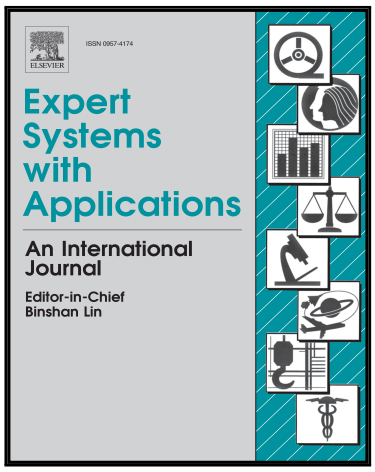


Hybrid Fast Unsupervised Feature Selection for High-dimensional Data

Zhaleh Manbari , Fardin Akhlaghian Tab , Chiman Salavati

PII: S0957-4174(19)30016-8
DOI: https://doi.org/10.1016/j.eswa.2019.01.016
Reference: ESWA 12417

To appear in: *Expert Systems With Applications*

Received date: 8 March 2018
Revised date: 28 December 2018
Accepted date: 4 January 2019

Please cite this article as: Zhaleh Manbari , Fardin Akhlaghian Tab , Chiman Salavati , Hybrid Fast Unsupervised Feature Selection for High-dimensional Data, *Expert Systems With Applications* (2019), doi: https://doi.org/10.1016/j.eswa.2019.01.016

Highlights

- Propose a new hybrid feature selection algorithm based on BACO and clustering.
- Modify linear binary ant system to reduce the search space complexity.
- Inject mutation to increase randomness of search space.
- Feature clustering to decrease the challenges of processing high-dimensional dataset.
- Experiment the method in several real-world social datasets and obtain more efficiency.

Hybrid Fast Unsupervised Feature Selection for High-dimensional Data

Zhaleh Manbari [a], Fardin Akhlaghian Tab [a,*], Chiman Salavati[a]

[a] Department of Computer Engineering, University of Kurdistan, Sanandaj, Iran

* Corresponding author. Tel.: +98 8733660066.

E-mail addresses: Zh.manbari@eng.uok.ac.ir (Zhaleh Manbari), f.akhlaghian@uok.ac.ir (Fardin Akhlaghian Tab), chiman.salavati @eng.uok.ac.ir (Chiman Salavati)

Abstract- The genesis of "curse of dimensionality" issue as a result of high dimensional datasets deteriorates the capability of the learning algorithms and also requires high memory and computational costs. Feature selection by discarding redundant and irrelevant features plays as a crucial machine learning technique aimed at reducing the dimensionality of these datasets, which improves the performance of the learning algorithm. Feature selection has been significantly applied in many application areas relevant to expert and intelligent systems, such as data mining and machine learning. Although many algorithms have been developed so far, they are still unsatisfied confronting high dimensional data. This paper presents a new hybrid filter-based feature selection algorithm based on the combination of clustering and modified Binary Ant System (BAS), called FSCBAS to overcome the search space and high-dimensional data processing challenges efficiently. This model provides both global and local search capabilities between and within clusters, respectively. In the proposed method, inspired by genetic algorithm and simulated annealing, a damped mutation strategy is introduced that avoids falling into local optima, and a new reduction redundancy policy to estimate the correlation between selected features which have further improved this algorithm. The proposed method can be applied in many expert system applications such as microarray data processing, text classification and image processing in high dimensional to handle the high dimensionality of the feature space and improve classification performance simultaneously. The performance of the proposed algorithm is compared to that of state-of-the-art feature selection algorithms using different classifiers on real-world datasets. The experimental results confirmed that the proposed method reduces computational complexity significantly and achieves better performance than other feature selection methods.

Keywords: Feature selection, High-dimensional data, Binary Ant System, Clustering, Mutation

1 Introduction

In big data era, high-dimensional data in different fields such as social media, bioinformatics, image processing, and natural language processing has been extended (L. Hu, Gao, Zhao, Zhang, & Wang, 2018). Computational burden, over-fitting, and poor performance are some of the disadvantages bringing by appearance of high dimensional data (Lin, Liu, & Liu, 2015). In pattern recognition, selecting a subset of informative features from high-dimensional data continues to be a challenge due to the presence of thousands of features (Gangeh, Zarkoob, & Ghodsi, 2017). Filtering out irrelevant and redundant features can reduce overfitting, save computational cost and improve the accuracy of classification. Therefore, efficient and highly desired the feature selection algorithms are those which can reduce the original data into a low dimensional space effectively. Feature selection is a combinatorial optimization problem and is an important issue in a wide range of scientific disciplines, nowadays.

Feature selection techniques have been successfully applied to various expert system fields such as Text Mining(Ghareb, Bakar, & Hamdan, 2016), Image processing /computer vision(Lin, Chen, & Wu, 2014; Muštra, Grgić, & Delač, 2012), Bioinformatics (Dessì, Pascariello, & Pes, 2013) (Ghosh, Begum, Sarkar, Chakraborty, & Maulik, 2018; Latkowski & Osowski, 2015; Ma & Tavares, 2017; Mohamed, Zainudin, & Othman, 2017), Industrial applications (Pacheco et al., 2017; Yuyan Zhang, Li, Gao, & Li, 2018; Zhou, Pang, Lewis, & Zhong, 2011), and other fields (Amiri, Yousefi, Lucas, Shakery, & Yazdani, 2011; Balabin & Smirnov, 2011; Z. Hu, Bao, Xiong, & Chiong, 2015; Huan Liu & Yu, 2005). In text mining, the bag-of-words model is the standard way of representing a document, in which each document is modeled with the counts of words occurring in it and each feature depicts the count of a specific word. The performance and efficiency of subfields of text mining including text clustering and text classification are improved by using the feature selection. In image processing, due to the enormous number of image features in practical, representing images is not a simple task and the selection of features depend on the target application. Examples of features include edges, corners, the histograms of oriented gradients, raw pixels, gradient values, color channels, etc. (Brkić, 2013). Moreover, feature selection is used in object detection, image classification, or image clustering. One of the applications of feature selection in image classification is breast density classification in mammographic images (Muštra et al., 2012). Feature selection improves the accuracy of detecting a fault in industrial applications, where numerous redundant sensors monitor the performance of a machine. Another interesting application of feature selection is in biomarker discovery from genomics data. In genomics data, individual features correspond to genes, when the most related features are selected, an important knowledge about the genes is achieved that is the most discriminative for a particular problem. In bioinformatics, feature selection methods are applied to select the genes of unknown function, reveal natural structure inherent in gene expression data, microarray dimension reduction and discover implicit links between the genes.

Although many algorithms have been developed, none of these are suitable for all situations, and researchers are still looking for better solutions. In general, the feature selection algorithms can be grouped in different categorization methods which three of them are expressed as follows.

1.1 Categorize the methods of selecting features based on the accessibility of class label data.

According to the accessibility of class label data, feature selection can be categorized as supervised, unsupervised and semi-supervised feature selection (Cai, Luo, Wang, & Yang, 2018; Huan Liu & Yu,

2005). In supervised feature selection, class labels are accessible (Jenatton, Audibert, & Bach, 2011; S. Kim & Xing, 2012; Y. Kim & Kim, 2004; H. Peng, Long, & Ding, 2005; Xiang, Nie, Meng, Pan, & Zhang, 2012; J.-B. Yang & Ong, 2011; Z. Zhao, Wang, Liu, & Ye, 2013), while in unsupervised feature selection, the lack of class labels is a primary challenge. In recent years, a variety of unsupervised methods have been proposed (He, Cai, & Niyogi, 2005b; Y. Jiang & Ren, 2011; Padungweang, Lursinsap, & Sunat, 2012; Y. Yang, Shen, Ma, Huang, & Zhou, 2011; Z. Zhao & Liu, 2007). Semi-supervised feature selection algorithms utilize both labeled and unlabeled data (Xu, King, Lyu, & Jin, 2010; Zeng, Wang, Zhang, & Wu, 2016; J. Zhao, Lu, & He, 2008).

1.2 Categorize the methods of selecting features based on their relationship with learning model

Feature selection algorithms based on different evaluations are divided into four main categories: the filter, wrapper, embedded, and hybrid approaches (Gheyas & Smith, 2010; Huan Liu & Yu, 2005; Saeys, Inza, & Larrañaga, 2007; Sotoca & Pla, 2010; Tabakhi, Moradi, & Akhlaghian, 2014). The filter approach is based on statistical information of features and does not require any learning algorithms for evaluation. Therefore, filter-based feature selection methods are computationally very efficient and fast. The wrapper approach uses a specific learning algorithm and often improves learning performance at the cost of increasing computational complexity. The feature subset selected in this approach partially depends on the type of learning algorithm. (Huang & Huang, 2009; Wan, Wang, Ye, & Lai, 2016; Wei et al., 2017). Embedded methods embed feature selection into the learning algorithm. In this approach, a specified learning model is trained by an initial feature, and also the optimal feature subset is generated based on the constructed classifier. Therefore, there is a deeper interaction between the feature selection process and the learning model. The main weakness of this class of feature selection algorithms is its long-time processing, particularly on high-dimensionality datasets, to train a learner with the full feature set. The hybrid approach utilizes the benefits of both the filter and wrapper approaches to providing a balance between the execution time and the performance of learning algorithms (Das, Goswami, Chakrabarti, & Chakraborty, 2017). Approaches based on Filter techniques are worth mentioning. Filter-based feature selection methods reduce correlation among features and enhance the correlation between the features with the class label by using the evaluation criteria. The filter methods can be divided into two categories, as follows:

1.2.1 Relevance and redundancy based filter model.

Relevance and redundancy analysis is the basis of methods in this category. MRMR (Max-Relevance and Min-Redundancy) is a classical criterion for feature selection based on relevance and redundancy analysis. MRMR (Max-Relevance and Min-Redundancy) is a classical criterion for feature selection based on relevance and redundancy analysis (Che et al., 2017; Herman, Zhang, Wang, Ye, & Chen, 2013; H. Peng et al., 2005; Tabakhi & Moradi, 2015). There are feature selection algorithms that offered various mechanisms to reduce redundancy or irrelevance or both in the feature selecting processing. When a feature is considered as redundant, this means that the inferred information by this feature is such as one other feature. Also, in a redundant subset feature to prevent any negative impact to learn model accuracy, only a small number of features can be selected as a part of the final feature subset. On the other hand, when a feature is marked as potentially irrelevant, it means that this feature plays an insignificant role, and give no information in classifying or clustering a set of data instances. Since with the increasing number of features in the

high-dimensional data, the correlation between features become more complicated, consequently, the redundancy among features should be investigated more precisely in feature selection process. Although some of the feature selection methods consider this issue partly; nonetheless, the literature of works that considered the correlation between features and remove redundant features are insufficient and this should be considered more accurate.

Lack of a comparative method to investigate the redundancy of a feature is one of the major challenges of these methods. Although these methods consider correlation between features, the subset of feature in order to investigate the redundancy of a feature is not specified properly. Another challenge is that the computational complexity of these methods is often high; therefore, these methods are not suitable for high dimensional data**.**

1.2.2 Relevance and diversity based filter model

Feature selection methods are studied with emphasis on diversity among features by some of the researchers (Novovicová, Pudil, & Kittler, 1996; Yishi Zhang, Li, Wang, & Zhang, 2013). Two optimization problems including max Relevance and max Diversity (MRMD) can express this type of model (Ienco & Meo, 2008; Huawen Liu, Wu, & Zhang, 2011) (Dhillon, Mallela, & Kumar, 2003). As it is shown in Fig. 1 the basic process in these methods comprise of three steps: First, selecting the suitable measure of the distance, to form feature space. Second, categorizing features by clustering method; third, selecting the representative features of each cluster, to acquire the desirable feature subset.

Cluster-based feature selection methods can be applied on high-dimensional data to take benefit of its relatively low complexity. Furthermore, the performance of the feature selection process increase by using the information obtained from clusters (Chuang, Tsai, & Yang, 2011; García-Torres, Gómez-Vela, Melián-Batista, & Moreno-Vega, 2016; Jang et al., 2011; Moradi & Rostami, 2015; Song, Ni, & Wang, 2013). Many of these works are too sensitive to the clustering method, and there are some challenges in selecting features from each cluster.

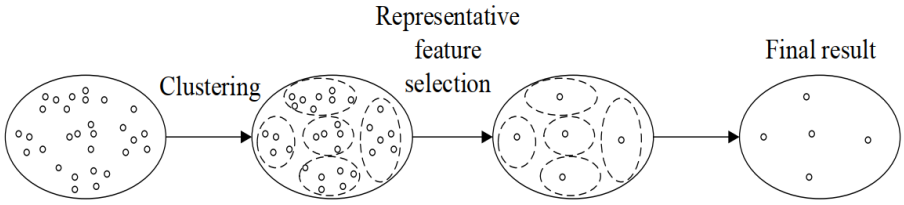


Fig. 1. The process of clustering-based feature selection (Cai et al., 2018)

1.3 Categorize the methods of selecting features based on the search strategy

Many feature selection algorithms are based on a search strategy for the selection of candidate feature subsets. According to the search strategy, feature selection methods are divided into three categories: *complete, randomized* and *heuristic search*. *Complete search* includes searching the best subset of features in the whole search space with size 2^n, where n is the number of features. As a result, detecting the best subset of features is practically impossible in a high-dimensionality dataset within a reasonable time. *Randomized search* methods explore a limited space of the total state

space, and the size of the subspace depends on the stopping criterion, such as the maximum number of iterations. Although a tradeoff is made between the speed of convergence and the optimality of the result using parameter setting, algorithm still has a tendency to become trapped in a local optimum. Clearly, the computational complexity of these algorithms is less than that of algorithms based on complete search. In feature selection algorithms based on *heuristic search*, one feature is added to or removed from the selected feature set in each iteration. Therefore, their computational complexity is much less than that of algorithms using complete search (Huan Liu & Yu, 2005). So far, many algorithms based on heuristic search have been designed. In recent years, in particular, more attention has been paid to swarm intelligence based approaches, such as genetic algorithm (GA) (Oh, Lee, & Moon, 2004), ant colony optimization (ACO) (Wan et al., 2016), particle swarm optimization (PSO) (Chuang, Tsai, et al., 2011; Chuang, Yang, & Li, 2011), and forest optimization algorithm (Ghaemi & Feizi-Derakhshi, 2016). The global search ability of the metaheuristic search methods, especially in high dimensional data, is very useful (Amoozegar & Minaei-Bidgoli, 2018).

