)$, which are reduced to $O(n^2p + p^2n)$.

Statistical analysis

In this subsection, the Friedman test [85] is applied to the statistical analysis of the reported results. The Friedman test is a nonparametric test utilized to compare the performance of different feature selection on various datasets. For this purpose, each feature selection method is ranked on each dataset. To this end, the SPSS statistics acquired by IBM is used. In the Statistical test results, it is not possible to say that if the level of significance is less than the level of error, the difference between at least a pair of specimens is deducted. Since the test errors are considered at 5%, the level of significance must be lower than 0.05 to satisfy this constraint. Table 6 present the average calculated ranking for different wrapper-based feature selection methods on each classifier. The results of Table 6 show that the CDGAFS method has the best average ranking. Table 7 shows that the Friedman test has reported a p-value of 0.003847, 0.008101, and 0.036874 in the wrapper-based methods on KNN, SVM, and AB classifiers, respectively. Since these values are below 0.05, it can be claimed that the results of the proposed CDGAFS method are significantly different from those of other wrapper-based methods.

Discussion

The main reasons that lead to the effectiveness of the proposed method are explained, as follows.

- Unlike the other clustering-based feature selection methods such as k-means and fuzzy c-means, the proposed community detection feature selection method identifies the number of clusters automatically, and there is no longer a need to determine the number of clusters in advance. The proposed method uses a community detection-based repair operation which considers both the local and global structure of the graph in computing similarity values.
- The proposed method clustered similar features into the groups and then utilized a multi-objective fitness function to assign an importance value to each feature subset. In the proposed multi-objective fitness function, two objectives of

feature relevance and feature redundancy are considered, simultaneously. Unlike the other multi-objective methods that identify a set of non-dominated solutions in an iterative process, the proposed method finds the near-optimal solution in a reasonable time.

- The main goal of gene selection is to avoid keeping too many or too few genes. If too few genes are chosen, there will not be enough information for the microarray data classification task. In contrast, if too many genes are selected, the gene space of the dataset will be blurred by irrelevant and redundant features. In the proposed method, unlike many previous works, the optimal number of selected features is determined automatically based on the overall structure of the original features and their inner similarities.

Conclusion

Feature selection contributes significantly to machine learning and particularly classification tasks. The computational cost is minimized and the model is designed from simplified data that enhance the overall capabilities of classifiers. A framework was proposed which integrates the advantages of filter and wrapper methods and embeds such a framework into the genetic algorithm in the present article. Some excellent aspects of the proposed technique enhance the efficiencies, the summarization of which is presented as the following. Initially, feature similarities and feature relevance are calculated. Second, CGAFS applies community detection to eliminate redundant features. Hence, the proposed approach picks a certain number of features from each cluster. Also, in this method, unlike previous methods, a multi-objective evolutionary algorithm for the feature selection problem is proposed. The comparison of the performance of the suggested technique with the other feature selection methods is done. The reported results indicate that the proposed method gives higher efficiency, faster convergence, and search efficiency compared to other feature selection methods.

There are several user-specified parameters used in the developed feature selection methods and thus their corresponding values should be determined by the user. These parameters are important for feature selection methods because they directly control the behaviors of the learning model and have a considerable impact on the performance of the final prediction. To optimally choose these parameters, it is necessary to repeatedly set parameters and generate number of predictions with different combinations of values, and then evaluate the prediction accuracy to select the best parameter values. As a result, choosing the best values for the parameters is an optimization problem. One way to optimize the adjustment of parameter values is to use an exhaustive search algorithm. Given that the accuracy of the learning model must be calculated to evaluate each combination of parameter values, this approach will not be applicable in situations where the construction of the learning model has high computational complexity. It is suggested that in future work, a parameter optimization method can be used to adjust the parameters. Moreover, for future work, the authors intend to investigate various community detection and social network analysis techniques and apply the maximum clique algorithm for automatically determining the number of clusters and feature clustering.