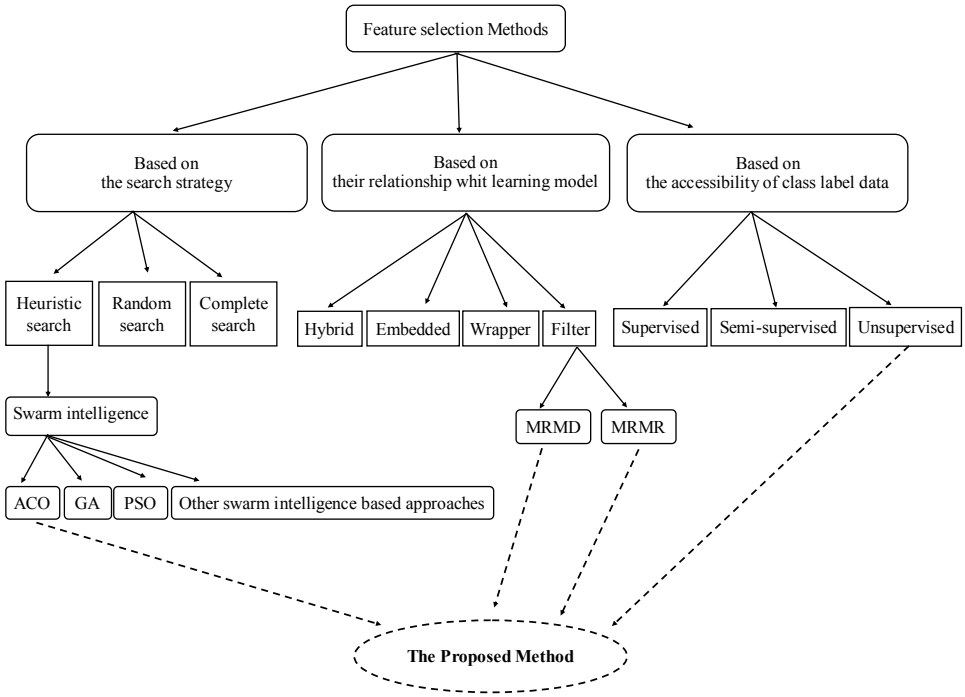


Fig. 2. The path of genesis of the proposed method

1.4 Motivate and contributions

The problems identified by the authors which motivate the current work are (Fig. 2):

- **Since in the worst case the class labels of data are not available, it is sharply essential to present a new *unsupervised* feature selection method.**
- **Filter method is simpler and faster, especially for high dimensional data; therefore, providing *a filter based method* is highly desired.**

- **With the increasing number of features in the high-dimensional data, the correlation between features become more complicated. Consequently, the *redundancy* among features should be investigated more precisely.**
- **The cluster-based feature selection methods are so *sensitive to the clustering method*, and there are some challenges in selecting features from each cluster.**
- **Swarm intelligence based feature selection approaches are unable to achieve an optimal subset in proper time due to *getting trapped into local optimum* especially in big data.**

In order to solving the mention challenges, we combine the advantages of ACO based method and the clustering based methods, and we presented a new *hybrid filter-based feature selection* algorithm which is a combination of *linear binary ant system*, *clustering*, *mutation injection* and a *new reduction redundancy policy*. In this paper, a new systematic way is presented to do a quick and efficient search in the state space for feature selection. A new feature selection algorithm is presented based on clustering and binary ant system (BAS) method. The proposed method improves learning methods by decreasing redundancy as far as possible and achieves an optimal solution by increasing search space in a short time. For this purpose, in the first step features are clustered. In each cluster, the features are organized sequentially in a circular graph and the proposed feature selection based on BAS, called BASM, is applied to them. In the BASM, the representation of the search space reduces computational time significantly, especially on the high-dimensional dataset. In addition, a new revision of computing of features correlation is utilized to consider the redundancy between selected features more efficiently, and furthermore, a mutation strategy is introduced to avoid falling into local optima. In the following, define the competition between the best features of all clusters so that the features are sorted based on pheromone values and in an iterative forward procedure BASM is applied between achieved top features from clusters. The main advantage of our approach is high performance, at least as great as that in full graph ACO based methods, and exhibits much lower complexity which significantly reduces runtime.

In summary, the main contributions of this paper are highlighted as follows:

- **An appropriate search space representation is introduced to decline the computational complexity.**
- **A reasonable technique (damped mutation) is applied to overcome the challenge of getting trapped into local optimum.**
- **An excellent policy is considered to study redundancy among features.**
- **By feature clustering, the challenge of processing high-dimensional datasets is reduced to some extent.**
- **Using both global and local search capabilities inter and intra the clusters is provided.**
- **The proposed method can be implemented in *semi-parallel processes* and increase the efficiency, significantly.**
- **Performance and computational complexity are improved simultaneously.**
- **Having high performance in high dimensional data and low computational complexity which caused to be applied in the search-based platform to solve other optimization problems.**
- **The presented method is practical in processing big data.**

- **By dividing and conquering the problem involving feature clustering, applying BACO on clusters, and defining the competition between the best features of all clusters, the proposed method has been able to drastically reduce complexity meanwhile improving the results.**

For clarifying our contribution, strengths from the methodological and industrial viewpoints are highlighted in Table 1.

Table 1. Contribution, methodological and industrial viewpoint of the proposed method

<table>
<tr><th rowspan="2">Effective in</th><th rowspan="2">Contribution</th><th colspan="2">Strengths from the methodological and industrial viewpoints</th></tr>
<tr><th>methodological</th><th>industrial</th></tr>
<tr><td>Computational complexity

Classification Performance

Sensitive of parameters/ methods</td><td>(1) Combination of linear binary ant system, clustering, damped mutation, and a new reduction redundancy policy
(2) Applying reasonable techniques to overcome the challenge of getting trapped into local optima
(3) Considering an excellent policy to study redundancy among features
(4) Definition a comparative space among best features to enhance the performance
(5) Combining advantages of clustering and ACO
(6) Divide and conquer the problem to decline computational complexity</td><td>(1) Effective in high dimensional data
(2) Practical in processing the big data
(3) Implementation in semi-parallel
(4) Having low computational complexity
(5) Solving the challenges of clustering-based feature selection methods such as:
• Selecting the representative features of each cluster to generate the subset of selected features
• Sensitive to the clustering method</td><td>(1) Big image processing
(2) Microarray processing
(3) Microarray gene expression data classification
(4) Applicable in the search-based platform to solve other optimization problems such as Identifying influential nodes in complex networks</td></tr>
</table>

The rest of this article is organized as follows. Some related works are introduced in Section 2. In Section 3, the background of ant colony optimization and the binary ant system is reviewed briefly. In Section 4, the proposed feature selection method, called FSCBAS, is detailed. Section 5 provides the experimental results and discussion. Finally, in Section 6 the paper is concluded.

2 Related Works

In the previous section, feature selection algorithms were categorized based on different criteria and each category was thoroughly described. Table 2 shows a summary of this assortment and the challenges of each group is determined.

Given the path to the proposed method and the special concentrate of this article, this section is more focused on the relevant literature on ACO based feature selection algorithms and clustering-based feature selection algorithms so that the innovations of the proposed method can be compared to this literature. In following the relevant literature in four categories is discussed.

Table 2. A summary of four categories of feature selection methods compared with the proposed method

Methods	Summary of Challenges

Complete search- based feature selection methods	Impractical on high-dimensionality data
MRMR-based feature selection methods	High computational complexity Unsuitable comparative model
Cluster-based feature selection methods	Sensitive to parameters or methods
Metaheuristics-based feature selection methods	Getting trapped into local optima
The proposed method	**A novel hybrid method to overcome the above-mentioned challenges of the four categories**

The first category of feature selection algorithms is complete search-based feature selection methods. The aims of these methods are to examine all the states of the problem space and select the optimal mode. Since there is 2^n modes, the complete search is impossible in practical. The reason for classifying this category is to evaluate the proposed method and it compares to the best method in terms of efficiency. There are some complete search methods and their extensions in the literature (Narendra & Fukunaga, 1977; Y. Peng, Li, & Liu, 2006; Somol, Pudil, & Kittler, 2004; B. Yu & Yuan, 1993), Gaining an optimal feature subset is an NP-hard problem. Hence, suboptimal algorithms are typically used to find acceptable solutions.

The second category is MRMR-based feature selection methods. Relevance and redundancy analysis is the basis of methods belong to this category. These models use Euclidean distance, Pearson correlation, and information measures for relevance and redundancy analysis (Huan Liu & Motoda, 2012; Huan Liu & Yu, 2005). The features in an original set can be divided into four groups: (a) completely irrelevant and noisy features, (b) weakly relevant and redundant features, (c) weakly relevant and non-redundant features, and (d) strongly relevant features (L. Yu & Liu, 2004). Peng et al. (H. Peng et al., 2005) proposed mRMR that implements this criterion by a first-order incremental search. However, mutual information based MRMR only minimizes feature mutual information and ignores the classification performance of candidate features, which might be influenced by the selected features. Conditional mutual information analysis is then introduced to overcome this problem (Herman et al., 2013; Yishi Zhang & Zhang, 2012). Recntly, Che et al (Che et al., 2017) proposed N-MRMCR-MI seek to maximize relevance and minimize redundancy under an optimization problem for nonlinear data. The approximate expression and incremental heuristic search approaches are usually employed by these methods because directly calculating mutual information between feature subset and the class label is difficult. Tabakhi and Moradi (Tabakhi & Moradi, 2015) presented an ACO-based algorithm by considering relevance and redundancy of features. In order to investigate redundancy in this work, a feature is compared to all the features selected in the previous section. However, the subset of feature aimed to investigate the redundancy of a feature is not specified appropriately. This challenge is more precisely described in the proposed method and some strategies are presented to overcome this issue.

The third category of feature selection algorithm is cluster-based feature selection methods. Clustering can be considered as the most important issue in unsupervised learning. The feature clustering is the task of clustering a set of features in such a way that features in the same cluster are more similar to each other than to those in other clusters.

In general, feature clustering or feature grouping by improving the stability of feature selection and reducing the complexity of the model, is a useful technique in learning when there are high-dimensional data (Jörnsten & Yu, 2003; Shen & Huang, 2010). On the other hand, feature grouping has arisen as a powerful tool to reduce dimensionality in high-dimensional data, and it also increases the effectiveness of the search. Furthermore, the information obtained from clusters can be used to increase performance in the feature selection process (Chuang, Tsai, et al., 2011; Jang et al., 2011; Huan Liu & Yu, 2005; Moradi & Rostami, 2015; J. Zhao et al., 2008). There are several works in which feature grouping is performed using learning algorithms such as K-means (Wang, Chu, & Xie, 2007), self-organizing map (Silva & Marques, 2010), and logistic regression (Dettling & Bühlmann, 2004). Moreover, some approaches apply kernel density estimation (L. Yu, Ding, & Loscalzo, 2008), information-theory measures (Krier, François, Rossi, & Verleysen, 2007), graph theory (Song et al., 2013), and regularization techniques (Tibshirani, Saunders, Rosset, Zhu, & Knight, 2005).

In literature (Yishi Zhang et al., 2013) Kullback-Leibler divergence is used to build a feature selection framework which focuses on separate class. Dhillon (Dhillon et al., 2003) proposed a global criterion for feature/word clustering, and presented a fast, divisive method for text classification. This method maximizes the Jensen-Shannon divergence between feature clusters. Ienco and Meo (Ienco & Meo, 2008) hierarchically clustered the feature sets according to their correlations and selected the best feature from each cluster to form the final feature subset by using a packing method. This method does not require to adjust any parameters, however, increases the time complexity and also append the learning method bias. Witten et al (Witten & Tibshirani, 2010) presented a feature clustering framework in which sparse K-means and hierarchical clustering are integrated. At first, multiple feature clusters are formed as a result of clustering the feature set. And in the second step, the representative features from each cluster are selected by using the Lasso-type penalty factors. Zhao (X. Zhao, Deng, & Shi, 2013) applied the maximum information coefficient as the measure of feature correlation, carried out the affinity propagation clustering of the feature subset, and then selected the “centroid” from each cluster as the representative feature of the cluster. Since the clustering process might fall into a dead loop, the number of iterations should be predefined to stop the clustering process.

As a result, many of these works are too sensitive to the clustering method or setting its parameters, and there are some challenges in selecting features from each cluster, such as selecting the cluster centers as the final feature set in more of them.

In high-dimensional data, relevant features are highly correlated; therefore, clustering correlated features that are resistant to the variations of the sample size is possible, and it is the definite advantage of the clustering- based methods. Fast clustering-based feature selection algorithm (FAST) (Song et al., 2013) is based on grouping correlated features. In FAST, first features are clustered by utilized graph-theoretic clustering methods. Then, in order to obtain a subset of features, a feature which is strongly related to target classes is selected from each cluster. Feature clusters also provide additional informative group structure for expert domains to further investigate (García-Torres et al., 2016). In (L. Yu et al., 2008) Dense Feature Groups (DFG) based on a popular non-parametric method referred to as Kernel Density Estimation (KDE) is presented (L. Yu et al., 2008). KDE estimates probabilistic density functions for estimating the density of the features. In (Loscalzo, Yu, & Ding, 2009) the authors presented Consensus Group Stable Feature Selection called CGS which approximates intrinsic feature groups by a set of consensus feature groups aggregated from multiple

sets from ensemble learning so that it uses the idea of DFG to generate groups on each sample. Recently, another group-based feature selection framework is proposed in which approximate algorithms are applied that provide good solutions in a reasonable time (García-Torres et al., 2016). In this model, the concept of Markov blankets is used for grouping the input space into subsets of features, and then the predominant groups are used to design a Variable Neighborhood Search (VNS) metaheuristic to handle high-dimensional datasets.

In this paper, another cluster-based feature selection algorithm is proposed in which the simple K-means is applied for features clustering to indicate the point that the presented method is not sensitive to the clustering method.

The fourth category of feature selection algorithm in the categorization of authors is Swarm Intelligence -based feature selection methods. Swarm Intelligence includes a population of artificial agents that try to simulate the behavior of a group of animals within the natural world. Each agent performs a simple task individually, but they solve a complex problem together. Genetic algorithm (GA), Ant colony optimization (ACO), Particle swarm optimization (PSO), and Artificial bee colony (ABC) are the most popular swarm-based algorithms (Amoozegar & Minaei-Bidgoli, 2018; S. Jiang, Chin, Wang, Qu, & Tsui, 2017; Zorarpacı & Özel, 2016). Feature selection algorithms exert metaheuristic approaches such as GA (Ghareb et al., 2016), ACO(Tabakhi & Moradi, 2015), PSO (Amoozegar & Minaei-Bidgoli, 2018; Tran, Xue, & Zhang, 2018), Taboo (Shi, Wan, Gao, & Wang, 2018; H. Zhang & Sun, 2002), and Scatter (López, Torres, Batista, Pérez, & Moreno-Vega, 2006)as a fundamental tool, which each of them has some specific strengths.

Ant colony optimization algorithm is more flexible than other algorithms in the scope of feature selection. It is implemented easily and enables global and local search. Huang et al. (Huang & Huang, 2009) proposed a wrapper method based on ACO that utilized the SVM classifier. Chen (Y. Chen, Miao, & Wang, 2010) proposed a rough set approach based on ant colony optimization for feature selection, along with a feature selection algorithm based on ACO and the power set tree for image classification and recognition (Y. Chen, Miao, Wang, & Wu, 2011). Youchuan et al. (Wan et al., 2016) presented a wrapper method based on the modified binary coded ant colony optimization algorithm, which combines GA and ACO. Since the high computational complexity of these wrapper based approaches, filter based methods are more prevalent such as an unsupervised feature selection approach based on ACO (UFACO) (Tabakhi et al., 2014), relevance–redundancy feature selection using ACO(Tabakhi & Moradi, 2015), and gene selection for microarray data classification based on a novel ACO (Tabakhi, Najafi, Ranjbar, & Moradi, 2015). Most feature selection methods based on ACO have high computational complexity with $O(n^2)$ edges; where the number of features is represented by n, due to the use of a fully connected graph (Kashef & Nezamabadi-pour, 2015; Tabakhi & Moradi, 2015; Tabakhi et al., 2014; Tabakhi et al., 2015). This means that $O(n^2)$ heuristic information and pheromone should be computed and stored.

Kong and Tian (Kong & Tian, 2006) introduced a binary ant colony optimization (BACO) in which the features are presented as nodes and they stand in a sequential order. There are two arcs between every two nodes corresponding select or deselect of the node. By utilizing this policy, they have reduced graph edge complexity to $O(2n)$. Jang (Jang et al., 2011) proposed a novel BACO algorithm based on ACO and developed it to solve the unit commitment problem of power systems.

Chen (B. Chen, Chen, & Chen, 2013) presented a feature selection algorithm based on ant colony optimization for image clarification. In this work, the ants traverse the directed graph with $O(2n)$ edges. A new hybrid method was proposed by Kadri (Kadri, Mouss, & Mouss, 2012) utilized BACO which aimed at fault diagnosis of the rotary kiln, which was based on SVM and BACO to obtain an optimization subset by eliminating the noise feature. Kadir (Kadri et al., 2012) also used fully connected graph which represented binary. Consequently, this has led to high computational complexity. The main shortcoming of BACO based methods is its limited search space; thus, the algorithm gets stuck in local optima easily which caused lower efficiency compared to full graph ACO based methods. Most of the BACO based methods employ the wrapper approach to improve efficiency which in turn increases computational complexity, in contrast, the presented algorithm in this paper is a filter approach which overcomes these defects.

3 Background

This section provides a background of the two core concepts of this article: ant colony optimization and the binary ant system (BAS) (Kong & Tian, 2005, 2006).

3.1 Ant colony optimization

In computer science and operations research, the ant colony optimization (ACO) algorithm is one of the swarm intelligence methods which is based on a probabilistic technique for solving computational problems. The ant colony algorithm that was initially proposed by Marco Dorigo in 1992 (Dorigo & Gambardella, 1997), constitutes some metaheuristic optimization. An optimal path in a graph is obtained based on the behavior of ants searching throughout their colony for food sources and it is the main objective of the algorithm. In the natural world, ants search arbitrarily until they find food and then return to their colony while pouring some pheromone on their trails. The next ants traverse a path with maximum pheromone. Since the evaporation of the pheromone trail, more pheromones evaporate on long paths. Thus, on shorter paths, the pheromone density becomes higher than other paths. If there were no pheromone evaporations, the ants would tend to be attracted to the path chosen by the first ant. Under such conditions, exploration of the solution space is constrained and leads to convergence to a locally optimal solution. In view of the mentioned description, which is solving an optimization problem by utilizing the idea of the ant colony behavior, the search space is represented by a graph and “simulated ants” traverse it to find a solution.

Many combinatorial optimization problems can be solved by applying ant colony optimization method. ACO has simple implementation and lower execution time than most other swarm intelligence methods; such as genetic algorithm and simulated annealing in optimization problems including feature selection and classification domains. Moreover, when the graph changes dynamically (e.g. in network routing and urban transportation systems), ACO can be applied continuously and adapted to the environment.

3.2 The Binary Ant System

Kong and Tiam (Kong & Tian, 2005, 2006), introduced a Binary ant system (BAS) applied to constrained optimization problems with binary solution structure. This means that the search space is represented as a binary graph in which every node is connected to next node by two edges (Fig. 3).

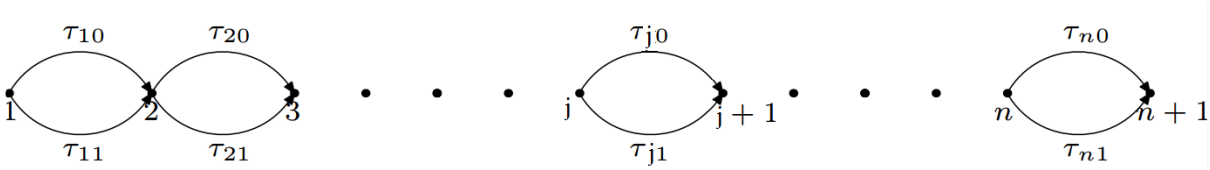


Fig. 3. Routing Diagram of Ants in BAS(Kong & Tian, 2006)

Ants construct their solutions by traversing the graph from beginning toward the last node. The representation of solution as a binary string: $x = \{x_1, \ldots, x_n\}$ where $x_j \in \{0,1\}$. At each iteration, a group of ants starts to search in the binary search domain and a solution can be constructed by ants walking sequentially from node 1 to node $n + 1$ on the graph. At every node j, the ant selects one of two paths to walk to the next node $j + 1$. The pheromone value for node x_j is represented by τ_{js} to $s \in \{0,1\}$ corresponding to selected/deselected states. An initial pheromone is distributed on all nodes for select/deselect state represented as τ_0. As the solutions are constructed, ant k selects node j based on the probability distribution stated below:

$$p_{js}^{k}(t) = \frac{\tau_{js}(t)}{\sum_{l \in \{0,1\}} \tau_{jl}(t)}, \ s \in \{0,1\} \tag{1}$$

For generation of solutions, all ants make a selection for every node. Once every ant has completed its tour, the objective function is used for evaluation and comparison of all the solutions that have been generated throughout the present iteration. The system keeps the record of the best solution that has been found up to now as the global best solution S^{gb} and it will be updated to the best solution in the present iteration i, S^{ib}, when it is better than S^{gb}. A process of updating global pheromone is performed at the end of every iteration cycle. This process consists of two steps. In the first step, an evaporation step, all the pheromone slightly evaporates based on the evaporation rule stated below:

$$\tau_{js}(t+1) \leftarrow (1-\rho)\tau_{js}(t) \tag{2}$$

The evaporation parameter is represented as $\rho \in [0,1]$. Furthermore, to intensify the pheromone of the selected nodes to S^{gb}, the second phase is performed:

$$\tau_{js}(t+1) \leftarrow \tau_{js}(t+1) + \rho\Delta\tau, \ (j,s) \in S^{gb} \tag{3}$$

where parameter $\Delta\tau$ denotes the amount of pheromone that has been intensified. The above loop is iterated until a specific termination criterion is met. This can occur where a satisfying solution is obtained or the function has been evaluated a maximum number of times.

4 Proposed method

In this section, a novel hybrid unsupervised feature selection algorithm based on BAS and clustering which is called FSCBAS is described. In the first phase of the proposed method, the features are clustered. Then, in a *parallel process* by applying modified BAS (BASM) to each cluster, the features of clusters are ranked. In the second phase, in an iterative forward process, the best features from each cluster are competed by the BASM method. This phase is repeated until the subset of desired features has been completed. In the following, at first BASM is explained by details, then FSCBAS in three steps is described.

4.1 BASM: Improved binary ant system and mutation strategy

The ACO-related methods by using full graphs can often achieve an acceptable solution; however, the complexity of the algorithms is extremely high, and it would make challenges in high-dimensionality datasets. On the other hand, in ACO-based works which rely on a linear graph, although the time complexity significantly decreases, the search space has been limited and these algorithms can easily get trapped in a local optimum. Note that most of the previous studies do not take into account the redundancy between features efficiently. Our proposed method considers two strategies to overcome the limitations mentioned above. In the subsequent sections, the search space representation, the redundancy between features and the mutation strategy will be elaborated.

4.1.1 Representation of the search space

In the ACO algorithm, representing a suitable model for the search space is a critical factor. In order to decrease the computational complexity, In the proposed method, a circular connected undirected weighted graph $G = < F, E >$ for search space representation is offered where $F = \{F_1, F_2, \ldots, F_n\}$ denotes a set of initial features in which each feature represents a node in the graph, and the graph edges are denoted by $E = \{(F_i, F_j): F_i, F_j \in F\}$. The weight of edge $(F_i, F_j) \in E$ is set to the pheromone value between F_i and F_j, shown by τ_0^i or τ_1^i. The representation of the feature selection problem is shown in Fig. 4.

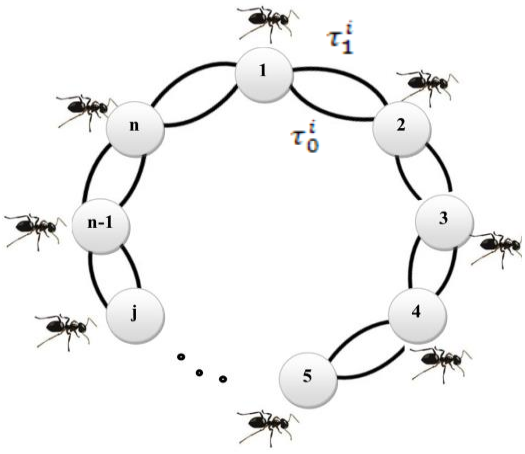


Fig. 4. The graph representation for feature selection problem.

In this process, for computing the similarity value between all features, the absolute value of the Pearson correlation coefficient criterion is applied (Benesty, Chen, Huang, & Cohen, 2009) according to the following equation:

$$sim(F_i, F_j) = abs\left(\frac{\sum_{s=1}^{|S|}(F_{i_s} - \bar{F}_i)(F_{j_s} - \bar{F}_j)}{\sqrt{\sum_{s=1}^{|S|}(F_{i_s} - \bar{F}_i)^2}\sqrt{\sum_{s=1}^{|S|}(F_{j_s} - \bar{F}_j)^2}}\right) \quad (4)$$

where F_i and F_j are two feature with S-dimensional. The number of samples presented by $|S|$, and F_{i_s} and F_{j_s} are features of sample s. Variables $\bar{F}_i$ and $\bar{F}_j$ represent the mean values of feature F_i and F_j, respectively. If two features are completely similar then the similarity between them is equal to 1 and it is equal to 0 for completely dissimilar features (Kabir, Shahjahan, & Murase, 2011). In many datasets, the similarity values between features are so close that makes it difficult to decide. Therefore, softmax scaling (Martens, Baesens, & Fawcett, 2011) which is a nonlinear method to normalization is applied to scale the similarity values into the range $[0, 1]$ as below:

$$w_{ij} = \frac{1}{1 + exp\left(-\frac{w_{ij} - \bar{w}}{\sigma}\right)} \quad (5)$$

where w_{ij} is the similarity value between features F_i and F_j, $\bar{w}$ and σ are the mean and variance of all the similarity values, respectively. $\hat{w}_{ij}$ is the normalized value of the similarity between features F_i and F_j.

In the proposed method, two values are assigned to each feature as heuristic information to select/deselect the feature. One of these pieces of information is obtained from the term variance corresponding to the feature:

$$\eta_i^0 = \frac{1}{|s|}\sum_{l=1}^{|s|}(x_i(l) - \bar{x}_i)^2 \quad , \eta_i^1 = 1 - \eta_i^0 \quad (6)$$

where η_i^0 and η_i^1 are, heuristic information assigned to the ith feature for selecting and deselecting the feature, respectively. The value of $|s|$ is the number of dataset samples, x_i is the ith feature, and $\bar{x}_i$ is the average value of the ith feature over all samples. It should be mentioned that η_i^0 has been normalized with equation (5), therefore η_i^1 is also mapped into the range $[0, 1]$, too.

4.1.2 Redundancy reduction strategy

In related works, to overcome the challenge of redundancy between the subset of selected features, a feature will be selected which have the least similarity to all chosen features subset up to now. These works often compute the average of similarity between the current processing feature and the previous selected feature (Tabakhi & Moradi, 2015; Tabakhi et al., 2014; Tabakhi et al., 2015).

Although in some cases the average of similarity of the current feature with the previous selected features is low, it might be high correlated to some of these features. Therefore, to overcoming this challenge the average of correlation of the current processing feature with the "*q of the most correlated selected features with the current features*" is computed. This means that to computing the average similarity of the current processing feature, only q features among all the previous selected ones with the greatest similarity to the current feature is considered. The final similarity value is the average similarity between the current feature and the q most similar features, according to the following equation:

$$\eta_{cor\,i}^{\;1} = \frac{1}{|q|}\sum_{l\in\Gamma_k} sim(i,l)\ ,\ \eta_{cor\,i}^{\;0} = 1-\eta_{cor\,i}^{\;1} \tag{7}$$

where Γ_k contains the q nodes/features with a maximum similarity between feature i and each feature in a $\mathrm{subset(k)}$, where $\mathrm{subset(k)}$ contains the nodes selected so far by ant k.