Abbreviations

EA: Evolutionary algorithm; ACO: Ant colony optimization; GA: Genetic algorithm; SA: Simulated annealing; TS: Taboo search; PSO: Particle Swarm Optimization; ABC: Artificial Bee Colony; SSB: Subset selection-based; KNN: K-Nearest neighbors; SVM: Support Vector Machines; SI: Swarm intelligence; CDGAFS: Community detection-based genetic algorithm for feature selection; ISCD: Iterative search algorithm for community detection; SMO: Sequential minimal optimization.

Acknowledgements

None.

Authors' contributions

The specific contributions made by each author is as follows: MR: Conceptualization, methodology, implementation, writing—original draft, writing—review & Editing. KB: Methodology, validation, writing—review & editing. SF: Methodology, validation, formal analysis. All authors read and approved the final manuscript.

Funding

None.

Availability of data and materials

Datasets are available in UCI repository of machine learning datasets, Available from: <http://archive.ics.uci.edu/ml/datasets.html>. Source code is available in: <https://github.com/mehrdad1988/FeatureSelection>.

Competing interests

The authors declare that they have no competing interests.

Author details

¹ Department of Computer Engineering, University of Kurdistan, Sanandaj, Iran. ² Department of Science and Engineering, Queensland University of Technology, Brisbane, Australia. ³ Department of Computer Engineering, University of Applied Science and Technology, Center of Tehran Municipality, ICT org, Tehran, Iran.

Received: 18 September 2020 Accepted: 11 December 2020

Published online: 04 January 2021

References

- Moradi P, Rostami M. A graph theoretic approach for unsupervised feature selection. *Eng Appl Artif Intell*. 2015;44:33–45.
- Robbins KR, Zhang W, Bertrand JK. The ant colony algorithm for feature selection in high-dimension gene expression data for disease classification. *J Math Med Biol*. 2008;24(4):413–26.
- Adebijoyi M, et al. Computational investigation of consistency and performance of the biochemical network of the malaria parasite, *Plasmodium falciparum*. *Computational science and its applications—ICCSA 2019*. Cham: Springer; 2019.
- Arowolo MO, Adebijoyi M, Adebijoyi A, Okesola O. PCA model for RNA-Seq malaria vector data classification using KNN and decision tree algorithm. In: 2020 international conference in mathematics, computer engineering and computer science (ICMCECS). 2020. p. 1–8.
- Forouzandeh S, Berahmand K, Rostami M. Presentation of a recommender system with ensemble learning and graph embedding: a case on MovieLens. *Multimed Tools Appl*. 2020. <https://doi.org/10.1007/s11042-020-09949-5>.
- Jain AK, Duijn RP, Mao J. Statistical pattern recognition: a review. *Pattern Anal Mach Intell IEEE Trans*. 2000;22(1):4–37.
- Olaolu AM, Abdulsalam SO, Mope IR, Kazeem GA. A comparative analysis of feature selection and feature extraction models for classifying microarray dataset. *Comput Inf Syst J*. 2018;29.
- Arowolo MO, Isiaka RM, Abdulsalam SO, Saheed YK, Gbolagade KA. A comparative analysis of feature extraction methods for classifying colon cancer microarray data. *EAI Endorsed Trans Scalable Inf Syst*. 2017;4(14):153147.
- Renuka Devi D, Sasikala S. Online Feature Selection (OFS) with Accelerated Bat Algorithm (ABA) and Ensemble Incremental Deep Multiple Layer Perceptron (EIDMLP) for big data streams. *J Big Data*. 2019;6(1):103.
- Tadist K, et al. Feature selection methods and genomic big data: a systematic review. *J f Big Data*. 2019;6(1):79.
- Rejer I, Twardochleb M. Gamers' involvement detection from EEG data with cGAAM—a method for feature selection for clustering. *Expert Syst Appl*. 2018;101:196–204.
- Cheng-Lung H, Tsai CY. A hybrid SOFM-SVR with a filter-based feature selection for stock market forecasting. *Expert Syst Appl*. 2009;36(2):1529–39.
- Tubishat M, et al. Improved Salp Swarm Algorithm based on opposition based learning and novel local search algorithm for feature selection. *Expert Syst Appl*. 2020;145:113122.
- Yazdi KM, Yazdi AM, Khodayi S, Hou J, Zhou W, Saedy S, Rostami M. Improving recommender systems accuracy in social networks using popularity. In: 2019 20th international conference on parallel and distributed computing, applications and technologies (PDCAT). IEEE. 2019. p. 301–7.
- Majbourni Yazdi K, et al. Prediction optimization of diffusion paths in social networks using integration of ant colony and densest subgraph algorithms. *J High Speed Netw*. 2020;26:141–53.
- Berahmand, K., et al. A new Attributed Graph Clustering by using Label Propagation in Complex Networks. *Journal of King Saud University-Computer and Information Sciences*, 2020.
- Berahmand K, Bouyer A. LP-LPA: a link influence-based label propagation algorithm for discovering community structures in networks. *Int J Mod Phys B*. 2018;32(06):1850062.