4.1.3 Mutation strategy

To avoid falling into local optima and overcome the search space limitation, several efficient factors are described as follows:

1- *Random motion of ants*, meaning that the artificial ants start moving on the mapping graph from a randomly selected node.

2- *Random direction of ants*, which means that the clockwise/counterclockwise moving direction of ants should be randomly determined.

3- *Damped mutation at special nodes*, which is described below:

The idea of damped mutation has been inspired by the mutation operator of the genetic algorithm and the cooling of a material (searching for the minimum energy state) of the simulated annealing algorithm. Damped mutation will be applied to avoid premature convergence and increase the states in the search space. Simply, by apply this mutation strategy, some of the nodes on the graph are exchanged to modify the order of the nodes. According to this approach, at the first iteration, ants are placed on the graph randomly, then the mutation operator might exchange special nodes on the graph based on the *mutation condition*; in the later iterations if the mutation has happened, ants move on the new mutated graph. The mutation condition occurs when the difference between *pheromone ratio* in the current iteration and that in the previous iteration is less than the *mutation rate*. The pheromone ratio is the value of pheromone on selected nodes in each iteration for every node. To compare with *mutation rate*, this variable should be normalized (Equation 8).

$$Pheromone\ ratio(t) = \frac{\sum_j FSC_j^{\,0}\times\tau_j^0(t)}{\sum_{j,l} FSC_j^{\,l}\times\tau_j^l(t)}\ ,\ j=1,...,n,\ l\in\{0,1\} \tag{8}$$

The values of $\mathrm{l}=0$ and $\mathrm{l}=1$ correspond to the select/deselect state pheromone, respectively. The parameter n denotes the number of features. $\mathrm{FSC_j^0}$ is the state counter corresponding to feature j,

when it is selected by ants. FSC_j^l is the state counter corresponding to feature j in state select/deselect. $\tau_j^0(t)$ is the pheromone value of feature j in the selected state at iteration t. The value shown by $\tau_j^l(t)$ is the pheromone value of feature j in iteration t in state select/deselect. Finally, $\mathrm{pheromone\ ratio}(t)$ is the value of pheromone on selected nodes in iteration t.

Mutation rate is a parameter which determines the rate of mutation. In the first iteration, the mutation rate value is equal to its maximum value (equal to 1). As iterations pass, mutation rate decreases in each mutation action. The reason for this policy is that the explorative ability of the algorithm is high at the early stages of the learning; then by decreasing the rate of mutation, as simulated annealing and genetic algorithm, the local search capabilities and the convergence of algorithms will be increased, gradually. By applying this technique, a balance between exploration and exploitation will be provided in the search space. Also, the policy for selection of nodes for the mutation is important, as the nodes will be less frequently selected by ants are replaced by those that have been selected most frequently.

In each iteration of ACO, an ant is placed randomly on the graph. Then, the clockwise or counterclockwise movement of the ant is selected randomly. Continuously, each ant traverses the circular graph, according to the movement direction, until it comes back to the initial position. In passing, the ant selects or deselects a feature based on the probabilistic *state transition rule*. The state transition rule seeks to select features with highest term variance values (features with highest term variance values often have more information) and lowest similarities to the previously selected features that are described by Equation (9). The number of times that specific nodes are selected/deselected by ants is defined as feature state counter. Then, at the end of each iteration, the pheromone of each node is updated through the feature state counter and an *updating rule* according to Equation (12). After the stopping criterion is met, the features with the highest selecting pheromone values will be selected.

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

Algorithm 1. BASM pseudo-code

Input:

D : $p \times n$ matrix, an n-dimensional dataset with p patterns.

Itr : Number of iterations.

A : Number of ants.

τ_0 : The initial amount of pheromone for each node state.

n_m: Number of mutation.

Output:

Sorted features based on pheromone.

Begin algorithm

1: *Represent* the problem with a circular graph.
2: *Compute* the heuristic information (using Equation 4 and 6).

3: *Set* the initial intensity of pheromone τ_i^l on the nodes to constant value τ_0, $\forall$ i=1…n *and* l=0,1.
4: *Create* lists **Startnodes** and **Endnodes** with the size of n_m.
5: **For t=1 to Itr do**
6: *Initialize mutation rate* =1;
7: *mutation condition* = pheromone ratio(t) − pheromone ratio(t− 1) < *mutation rate*;
8: **If** *mutations condition* is true
9: *Do Mutation* on Startnodes and Endnodes in graph using *Algorithm 2*;
10: mutation rate − = 0.1 × mutation rate;
11: **End If**
12: *Initialize* $FSC_i^l = 0$, $\forall$ i=1…n *and* l=0,1
13: *Place* the ants randomly on the mutated graph.
14: **For j=1 to A do**
15: *Determine* graph direction (clockwise or counter-clockwise) randomly.
16: *Ants moves* until back to their start node.
17: *Select/deselect* a node by ants according the state transition rule (Equations 9-11)
18: $FSC_i^s + +$;
19: **End For**
20: *Calculate* pheromone ratio (Equation 8).
21: *Update* pheromones (Equation 12).
22: *Set* Startnodes and Endnodes to the best and worst nodes based on FSC;
23: **End For**
24: *Sort* the features by select state pheromones in descending order

End algorithm

Algorithm 2. Mutation pseudo-code

Input:

G: graph

StartNodes list

EndNodes list

Output:

G': modified graph

Begin algorithm

1: Exchange StartNodes with EndNodes in G by keeping their pheromones;

2: Set the edges of G;

3: Return G'=G;

End algorithm

4.1.4 Transition rule

The state transition rule is probabilistic and designed based on the mixture of the node pheromone values and heuristic information, as follows:

$$P_j^l = \frac{(\tau_j^l)^\alpha \times (\eta_j^l)^\beta \times (\eta_{cor\,j}^{\ l})^\beta}{\sum_h (\tau_j^h)^\alpha \times (\eta_j^h)^\beta \times (\eta_{cor\,j}^{\ h})^\beta} \quad , l, h \in \{0,1\} \tag{9}$$

where P_j^l is the probability of selecting ($l = 0$) or deselecting ($l = 1$) feature j. Similarly, τ_j^l is the pheromone value corresponding to feature j, where τ_j^0 is the pheromone for selecting and τ_j^1 for deselecting of feature j. Parameter η_j^0 is considered as the heuristic information of feature j (term variance of feature j), where it is used for the select state. Parameter η_j^1 is used for the deselect state and is equal to $(1 - \eta_j^0)$. The average similarity between feature j and the previously selected q features with the greatest similarity to feature j is shown by $\eta_{\mathrm{cor}_j^1}$, which is used for the deselect state rule. Parameter $\eta_{\mathrm{cor}_j^0}$ is used for the select state and is equal to $(1 - \eta_{\mathrm{cor}_j^1})$. The values of parameters η and η_{cor} are computed according to equations (6) and (7), respectively. Parameters α and β are constant parameters that are used to control the importance of pheromone versus heuristic information. Finally, according to Equations (10) and (11), selecting (S=1) or not selecting (S=0) node i is determined by applying either greedy or probabilistic rules, respectively. To avoid falling into local optima and increasing search space, the roulette wheel is used such that based on threshold $\theta = 0.7$ the selecting or deselecting process is performed. If the random value γ is smaller than θ, the greedy rule is applied according to following formula:

$$S = \arg\max_{l \in 0,1} P_j^{\ l}, \quad if \ \gamma \le \theta \tag{10}$$

Otherwise, the probabilistic rule is applied as follows:

$$S = \begin{cases} 1 & if \ \gamma_0 \le P_j^l \\ 0 & else \end{cases}, \quad if \ \gamma > \theta \tag{11}$$

where, γ_0 is a random value.

4.1.5 Pheromone Update

The update pheromone rule is applied to update the pheromone on all edges. The pheromone values are updated after each ant has completed its traverse, by the following equation:

$$\tau_i^l(t+1) = (1-\rho) \times \tau_i^l(t) + w \times \frac{FSC_i^{\ l}}{\sum_{j,l} FSC_j^{\ l}} \ , j = 1, \ldots, n \ , l = 0,1 \tag{12}$$

where $\tau_i^l(t+1)$ and $\tau_i^l(t)$ are the amounts of pheromone of feature i at times t and $t+1$. The values of $l = 0$ and $l = 1$ correspond to the select/deselect state pheromone, respectively. The parameter ρ represents pheromone evaporation, and FSC_i^l is the state counter corresponding to ith feature. The variable n denotes the number of features, and w is a constant weight.

4.2 FSCBAS: Feature Selection based on Clustering and BASM

One of the recent feature selection method based on clustering is the GCACO method presented by Moradi et al. (Moradi & Rostami, 2015). In this approach, features are clustered with the Louvain community detection algorithm and then in each cluster, the features are represented in a complete graph that caused to increases the complexity significantly. Furthermore, in each cluster, an ant select a feature randomly based on a threshold value. Another shortcoming of this method is selecting at least one feature from each cluster which is not essential. Also, processing the clusters is done consecutive however the order visit the clusters is an efficient issue in feature selection method. In order to overcome the mentioned challenges, a method is provided that can be implemented in *semi-parallel processes* and can increase the efficiency, significantly. Consequently, the proposed method improves performance and computational complexity**,** simultaneously. After features clustering, instead of choosing the best features from each cluster and selecting them as the final subset features, the *competition* between the best features of all clusters is defined so that the top ones will be selected from each cluster and they compete again in the next stage. One of the differences between the proposed method and the other forward feature selection methods is that in proposed method *some of the top features* compete together, repeatedly. In contrast, in other methods, *all features* compete together. According to our method, after the feature clustering and ranking by BASM method, top features from the corresponding classes are placed on the circular graph to compete with the others. By using this policy, the feature with lower rank in a cluster has the chances of choosing in the final feature set relative to the feature with a higher rank in the other cluster. Therefore, the selecting process consists of two steps. The first step is selecting the top feature of each category, based on their pheromones values, which is called *"Parallel selection"*, and the second step is competing obtained top features from clusters until reaching the collection Best of the Bests, which this step is called *"Sequential Selection"*. In the following sub-sections, the feature clustering and the representation problem in a proper form for the use of the BASM algorithm are explained. Then, the methodology of the proposed method is described in two steps.

4.2.1 Feature clustering

In this section, the features are divided into several clusters based on the similarity values between them. It should be considered the proposed method is not sensitive to the clustering method. The Louvain community detection algorithm is applied to split graph to small partitions (Aynaud & Guillaume, 2010) in which detects the communities by maximizing a modularity function. This is a simple, efficient and easy way to implement algorithm which can be used to detect communities in very large graphs (Moradi & Rostami, 2015). Therefore, we applied Louvain community detection algorithm on the similarity graph of nodes.

4.2.2 Step1: Parallel feature selection

In the next step, BASM is performed in each cluster, separately. In the other words, the features are placed on the search space representation of the proposed BASM method and the ants start to search and pouring pheromone. The clustering and the graph formation of the features are shown in Fig. 5.

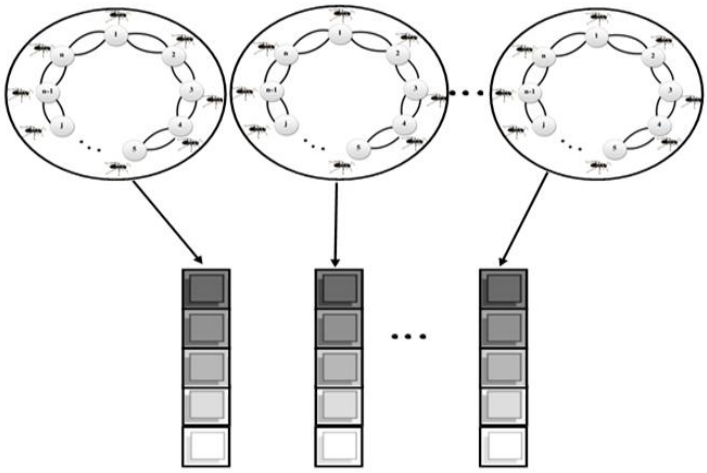


Fig. 5. Feature clustering and perform BASM on clusters.

Since the clusters are completely separate, each cluster scrolled by a batch of ants in parallel. As a result, time at this stage decreases significantly in compare of other clustering based feature selection methods in which the search processing is sequential and does not have parallel processing capabilities. After processing clusters and pouring pheromones by ants based on the BASM method, the features in each cluster are sorted according to their pheromones. Therefore, the salient features of each cluster are placed at the top of the lists.

4.2.3 Step2: Sequential feature selection

In the second step, a new iterative forward feature selection algorithm is performed. At each iteration, the K top remaining features extracted from each cluster, then BASM is applied between these top ones and the K_0 best features are selected. Aftermore, the selected features are removed from their clusters and clusters are updated to re-select the K features. This cycle will continue until the desired feature subset are completed. Note that although this step is repeated $\frac{n_f}{K_0}$ times (n_f is Number of selected features), it is much smaller than the processing the total number of features. Therefore, the processing time of this graph is far less than the initial graph. Given that this step is repeated several times, it may be time consuming, as it is still less than the processing of initial graph with all the features. Fig. 6 shows this step of the proposed FSCBAS method in which $\mathrm{K}_0 = 3$.

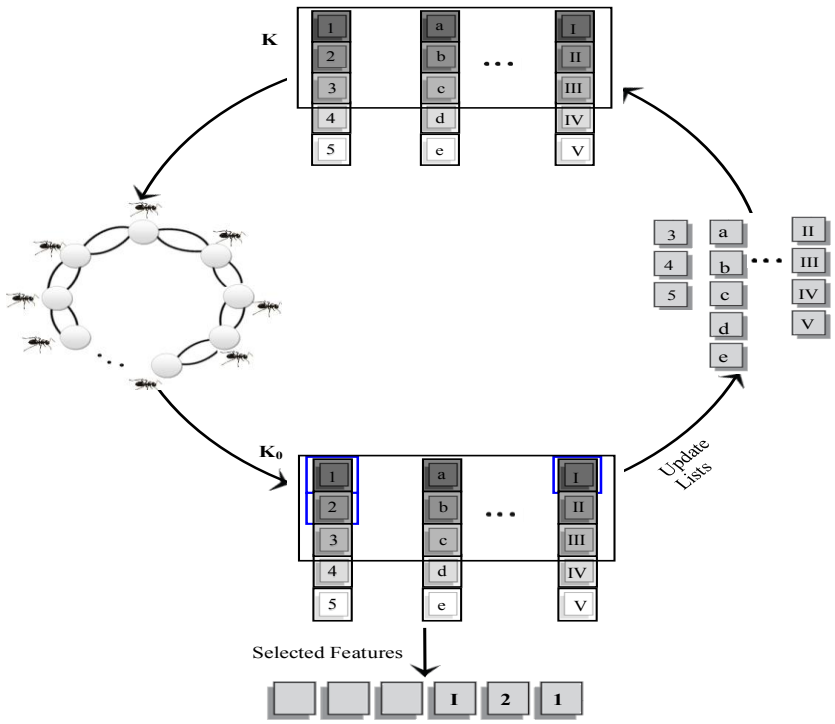


Fig. 6. Search strategy in the FSCBAS

The pseudo-code of the proposed method is shown in Algorithm 3. In lines 1, 2, the Parallel feature selection is done that consist of feature clustering, and apply BASM in each cluster. In the Parallel feature selection, the sorted features based on pheromone values in each cluster called *"Lists-of-Features"* are returned. In lines 4-9 the Sequential feature selection is done. In line 4, the loop is repeated until the number of desired feature subset is completed. In Line 5, K top features are selected from *each cluster*. Then, also the K best features are selected from *all clusters* in line 6. Since in order to choose the best features from all clusters, the pheromone values of clusters must be comparable, so that the pheromones of the clusters should be normalized by Equation 13.

$$\hat{\tau}_{ijl} = \tau_{ijl} \times \frac{L_{C_j}}{n} \quad , i = 1,...,L_{C_j} \ , j = 1,...,n_c \ , l = 0,1 \tag{13}$$

Where τ_{ijl} is the pheromone value of feature i from cluster j in $l = 0,1$ to select and deselect state, respectively. L_{C_j} is the number of features in cluster j, and n is the number of all features. The $n_c$is a number of clusters. $\hat{\tau}_{ijl}$ is the normalized pheromone feature i.

In Line 7, the top features achieved from clusters are placed on the small binary circular graph and BASM method executed on them. Then K_0 features are selected with the largest pheromone from these top features and added to the list of final selected features in line 8. These K_0 features are removed from the Lists-of-features and then the lists are updated and the loop is repeated until the desired subset features are completed. Finally, the final features are selected in line 10.

Determining the number of clusters is important. The best number of clusters obtained from an *empirical formula* in Equation 14.

$$n_c = \left\lceil \frac{1}{2}\mu \times n_f \times \log n \right\rceil \quad (14)$$

where n_f is the number of desired subset feature, and μ is the average of the standard deviation of the similarities between all features.

Algorithm 3. FSCBAS pseudo-code

Input:

D : $p \times n$ matrix, an n-dimensional dataset with p patterns.

n_f: Number of selected features.

n_c: Number of clusters.

Output:

D : $p \times n_m$ matrix, reduced dimensionality of the dataset.

Begin algorithm

1: *Clustering* features.
2: Lists-of-features: *Perform* BASM algorithm on each cluster and return sorted features from them.
3: $K = n_f/n_c$; $K = K_0$;
4: **while** subset features are completed
5: *Select* maximum K top features from *each cluster*.
6: *Select* K features between *deselected features from all clusters* based on pheromone.
7: *Perform* BASM on selected features in lines 5, 6.
8: *Select* K_0 best features, add to subset features and remove from *Lists-of-features.*
9: **End while**
10: *Build* D' from D by keeping the subset features.
End algorithm

5 Experiments and results

In this section, some experiments are performed on the proposed feature selection method (FSCBAS) method, and the results of the algorithm are reported and analyzed. For evaluation of the performance of the proposed method, it is compared to some well-known or state-of-the-art univariate and multivariate feature selection algorithms based on the filter model, including, random subspace method (RSM) (Lai, Reinders, & Wessels, 2006), Laplacian score (LS) (He, Cai, & Niyogi, 2005a), mutual correlation (MC) (Ghazavi & Liao, 2008; Haindl, Somol, Ververidis, & Kotropoulos, 2006), and relevance-redundancy feature selection (RRFS) (A. J. Ferreira & M. A. Figueiredo, 2012; A. J. Ferreira & M. A. T. Figueiredo, 2012), and some recently published methods, including unsupervised ant colony optimization feature selection (UFSACO) (Tabakhi et al., 2014), graph clustering based ACO (GCACO) (Moradi & Rostami, 2015), and microarray data classification using gene selection based on ACO (MGSACO) (Tabakhi et al., 2015). In the following, details on the experiments, such as the selected datasets, the classifiers and evaluation criteria, the parameter setting, the computational complexity analysis and, finally, the experimental results, and discussion are explained below.

5.1 Datasets

This paper uses several well-known real-world datasets from the UCI (University of Carolina, Irvine) repository (Asuncion & Newman, 2007), including *Wine, Ionosphere, Hepatitis, SpamBase, Arrhythmia, Madelon, Colon,* and *Leukemia* from the Bioinformatics Research Group at the Universidad Pablo de Olavide (Bolón-Canedo, Sánchez-Marono, Alonso-Betanzos, Benítez, & Herrera, 2014) to compare FSCBAS with other methods. These datasets have been used in many machine learning studies including feature selection and cover small, medium and large feature dimension ranges (A. J. Ferreira & M. A. T. Figueiredo, 2012; Gheyas & Smith, 2010; Martínez Sotoca & Pla, 2010; Unler, Murat, & Chinnam, 2011). The characteristics of these datasets are summarized in Table 3.

Table 3 Details on the real-world datasets taken from UCI

Type	Dataset	Features	Classes	Patterns
Physical	Wine	13	3	178
Life	Hepatitis	19	2	155
Physical	Ionosphere	34	2	351
Computer	SpamBase	57	2	4601
Life	Arrhythmia	279	16	452
Artificial dataset	Madelon	500	2	4400
Microarray	Colon	2000	2	62
Microarray	Leukemia	7129	2	72

5.2 Classifiers and evaluation criteria

Since the proposed FSCBAS method is a filter approach where no classifier is utilized to evaluate the selected feature subset, we expect FSCBAS to lead to the enhancement of the results of different classification methods. Therefore, in experiments, several classifiers including support vector machine (SVM), decision tree (DT), K-nearest neighborhood (KNN), and Random Forest (RF) were used for evaluation and comparison of FSCBAS with the other methods. These classifiers have been implemented in the WEKA (Hall et al., 2009) software package.

Classification error rate, namely *CER* (in percent), is an effective way for feature selection validation (Yishi Zhang, Yang, Xiong, Wang, & Zhang, 2014) which has been used in this article to evaluate the feature subsets produced by various methods. Classification error rate is the percentage of incorrectly classified samples and it is computed as follows:

$$Classification\ \ error\ \ rate(CER) = \frac{number\ of\ \ incorrectly\ classified\ samples}{number\ of\ \ samples} \tag{15}$$

5.3 Parameter setting

Our experiments were executed on a PC with an Intel Core i5 CPU system with 2.7 GHz frequency and 6 GB RAM. We implemented all of the algorithms using C# .NET programming language.

The parameters on the proposed method were set as can be seen in Table 4. During the program implementation of the FSCBAS algorithm, the maximum number of iterations was set to $\mathrm{Itr} = 50$, which is enough for the convergence. The two parameters $\propto$ and β controlled the balance between exploitation and exploration. The pheromone evaporation rate was set to ρ=0.2. The parameter τ_0 denoted the initial pheromone value for each feature and was set to 0.2. The parameter q denoted the number of features used for calculation of the average similarity to the current node. The number of mutation was set to $\mathrm{n_m} = 3$. For each dataset, the number of ants was considered to be equal to the number of features: $\mathrm{A} = \mathrm{n}$. When a dataset has more than 100 features, parameter A was set to 100. The parameter w denoted the pheromone intensity update and the $\mathrm{n_c}$ was the number of clusters which obtained from an empirical formula.

Table 4. Parameter setting for the proposed method

Notation	Value	Description
Itr	*50*	Iteration
α	*2*	Control parameter
β	*1*	Control parameter
ρ	*0.2*	Pheromone evaporation
τ_0	*0.2*	Initial pheromone
q	*3*	Number of features for evaluation of similarity
n_m	*3*	Number of mutation
A	*n*	Number of ants
w	*1.2*	Pheromone intensity updating rate

5.4 Computational complexity analysis

At first, the complexity of BASM method will be investigated and then the complexity of the proposed method (FSCBAS) can be considered completely. According to Algorithm1, the similarity between each two features is evaluated with a computational cost of $O(n^2)$ where the number of features is determined by *n*. Term variance is used to represent the relevance values of each feature, which requires a cost of $O(n)$ (line 2). Moreover, in an iterative process, the best subset of features can be found (lines 5-23). In this cycle, each iteration contains three main stages. In the first stage, the mutation strategy is applied (lines 6-11). Since four edges are exchanged in each mutation, it requires $O(4n_m)$. Since the parameter n_m holds a constant value; consequently, the complexity of this step is $O(1)$. In the second stage, ants generate candidate solutions (lines 13-19) with a

computational cost of $O(2An)$, where A is the number of ants. In the third step (line 21), it requires $O(2n)$ to update the pheromone values. So far, the overall computational complexity is $O(n^2 + n + I(2An + 1 + 2n + nlogn))$, where I is the number of iterations. In order to reduce this complexity, the similarity values of each feature will be computed and saved just one time and then the saved values in every step will be used, so that evaluation of the similarity can be removed. Consequently, the overall computational complexity of the proposed method is decreased to $O(n + 2IAn + I + 2In + Inlogn)$. Finally, sorting the features based on the pheromone of the selected state (line 24) has a complexity of $O(nlogn)$. According to the above description, the time complexity of the proposed method is $O(n + 2IAn + I + 2In + \mathrm{I}nlogn + nlogn) = O(nlogn)$.

According to algorithm 3, the feature clustering by Louvain algorithm needs complexity $O(nlogn)$. (line 1). In the next step, the BASM method is performed in each cluster. Based on above mentioned the BASM method has complexity $O(\hat{n}log\hat{n})$ where $\hat{n}$ is the average number of features of the clusters and in average case $\hat{n} = \frac{n}{n_c} \ll n$ (lines 2, 3). Therefore, the complexity of this step in the worst case is $O(nlogn)$. Then, the process of selecting K top features and K features from unselected features of clusters in lines 5, 6 have $O(nlogn)$ cost. Furthermore, the complexity of line 7 which run BASM on the selected features from previse steps is $O(n_{kc}logn_{kc})$ where $\mathrm{n_{kc}} = \mathrm{K} \times \mathrm{n_c}$. Since, n_{kc} is smaller than $\mathrm{n_f}$ and $\mathrm{n_{kc}} << n$ therefore, $(n_{kc}logn_{kc}) \ll (nlogn)$ and in the worst case the complexity of this step is $\mathrm{O(nlogn)}$, consequently the complexity of lines 4-10 is $O(\frac{n_f}{K_0} \times (1 + n + nlogn + nlogn + 1 + 1)) = O(nlogn)$. Overall, it can be concluded that the complexity of the proposed method is $O(nlogn)$.

It is worth noting that the search space representation of the proposed method has two directed arcs between every two connected nodes on the linear graph. Consequently, graph edge complexity is decreased to $O(2n)$, while in most of the recent algorithms with search spaces based on full graphs, graph edge complexity is $O(n^2)$. In these algorithms, therefore, $O(n^2)$ heuristic information and pheromone should be computed and stored. It is clear that the proposed method has much lower computational complexity than other filter-based methods. Furthermore, the proposed method is a filter-based method and the feature selection process does not require any classifier, therefore it has lower computational complexity than other wrapper-based methods. Table 5 Shows the computational complexity of different filter-based feature selection methods.

Table 5. Comparison of the computational complexity of different filter methods.

Method	**Type**	**Complexity**	**Description of notations**
LS **(He et al., 2005b)**	Univariate, Unsupervised	$\mathrm{O(p^2n)}$	n: total number of features
MRMR **(H. Peng et al., 2005)**	Multivariate, Supervised	$\mathrm{O(nn_fp)}$	p: number of instances
UFSACO **(Tabakhi et al., 2014)**	Multivariate, Unsupervised	$\mathrm{O(n^2p + ln_fn)}$	$\mathrm{n_f}$: number of selected features
MGSACO **(Tabakhi et al., 2015)**	Multivariate, Unsupervised	$\mathrm{O(In^2A)}$	l: maximum number of iterations
GCACO (**Moradi & Rostami, 2015**)	Multivariate, Unsupervised	$\mathrm{O(n^2p + ln)}$	
FSCBAS (**Proposed method**)	Multivariate, Unsupervised	$\mathrm{O(nlogn)}$	

5.5 Experimental results

In the following, the classification error rate of the proposed method is compared to different unsupervised methods, then execution time and statistical test results are investigated. In the end, the proposed method is compared with a supervised feature selection method, and an extensive comparison is particularly provided on high-dimensional data.

5.5.1 Classification Error Rate

Different classifiers were used in the first experiment for assessment of the performance of the proposed algorithm on different datasets. In this paper, K-fold cross validation was used to assessing the performance of a classifier. The proposed FSCBAS method was compared to a number of unsupervised filter based methods with respect to average classification error rate (in percent) of ten different simulations. For evaluated the proposed method, SVM, DT, RF, and KNN classifiers were used. The methods included MGSACO, UFSACO, GCACO, MC, RSM, RRFS, and LS. Moreover, the proposed method was compared to a well-known supervised filter base method named as mRmR (H. Peng et al., 2005). The best result in each case was shown in bold face and underlined and the second best is in bold face.

Tables 6-8 shows the comparison classification error rates (CER) of the proposed method with the other ACO-based feature selection methods on the UCI data sets.