18. Berahmand K, Bouyer A. A link-based similarity for improving community detection based on label propagation algorithm. *J Syst Sci Complexity*. 2019;32(3):737–58.
19. Berahmand K, Bouyer A, Vasighi M. Community detection in complex networks by detecting and expanding core nodes through extended local similarity of nodes. *IEEE Trans Comput Soc Syst*. 2018;5(4):1021–33.
20. Liu Y, et al. Flexible unsupervised feature extraction for image classification. *Neural Networks*. 2019;115:65–71.
21. Rostami M, M.P., *A clustering based genetic algorithm for feature selection*. *Information and Knowledge Technology (IKT)*, 2014: 112–116.
22. Arowolo MO, et al. A hybrid heuristic dimensionality reduction methods for classifying malaria vector gene expression data. *IEEE Access*. 2020;8:182422–30.
23. Ghosh M, Sanyal G. An ensemble approach to stabilize the features for multi-domain sentiment analysis using supervised machine learning. *J Big Data*. 2018;5(1):44.
24. Chen R-C, et al. Selecting critical features for data classification based on machine learning methods. *J Big Data*. 2020;7(1):52.
25. Welikala RA, et al. Genetic algorithm based feature selection combined with dual classification for the automated detection of proliferative diabetic retinopathy. *Comput Med Imaging Graph*. 2015;43:64–77.
26. Singh U, Singh SN. A new optimal feature selection scheme for classification of power quality disturbances based on ant colony framework. *Appl Soft Comput*. 2019;74:216–25.
27. Alshamlan HM, Badr GH, Alohalı YA. Genetic Bee Colony (GBC) algorithm: a new gene selection method for microarray cancer classification. *Comput Biol Chem*. 2015;56:49–60.
28. Moradi P, Rostami M. Integration of graph clustering with ant colony optimization for feature selection. *Knowl Based Syst*. 2015;84:144–61.
29. Hosseini FS, et al. Flash-flood hazard assessment using ensembles and Bayesian-based machine learning models: application of the simulated annealing feature selection method. *Sci Total Environ*. 2020;711:135161.
30. Oduntan IO, et al. A multilevel tabu search algorithm for the feature selection problem in biomedical data. *Comput Math Appl*. 2008;55(5):1019–33.
31. Rostami M, et al. Integration of multi-objective PSO based feature selection and node centrality for medical datasets. *Genomics*. 2020;112(6):4370–84.
32. Unler A, Murat A, Chinnam RB. mr2PSO: a maximum relevance minimum redundancy feature selection method based on swarm intelligence for support vector machine classification. *Inf Sci*. 2011;181(20):4625–41.
33. Wenzhu Y, Daoliang L, Zhu L. An improved genetic algorithm for optimal feature subset selection from multi-character feature set. *Expert Syst Appl*. 2011;38:2733–40.
34. Anusha M, Sathiaselalan JGR. Feature selection using K-Means genetic algorithm for multi-objective optimization. *Proc Comput Sci*. 2015;57:1074–80.