According to Table 6, the lowest CER on SVM classifier is achieved by the proposed FSCBAS method compared to the other filter-based methods over all the datasets. For example, for colon dataset the CER obtained by the proposed method is 12.41% while for MGSACO, GCACO, UFSACO, RSM, MC, RRFS, and LS this value is reported 16.12%, 18.01% 17.9%, 17.21%, 18.18%, 21.81% and 38.18%, respectively. Moreover, the proposed method by obtaining 17.27% average CER on all the datasets is better than MGSACO by 21.16%, GCACO by 18.54%, UFSACO by 21.07%, RSM by 25.76%, MC by 23.68%, RRFS by 29.28%, and LS by 24.34%. As can be seen from this table, the CER standard deviation (std) of the proposed method outperforms of the other methods specially, ACO-based methods. For example, for wine dataset, the std value is 0, compared to MGSACO, GCACO, UFSACO, RSM, MC, RRFS, and LS by 2.52%, 2.6%, 2.04%, 3.81%, 2.04%, 3.81%, and 1.49% values, correspondingly. In general, Table 6 results show that the proposed method by obtaining lowest CER acquired the first place for SVM classifier in all cases over all the datasets, and it proved the excellent performance of the proposed method on SVM classifier.

Table 6. Average classification error rate over 10 runs of the unsupervised feature selection methods using SVM classifier.

Datasets	# selected features		FSCBAS	GCACO	MGSACO	UFSACO	RSM	MC	RRFS	LS
Wine	6	CER	**2.8**	**4.9**	5.39	3.98	10.44	11.17	32.02	5.5
		Std	0	2.6	2.52	2.04	3.81	2.04	3.81	1.49
Hepatitis	6	CER	**18.45**	**16.36**	18.9	19.22	18.77	18.86	20	19.05

		Std	3.74	2.83	1.13	0.4	0	1.81	0	0.27
Ionosphere	15	CER	14.38	13.21	18.4	15.04	16.91	**12.91**	**13.16**	16.66
		Std	3.57	2.41	2.09	2.12	0.66	0.88	1.29	1.55
SpamBase	24	CER	**14.22**	15.49	18.14	17	17.27	29.06	25.81	**12.38**
		Std	0.22	2.12	2.67	1.12	1.68	1.76	0.75	0.13
Arrhythmia	20	CER	**32.52**	**38.62**	43.45	45.61	47.14	46.62	44.02	40.26
		Std	3.63	5.2	1.44	0.29	0.5	0.34	2.32	2.07
Madelon	40	CER	38.8	**35.39**	39.02	39.25	49.50	**20.28**	45.07	50.96
		Std	2.07	5.58	0.27	0.24	1.15	1.85	0	0
Colon	40	CER	**12.41**	18.01	**16.12**	17.9	17.21	18.18	21.81	38.18
		Std	1.31	2.23	2.94	3.26	1.36	3.74	6.37	0
Leukemia	40	CER	**4.58**	**6.33**	9.86	10.55	28.82	32.35	32.35	11.76
		Std	0.67	2.35	5.17	3.28	0	0	0	0
Average		CER	**17.27**	**18.54**	21.16	21.07	25.76	23.68	29.28	24.34
		Std	1.9	3.17	2.28	**1.59**	**1.15**	1.55	1.82	0.69

Table 7. Average classification error rate over 10 runs of the unsupervised feature selection methods using DT classifier.

Datasets	**# selected features**		**FSCBAS**	**GCACO**	**MGSACO**	**UFSACO**	**RSM**	**MC**	**RRFS**	**LS**
Wine	6	CER	**6.17**	**6.43**	7.07	6.91	14.94	10.39	29.77	7.41
		Std	0	2.98	3.47	3.02	2.74	3.71	0	2.32
Hepatitis	6	CER	20.64	19.69	19.74	22.96	21.41	**19.48**	20.64	**18.9**
		Std	0	1.14	0.92	1.22	2.56	0.79	0	1.61
Ionosphere	15	CER	11.68	9.46	13.3	10.94	**8.91**	**8.74**	11.59	10.68
		Std	1.41	1.98	2.14	1.51	0.9	1.18	2.04	2.33
SpamBase	24	CER	10.32	10.88	10.17	10.92	14.1	12.47	**7.34**	**8.33**
		Std	0.22	1.12	1.52	0.51	1.42	1.12	0.22	0.13
Arrhythmia	20	CER	**40.24**	**43.38**	50.06	49.11	46.74	45.79	46.17	44.35
		Std	1.37	4.13	2.05	2.32	1.38	0	1.85	3.16
Madelon	40	CER	32.6	**18.23**	**19.88**	20.51	50.24	49.8	42.86	47.96
		Std	1.25	2.38	0.57	0.67	0.79	0.76	0	1.22
Colon	40	CER	**13.87**	15.77	**14.83**	17.09	29.19	22.25	32.25	30.64

		Std	0.83	2.39	2.82	4.18	3.98	4.92	8.5	0
Leukemia	40	CER	**6.38**	14.26	**9.72**	22.08	20.83	20.83	6.94	27.77
		Std	0	1.52	4.29	2.96	0	0	0	0
Average		CER	**17.74**	**17.26**	18.1	20.07	25.80	23.72	24.7	24.51
		Std	**0.64**	2.21	2.22	1.72	1.72	1.56	1.58	**1.35**

Table 7 shows the CER values of different feature selection methods on the DT classifier. As can be seen, the proposed method on wine, arrhythmia, colon, and leukemia by 6.17%, 40.24%, 13.87% and 6.38% CER values obtained the first place in comparison with the other filter based methods. Furthermore, Table 6 and Table 7 reports similar results on the KNN and RF classifiers, respectively. Table 8 demonstrate the proposed method has the best results for wine, colon, and leukemia on the KNN classifier, and it has acceptable results on the other datasets in comparison with the other methods. Moreover, according to this table, average CER of the proposed FSCBAS with a little difference (0.06%) by GCACO has the second-best rank between the eight filter-based feature selection methods.

Table 8. Average classification error rate over 10 runs of the unsupervised feature selection methods using KNN classifier.

Datasets	**# selected features**		**FSCBAS**	**GCACO**	**MGSACO**	**UFSACO**	**RSM**	**MC**	**RRFS**	**LS**
Wine	6	CER	**2.8**	4.32	5.39	**3.98**	10.44	11.17	32.02	5.5
		Std	0	2.78	2.52	2.04	3.81	2.04	0	1.49
Hepatitis	6	CER	20.64	**18.64**	**18.9**	19.22	20.64	19.93	20.64	20.77
		Std	3.74	1.33	1.13	0.41	0	1.81	0	0.27
Ionosphere	15	CER	**14.38**	14.64	18.4	15.04	**11.99**	**11.99**	16.52	15.98
		Std	3.57	4.14	2.1	2.12	0.66	0.89	1.3	1.56
SpamBase	24	CER	14.22	12.96	18.14	**11.67**	16.47	14.46	**11.28**	13.10
		Std	0.22	1.32	2.68	0.68	1.37	1.23	0.3	0.23
Arrhythmia	20	CER	**42.69**	46.25	44.75	49.13	48.62	45.7	44.95	**43.51**
		Std	1.33	5.38	1.84	1.2	1.93	0.11	1.73	1.82
Madelon	40	CER	40.2	**23.79**	**24.15**	24.87	50.47	50.10	47.73	51.23
		Std	1.34	1.74	0.87	0.4	0.61	0.78	0	0.79
Colon	40	CER	**12.41**	**18.87**	**17.9**	19.35	24.03	29.03	33.87	33.87
		Std	2.52	4.32	2.79	2.28	2.79	5.64	13.21	0
Leukemia	40	CER	**4.58**	11.98	11.94	12.08	16.66	20.83	**6.94**	19.44

	Std	0.67	3.88	4.25	2.45	0	0	0	0
Average	CER	**18.99**	**18.93**	19.95	19.42	24.92	25.4	26.74	25.43
	Std	1.64	3.11	2.27	**1.45**	**1.4**	1.56	2.07	0.77

The classification error rates of the different methods on the RF classifier is shown in Table 9. The proposed method for wine, hepatitis, arrhythmia and, colon provides the best results by 2.8%, 18.62%, 33.62% and, 12.41% CER values, respectively.

Table 9. Average classification error rate over 10 runs of the unsupervised feature selection methods using RF classifier.

Datasets	# selected features		FSCBAS	GCACO	MGSACO	UFSACO	RSM	MC	RRFS	LS
Wine	6	CER	**2.8**	**3.89**	5.39	3.98	10.44	11.17	32.02	5.5
		Std	0	2.32	2.53	2.04	3.81	2.04	0	1.49
Hepatitis	6	CER	**18.61**	**18.7**	18.9	19.22	20.64	19.93	20.64	20.77
		Std	0	1.76	1.14	0.41	0	1.81	0	0.27
Ionosphere	15	CER	**14.23**	14.9	18.4	15.04	**11.99**	**11.99**	16.52	15.98
		Std	3.57	2.41	2.1	2.12	0.66	0.89	1.31	1.56
SpamBase	24	CER	14.22	10.98	18.14	10.04	11.75	10.71	**5.9**	**6.94**
		Std	0.23	1.98	2.68	0.54	1.26	0.94	0.24	0.21
Arrhythmia	20	CER	**33.62**	**38.82**	42.78	49.95	54.09	45.75	41.43	43.2
		Std	1.33	2.22	1.6	2.91	5.87	0.49	1.73	4.29
Madelon	40	CER	38.81	27.73	**16.71**	**16.76**	50.15	50.36	47.76	49.8
		Std	1.63	1.72	0.48	0.49	0.9	1.02	0	0
Colon	40	CER	**12.41**	18.78	**17.09**	18.22	23.87	21.29	35	30.64
		Std	2.53	3.59	2.18	4.93	5.31	4.42	13.26	0
Leukemia	40	CER	**4.58**	10.31	10.64	13.88	19.44	22.22	**1.38**	23.61
		Std	0.67	3.16	5.16	3.53	0	0	0	0
Average		CER	**17.41**	**18.01**	18.51	18.39	25.3	24.18	25.08	24.56
		Std	1.25	2.4	2.23	2.12	2.23	1.45	2.07	0.98

According to the findings, the proposed method often obtains either first rank or second rank with a lower difference in comparison with the first rank over all the classifiers on all the datasets. It can be concluded that the presented method achieved the best results on *high-dimensional datasets*. This claim was also illustrated in Fig. 7, which the proposed method depicted by blue color obtained lower CER than the other methods on all the classifiers.

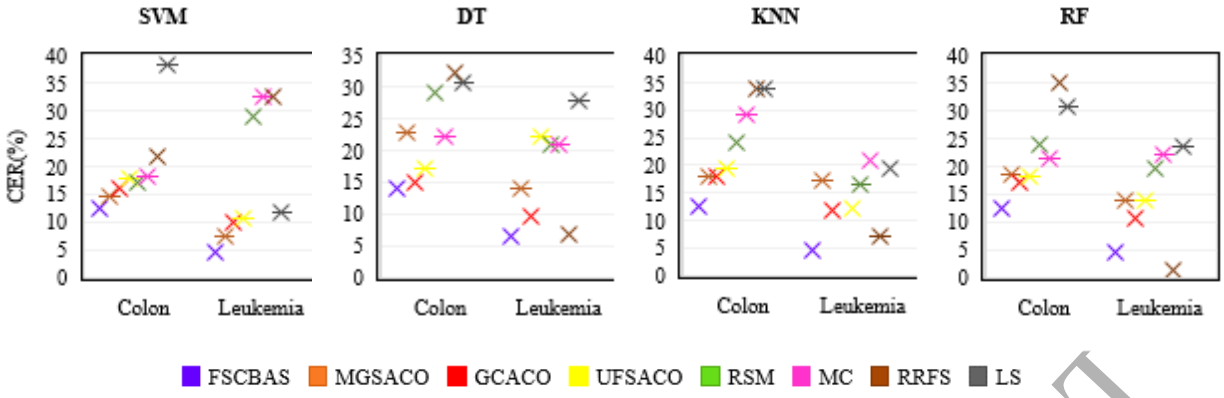


Fig. 7. Classification Error rate on two Colon and Leukemia datasets over all classifiers.

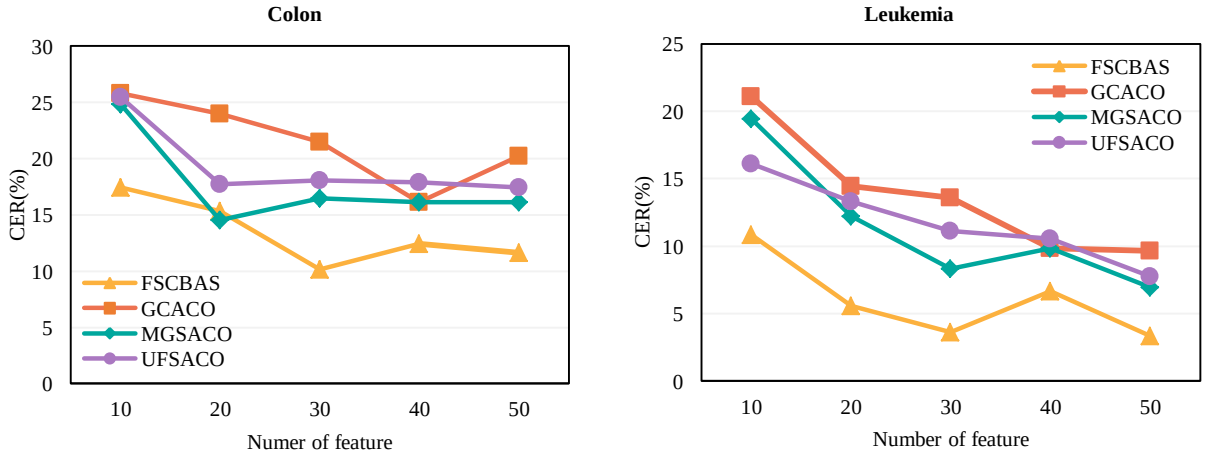
Since the accuracy does not provide enough information to evaluate the level of robustness of the obtained results, the proposed method should be evaluated with at least one other criterion. Precision, Recall, and F-measure are criterion for this purpose. Precision is the number of True Positives divided by the number of True Positives and False Positives. Recall is the number of True Positives divided by the number of True Positives and the number of False Negatives. The F1-Score is the $2 * ((\text{precision} * \text{recall})/(\text{precision} + \text{recall}))$. It is also called the F-Score or the F-Measure. Therefore, in this paper F-measure, Precision and Recall for all the methods on SVM classifier are reported in Table 10. It can be seen from the results, the proposed FSCBAS method obtained good results in comparison with the other methods. For example, for colon dataset, the proposed method by 86.39%, 86.28% values for F-measure and Recall obtained the best performance compared to the other feature selection methods. Also, the proposed method by 86.59% value for Precision is in the second-best place compared to others. In the average case, the proposed method obtained the best F-measure and Precision by 77.26% and 80.13% values over all datasets.

Table 10. Average other criterions over 10 runs of the unsupervised feature selection methods using SVM classifier.

Datasets	# selected features	(%)	FSCBAS	MGSACO	GCACO	UFSACO	RSM	MC	RRFS	LS
		F-measure	**97.10**	94.73	95.34	96.02	89.61	89.03	53.52	94.66
Wine	6	Recall	**97.2**	94.61	95.44	96	89.84	89.41	61.96	95.11
		Precision	**97.03**	94.88	95.28	96	89.5	88.81	47.99	94.43
		F-measure	**61.03**	53.94	**56.44**	53.37	44.24	54.73	44.24	47.98
Hepatitis	6	Recall	50	55.14	**56.88**	54.59	50	**57.03**	50	51.88
		Precision	78.33	73.07	**79.39**	78.28	39.67	54.31	39.67	47.09
		F-measure	**82.4**	77.43	79.85	82.26	**86.25**	**86.25**	80.45	80.97
Ionosphere	15	Recall	80.3	75.6	78.12	80.46	**84.67**	**84.69**	78.74	79.01
		Precision	**89.94**	85.04	85.02	86.95	**89.23**	89.22	85.36	86.61

		F-measure	**85.04**	79.28	80.33	80.76	65.02	70.09	**86.55**	80.7
SpamBase	24	Recall	**83.51**	78.02	78.89	79.37	65.52	69.60	**86.04**	80.06
		Precision	**86.62**	84.87	85.98	85.88	77.16	80.3	**87.29**	81.84
		F-measure	**59.1**	41	**52.4**	38.2	38.6	40.1	47.3	46.6
Arrhythmia	20	Recall	**67.5**	55.8	**63.7**	54.2	54	54.9	60	58.4
		Precision	**55.7**	35	46	29.4	40.6	50.6	43.7	45.2
		F-measure	52.24	**60.79**	**60.8**	60.74	50.48	49.69	54.67	49.03
Madelon	40	Recall	52.03	**60.79**	**60.8**	60.74	50.49	49.71	54.92	49.03
		Precision	52.45	**60.79**	**60.8**	60.74	50.49	49.71	55.03	49.03
		F-measure	**86.39**	81.29	82.43	78.56	82.78	70.22	47.25	**85.23**
Colon	40	Recall	**86.28**	80.03	81.45	77.12	82.27	69.47	54.65	**83.86**
		Precision	**86.59**	84.08	84.10	83.27	83.48	73.70	46.87	**87.62**
		F-measure	94.8	88.84	**96.37**	88.09	86.07	86.9	**96.77**	75
Leukemia	40	Recall	95.24	89.52	**96.96**	89.01	87.35	87.95	**95.65**	75.59
		Precision	94.37	88.37	**95.91**	87.45	85.2	86.06	**98.03**	74.55
Average		F-measure	**77.26**	72.16	**75.5**	72.25	67.88	68.38	63.84	70.02
		Recall	**76.51**	73.69	**76.53**	73.94	70.25	70.35	67.75	71.62
		Precision	**80.13**	75.76	79.06	76	69.42	71.59	62.99	70.80

Finally, to further evaluate the efficiency of the proposed algorithm, it has been assessed with the different number of selected features. The average CER on the number of selected feature for FSCBAS and others methods on the Colon and Leukemia datasets with SVM classifier are provided in Fig. 8. As can be seen from this figure, for Colon dataset, the FSCBAS has the lowest CER versus the other methods. Moreover, for Leukemia dataset, there is a big gap between the proposed method and the other feature selection methods.



(a) (b)

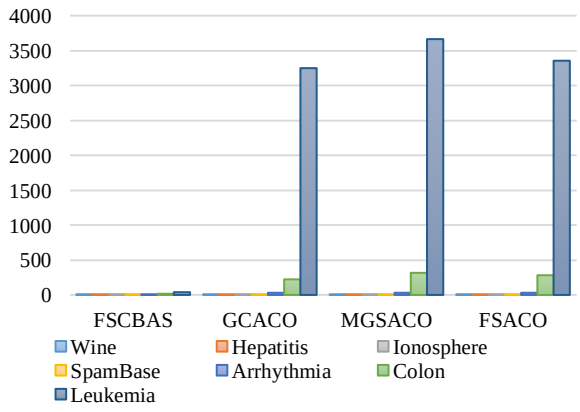
Fig. 8. Classification Error rate SVM on two Colon and Leukemia datasets in deferent features

5.5.2 Execution time

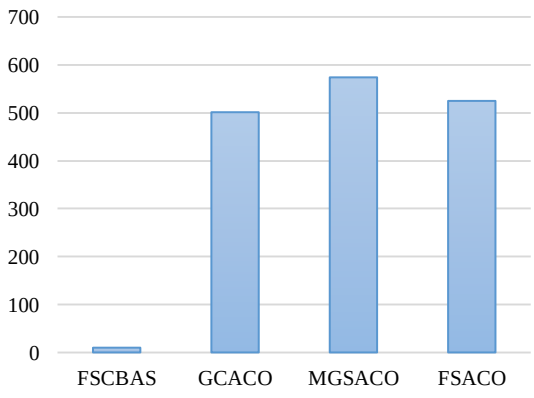
This section of experiments concerned assessment of the runtime of FSCBAS as compared to other ACO based algorithms over all the datasets. A comparison of the runtimes (in milliseconds) of the FSCBAS method with those of the other methods is provided in Table 11. Based on these results, the average execution time of the proposed method was about 5-8 times better than those of the other algorithms. This excellence is clearly illustrated in Fig. 9, clearly. The main strength of our approach is that it has high performance equal to or better than that of full-graph methods with very low computational complexity, which significantly reduces run time in high dimensional datasets.

Table 11. Execution time (millisecond)

Datasets	FSCBAS	GCACO	MGSACO	FSACO
Wine	125.9	35.8	36.2	36.4
Hepatitis	71.6	46.9	54.8	55.1
Ionosphere	464	640.2	616.6	688.8
SpamBase	3796.3	3712.0	3766.0	3753.9
Arrhythmia	8365.8	31158.4	31172.0	31172.0
Colon	17055.6	223378.6	316846.0	282260.9
Leukemia	39698.6	3252132.6	3668300.6	3358357.8
Average	**9939.6**	**501586.3571**	574398.9	525189.3



(a)



(b)

Fig. 9. (a) The execution time and (b) average executation time (in seconds) of the proposed method and the state of art ACO- based methods over all mentioned datasets.

5.5.3 Statistical test result

In this section, we analyze the results obtained from different feature selection methods using the Friedman test (Almuallim & Dietterich, 1991; Friedman, 1940). The Friedman test is a nonparametric statistical test for finding differences in behavior across multiple approaches. Being nonparametric means that the test does not assume the data coming from a particular distribution. The Friedman test can be used to evaluate the results of N different methods on K datasets using different classifiers. In this test, the methods are ranked based on their performance criterion. In this paper, the evaluation criterion was classification error rate; consequently, the method with the lowest rank would have the best performance. Table 12 shows the ranking of mentioned feature selection methods. As it can be seen, the proposed method had the highest rank on three classifiers (1.88, 2.75 and 2.50 for SVM, KNN and RF classifiers, correspondingly) and it is in the second place with 3.06 score on DT. According to Table 13, the P-values for the error rate values of the SVM, and RF classifiers is less than 0.05; thus, it has been found that these results are statistically significant.

Table 12. Ranking of the methods.

Classifier	FSCBAS	GCACO	MGSACO	UFSACO	RSM	MC	RRFS	LF
SVM	**1.88**	2.63	4.5	4.63	5.63	5.19	6.31	5.25
DT	**3.06**	**2.88**	4.25	5.38	6.19	4.69	5.06	3.06
KNN	**2.75**	3.13	4	4	5.69	5.44	5.06	5.94
RF	**2.5**	3.13	4.25	4	5.88	5.44	4.94	5.88

Table 13. The results of the Friedman test.

Classifier	N	Chi-Square	df	p-value
SVM	8	21.376	7	**0.003**
DT	8	11.681	7	0.112
KNN	8	13.622	7	0.058
RF	8	14.784	7	**0.039**

5.5.1 Comparison with a supervised filter based method

In this section, the proposed method is compared with a supervised filter based method called mRMR over the SVM and DT classifiers and the results are shown in Fig. 10. As can be seen from this figure, for Leukemia on the SVM classifier the classification error rate of the proposed method was close to that of mRMR when the number of features was 10 and for 20,30,40 and 50 features the proposed method was excellent as compared to that of mRMR. Furthermore, there is a considerable gap between the effectiveness of the proposed method and mRMR on SVM and DT classifiers. Therefore, it can be concluded that the proposed FSCBAS is superior to the mRMR method applied on the DT classifier for various number of features.

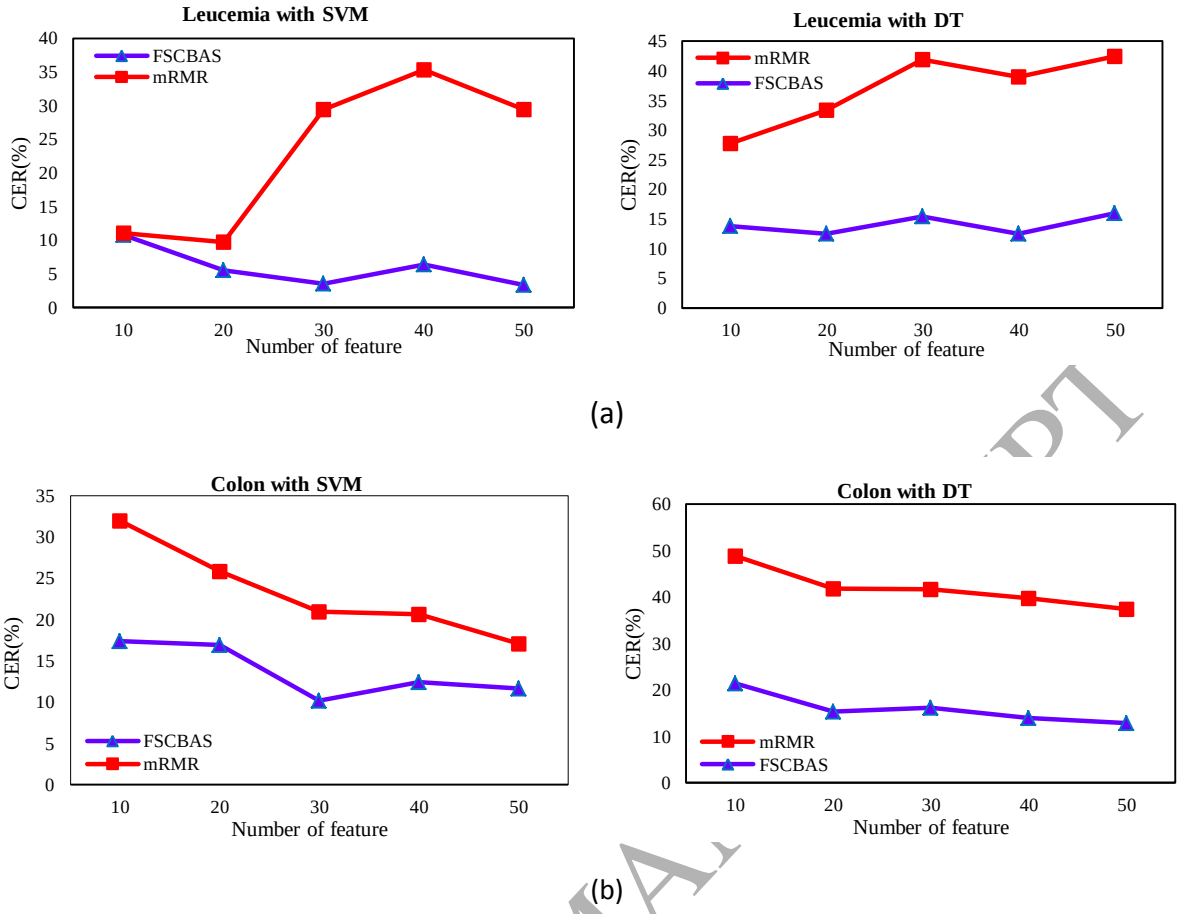


Fig. 10. Classification error rates of the proposed method and mRMR with respect to the number of selected features for SVM and DT classifiers on the (a) Leukemia, (b) Colon dataset.

5.5.2 An extensive comparison with state-of-art methods on high-dimensional data

In this section, the performance of the proposed method is compared with the results produced by the two other methods for two general purpose: 1- a novel maximum relevance and minimum common redundancy criterion and a minimax nonlinear optimization approach (N-MRMCR-MI) (Song et al., 2013), to exposes the effect of investigating the effect of redundancy reduction technique introduced by the proposed method , and, 2- a new group-based feature selection method using Markov blankets to search (PGVNS) (García-Torres et al., 2016), in purpose of inquiring the feature selection framework from the clusters and its effect on the performance of the proposed method.

1. N-MRMCR-MI seeks to maximize relevance and minimize redundancy under an optimization problem for nonlinear data. In Fig. 11, the results obtained from the proposed method in comparison with N-MRMCR-MI is shown. The selection of datasets and the number of features is set according to the N-MRMCR-MI method presented in (Song et al., 2013). As can be seen from the figure, the performance of the proposed method outperforms N-MRMCR-MI method on Colon, Lung cancer, and Leokemia datasets except to Inosphere. This experiment demonstrates that the proposed method was more successful in handling minimize the redundancy and maximize the relevancy.