35. Marcelloni F. Feature selection based on a modified fuzzy C-means algorithm with supervision. *Inf Sci*. 2003;151:201–26.
36. González J, et al. A new multi-objective wrapper method for feature selection—accuracy and stability analysis for BCI. *Neurocomputing*. 2019;333:407–18.
37. Xue B, Zhang M, Browne WN. Particle swarm optimization for feature selection in classification: a multi-objective approach. *Cybernetics, IEEE Trans*. 2013;43(6):1656–71.
38. Tuba E, et al. Classification and feature selection method for medical datasets by brain storm optimization algorithm and support vector machine. *Proc Comput Sci*. 2019;162:307–15.
39. Yan K, et al. Cost-sensitive and sequential feature selection for chiller fault detection and diagnosis. *Int J Refrig*. 2018;86:401–9.
40. Li S, et al. Dual graph regularized compact feature representation for unsupervised feature selection. *Neurocomputing*. 2019;331:77–96.
41. Jayaraman V, Sultana HP, Artificial gravitational cuckoo search algorithm along with particle bee optimized associative memory neural network for feature selection in heart disease classification. *J Ambient Intell Hum Comput*, 2019.
42. Zhang Y, et al. Binary differential evolution with self-learning for multi-objective feature selection. *Inf Sci*. 2020;507:67–85.
43. Emary E, Zawbaa HM, Hassanien AE. Binary grey wolf optimization approaches for feature selection. *Neurocomputing*. 2016;172:371–81.
44. Neggaz N, et al. Boosting salp swarm algorithm by sine cosine algorithm and disrupt operator for feature selection. *Expert Syst Appl*. 2020;145:113103.
45. Rostami M, Berahmand K, Forouzandeh S. A novel method of constrained feature selection by the measurement of pairwise constraints uncertainty. *J Big Data*. 2020;7(1):83.
46. Arowolo MO, et al. A hybrid dimensionality reduction model for classification of microarray dataset. *Int J Inf Technol Comput Sci*. 2017;9(11):57–63.
47. Tabakhi S, Moradi P. Relevance–redundancy feature selection based on ant colony optimization. *Pattern Recogn*. 2015;48(9):2798–811.
48. Tabakhi S, Moradi P, Akhlaghian F. An unsupervised feature selection algorithm based on ant colony optimization. *Eng Appl Artif Intell*. 2014;32:112–23.
49. Barak S, Dahooie JH, Tichy T. Wrapper ANFIS-ICA method to do stock market timing and feature selection on the basis of Japanese Candlestick. *Expert Syst Appl*. 2015;42(23):9221–35.
50. Agor J, Özalın OY. Feature selection for classification models via bilevel optimization. *Comput Oper Res*. 2019;106:156–68.
51. Gao W, et al. Feature selection considering the composition of feature relevancy. *Pattern Recogn Lett*. 2018;112:70–4.
52. Ferreira AJ, Figueiredo MA. An unsupervised approach to feature discretization and selection. *Pattern Recogn*. 2012;45(9):3048–60.

53. Battiti R. Using mutual information for selecting features in supervised neural net learning. *Neural Netw IEEE Trans*. 1994;5(4):537–50.
54. Estévez PA, et al. Normalized mutual information feature selection. *Neural Netw IEEE Trans*. 2009;20(2):189–201.
55. Kwak N, Choi C-H. *Input feature selection for classification problems*. *Neural Networks, IEEE Transactions on*. 2002;13(1):143–59.
56. Hoque N, Bhattacharyya DK, Kalita JK. MIFS-ND: a mutual information-based feature selection method. *Expert Syst Appl*. 2014;41(14):6371–85.
57. Bennisar M, Hicks Y, Setchi R. Feature selection using joint mutual information maximisation. *Expert Syst Appl*. 2015;42(22):8520–32.
58. Labani M, et al. A novel multivariate filter based feature selection method for text classification problems. *Eng Appl Artif Intell*. 2018;70:25–37.
59. Pashaei E, Pashaei E, Aydin N. Gene selection using hybrid binary black hole algorithm and modified binary particle swarm optimization. *Genomics*. 2019;111(4):669–86.
60. Nematzadeh H, et al. Frequency based feature selection method using whale algorithm. *Genomics*. 2019;111(6):1946–55.
61. Tawhid MA, Dsouza KB. Hybrid Binary Bat Enhanced Particle Swarm Optimization Algorithm for solving feature selection problems. *Appl Comput Informatics*. 2018;1(2):181.
62. Prasad Y, Biswas KK, Hanmandlu M. A recursive PSO scheme for gene selection in microarray data. *Appl Soft Comput*. 2018;71:213–25.
63. Zhang S, et al. Swarm intelligence applied in green logistics: a literature review. *Eng Appl Artif Intell*. 2015;37:154–69.
64. Wang C, Pan H, Su Y. A many-objective evolutionary algorithm with diversity-first based environmental selection. *Swarm Evol Comput*. 2020;53:100641.
65. Hu Y, et al. A dynamic multi-objective evolutionary algorithm based on intensity of environmental change. *Inf Sci*. 2020;523:49–62.
66. Gong D, et al. A similarity-based cooperative co-evolutionary algorithm for dynamic interval multiobjective optimization problems. *IEEE Trans Evol Comput*. 2020;24(1):142–56.
67. Yong Z, Dun-wei G, Wan-qiu Z. Feature selection of unreliable data using an improved multi-objective PSO algorithm. *Neurocomputing*. 2016;171:1281–90.
68. Maleki N, Zeinali Y, Niaki STA. A k-NN method for lung cancer prognosis with the use of a genetic algorithm for feature selection. *Expert Syst Appl*. 2021;164:113981.
69. Amini F, Hu G. A two-layer feature selection method using genetic algorithm and elastic net. *Expert Syst Appl*. 2021;166:114072.
70. Rathee S, Ratnoo S. Feature selection using multi-objective CHC genetic algorithm. *Proc Comput Sci*. 2020;167:1656–64.
71. Sayed S, et al. A Nested Genetic Algorithm for feature selection in high-dimensional cancer Microarray datasets. *Expert Syst Appl*. 2019;121:233–43.
72. Yan C, et al. A novel hybrid feature selection strategy in quantitative analysis of laser-induced breakdown spectroscopy. *Anal Chim Acta*. 2019;1080:35–42.
73. Xue Y, et al. Self-adaptive parameter and strategy based particle swarm optimization for large-scale feature selection problems with multiple classifiers. *Appl Soft Comput*. 2020;88:106031.
74. Dadaneh BZ, Markid HY, Zakerolhosseini A. Unsupervised probabilistic feature selection using ant colony optimization. *Expert Syst Appl*. 2016;53:27–42.
75. Liu Y, et al. A classification method based on feature selection for imbalanced data. *IEEE Access*. 2019;7:81794–807.
76. Arslan S, Ozturk C. Multi Hive Artificial Bee Colony Programming for high dimensional symbolic regression with feature selection. *Appl Soft Computing*. 2019;78:515–27.
77. Zhang Y, et al. Cost-sensitive feature selection using two-archive multi-objective artificial bee colony algorithm. *Expert Syst Appl*. 2019;137:46–58.
78. Wang X-H, et al. Multi-objective feature selection based on artificial bee colony: an acceleration approach with variable sample size. *Appl Soft Comput*. 2020;88:106041.
79. Bai L, et al. Fast graph clustering with a new description model for community detection. *Inf Sci*. 2017;388–389:37–47.
80. Kennedy J, Eberhart R. Particle swarm optimization. In: *The Proceedings of the 1995 IEEE International Conference on Neural Network*, 1995: 1942–1948.
81. Dorigo M, Caro GD. Ant colony optimization: a new meta-heuristic. In: *Proceeding of the Congress on Evolutionary Computing*, 1999.
82. Karaboga D. An idea based on honey bee swarm for numerical optimization, Technical Report-TR06. Kayseri: Erciyes University, Engineering Faculty, ComputerEngineering Department; 2005.
83. Wu J, et al. Hyperparameter optimization for machine learning models based on Bayesian Optimization. *J Electr Sci Technol*. 2019;17(1):26–40.
84. Hall M, et al. The WEKA data mining software. <http://www.cs.waikato.ac.nz/ml/weka>.
85. Friedman M. A comparison of alternative tests of significance for the problem of m rankings. *Ann Math Stat*. 1940;11(1):86–92.

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.