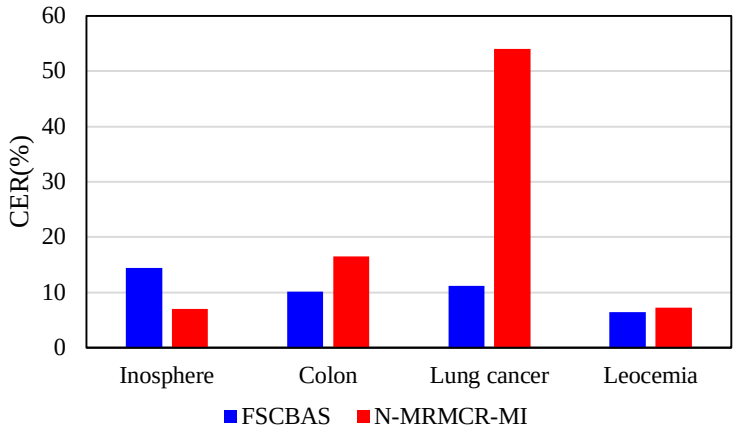


Fig. 11. Classification error rates of the proposed method and N-MRMCR-MI.

2. PGVNS is a feature selection method based on grouping features which focus on high dimensional data. Numerical results from the proposed method and PGVNS are illustrated in Figures 12-14. In order to provide a fair comparison, the number of selected features are set as same as the number of selected features by PGVNS method presented in (García-Torres et al., 2016). The datasets in this experiment include four microarray datasets *Colon*, *Lung Cancer*, *Prostate* and *SRBCT*, four text mining datasets include *Alt, Structure, Function* and *Disease*, and two other datasets include *Madellon* and *Arcence* with NIPS2003 challenge. The properties of these datasets such as the number of features and samples are shown in Table 14.

Table 14. Details on the real-world datasets

Type	Dataset	Features	Classes	Patterns
NIPS2003 challenge	Madellon	500	2	4400
	Arcene	10000	2	900
Microarray	Colon	2000	2	62
	SRBCT	2308	4	83
	Leukemia	7129	2	72
	Lung Cancer	12600	5	203
	Prostate	12600	2	102
Text mining	Alt	4157	2	2112
	Structure	3548	2	2368
	Function	3907	2	2708
	Disease	3237	2	2376

As shown in the Fig. 12, the proposed method has better performance on the challenging datasets

when the number of selected features is small. One challenge of the PGVNS method is that it automatically gives the number of features and its performance is not allowed to be increased to a greater degree. However,the performance can be increased by increasing or decreasing the number of selected features.

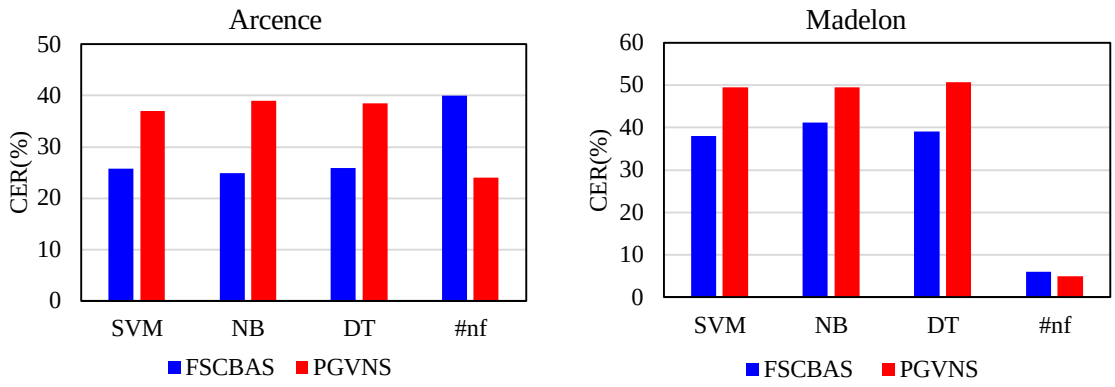


Fig. 12. Classification error rates of the proposed method and PGVNS on Arcence and Madelon.

Furthermore, the proposed method and PGVNS are compared on microarray datasets. On *Colon*, the presented method is markedly efficient, and on the other microarray datasets has relatively good performance (Fig. 13).

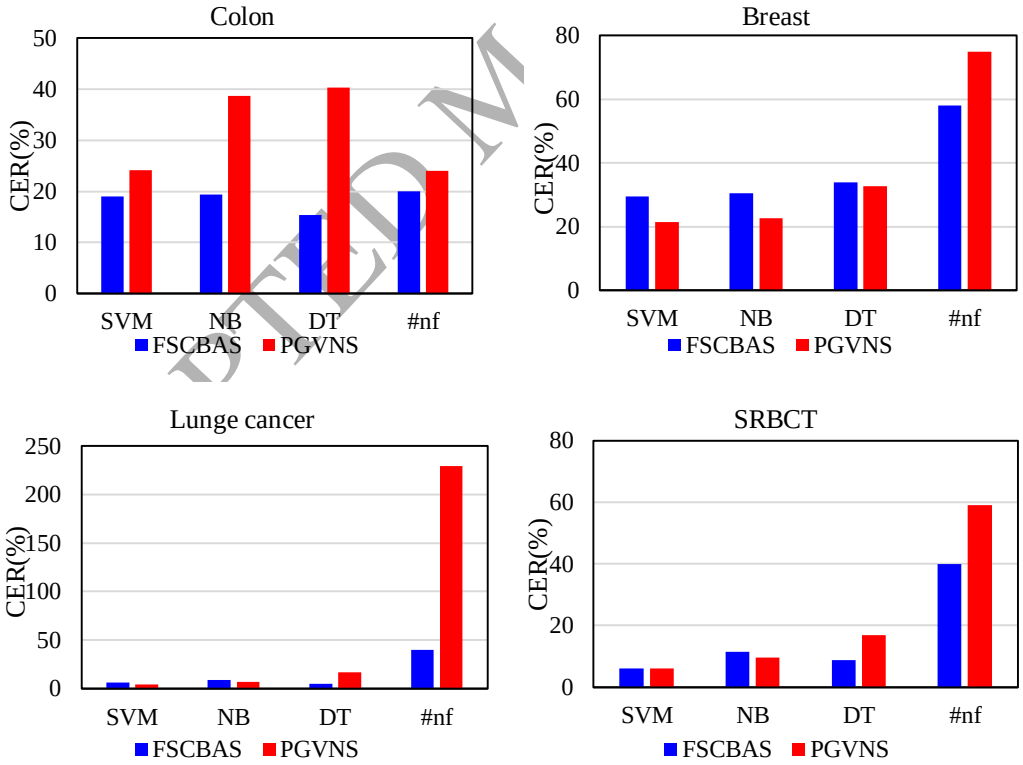


Fig. 13. Classification error rates of the proposed method and PGVNS on microarray datasets.

On the other hand, for further emphasize on the excellent performance of the proposed method on high dimensional data, PGVNS and the proposed method are compared on the text mining datasets.

It is clear from the Fig. 14 that on Alt dataset, the CER of the proposed FSCBAS is lower than the PGVNS algorithm, and on the other datasets FSCBAS often has better performance. It is worth mentioning that the results of the proposed method are achieved by selecting the small number of selected features.

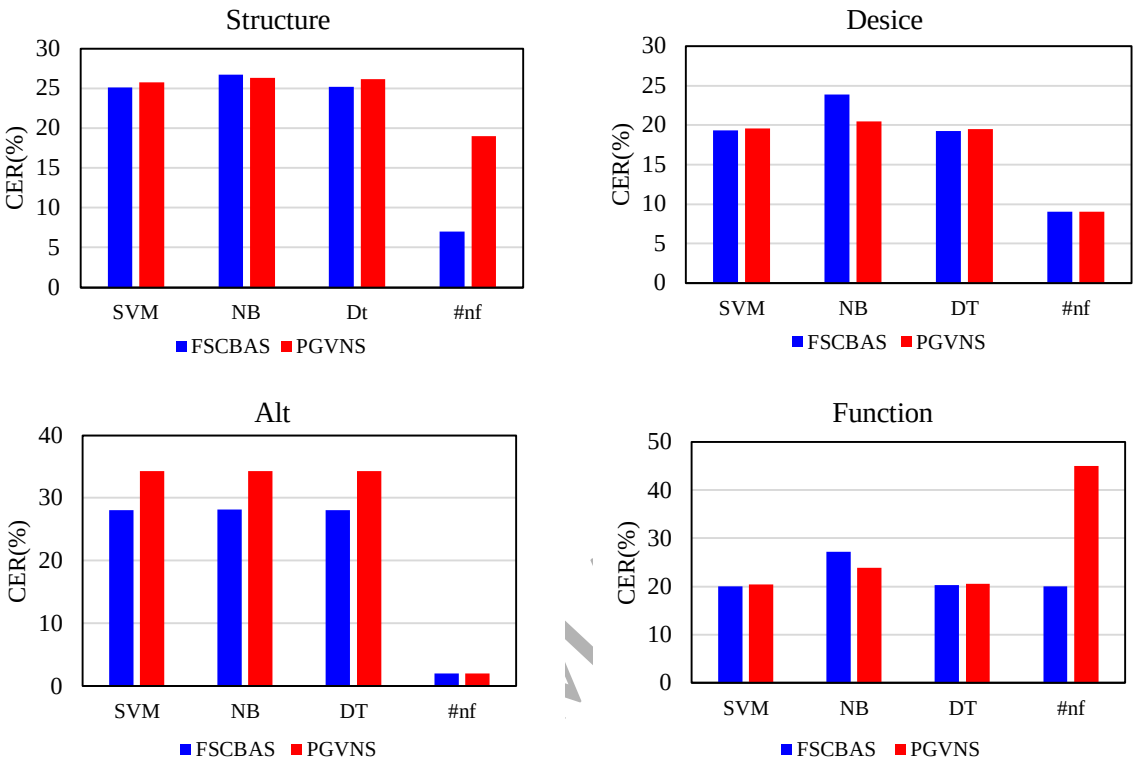


Fig. 14. Classification error rates of the proposed method and PGVNS on text mining datasets.

5.6 Discussion

The experiments are designed aim to two purposes in two logical sections. In the first section, the proposed method is generally analyzed in terms of accuracy, execution time, computational complexity, parameter setting, evaluation criteria and The Friedman test. Then, in order to approve the claim of being efficient of the proposed method on the high dimensional data, the proposed method is evaluated on the large datasets in the second section. In the first step, to evaluate and compare the performance of the proposed method with different algorithms, several experiments were performed. The results were obtained on each dataset for the proposed method and the well-known algorithms. In order to confirm the effectiveness of the presented algorithm, the different measures include CER, Recall, Precision and F-measure are utilized for assessing the presented algorithm and the other compared feature selection methods. The performance of FSCBAS was compared to the other methods with various classifiers, including SVM, DT, KNN, and RF. Among the comparative algorithms, LS is a univariate method which is capable to address only the irrelevant features by low performance. Other multivariate methods, including RRFS, RSM, MC, UFSACO, GCACO, and MGSACO can remove redundant and irrelevant features, efficiently. MC and RRFS which begin search space exploration from specific points, might fall into local optima.

On the other hand, since the search space representation of UFSACO and MGSACO methods as a full graph, the computational complexity of these methods is relatively high. Also in these methods, the redundancy among features is not considered efficiently. In order to overcome these challenges, the proposed method presents the search space as a circle graph and introduced an enhanced equation to reduce redundancy as much as possible. Furthermore, this method utilizes the benefits of clustering method with a new policy in contrast to other clustering-based methods, such as GCACO, which selects one feature from each cluster sequentially and has high computational complexity. Therefore, the proposed method converts the search space from full graph to circular graph and apply modified binary ACO, and also uses clustering advantages with an *incremental view by* definition of the competition role between the best features of all clusters

It can be seen from Tables 6-9 that the proposed method achieved good average performance on all of the datasets over various classifiers. The considerable effectiveness of the proposed method is shown in the Fig. 7 on two high dimensional data. Moreover, in order to prove the robustness of the proposed method in addition to accuracy, other measures including F-measure, Recall, and Precision are analyzed and their results reported in Table 10. The proposed method is compared with other methods for the various number of features on high dimensional datasets in Fig.8, which clearly shows the excellent performance of the presented method. Another preference of the proposed method is its low computational complexity and this claim is confirmed by investigating the numerical results of execution time which shown in Fig. 9 and Table 11. Furthermore, a brief description of the performance of the proposed method using the Friedman test can be seen in Tables 12 and 13 as well.

In the second step, firstly, the proposed method is compared with mRMR on two microarray datasets, and the results illustrated a big gap between the performance of the proposed method and mRMR in Fig. 10. Secondly, in order to prove the authors' claim to provide excellent performance on the high dimensional data, further experiments are performed and the proposed method is compared with two state-of-the-art algorithms. In term of investigating the selection of a feature subset with maximum relevancy and minimum redundancy, the proposed method is compared with N-MRMCR-MI, which the great performance of the proposed method is shown in Fig. 11. Moreover, for study the influence of clustering and the selection of features from clusters in the presented framework, it is compared with PGVNS on different large datasets, and the results are depicted in Figures 12-14.

The major advantage of the proposed method, by dividing and conquering the problem, is to drastically reduce complexity meanwhile improve the results, especially in high dimensional data. In all simulations, the proposed method on high dimensional datasets in terms of both *time* and *accuracy*, has excellent performance. These advantages make the presented method can be practiced in many expert system applications such as microarray data processing, text classification and image processing in high dimensional to handle the high dimensionality of the feature space and improve classification performance simultaneously.

6 Conclusion

Feature selection plays an important role in the classification task to reduce the computational cost, simplify the learning model and improve the general abilities of classifiers. In this paper, a novel

hybrid filter-based feature selection algorithm is presented. It is a combination of a linear binary ant system, clustering, damped mutation, and a new reduction redundancy policy. Each of them has special advantages. The linear binary ant system overcomes the search space challenge, clustering can reduce the challenge of processing high-dimensional datasets in some extent, by injecting mutation the search space can be more randomize, and redundancy is reduced by the new policy. In the proposed method, after clustering the features and placing them sequentially in a circular graph, the BASM algorithm was applied. Then, in an iterative process, some best features were selected until the desired subset features were completed. The presented model applied both of global and local search capabilities between and within clusters. In the proposed BASM inspired by genetic algorithms and simulated annealing, a damped mutation strategy was introduced to avoid falling into local optima. In addition, a new idea was applied to reduce the redundancy between selected features as much as possible. The performance of the proposed algorithm was compared to state-of-the-art feature selection algorithms using different classifiers on real-world datasets. The experimental results confirmed that FSCBAS reduces computational time significantly; and achieved better performance than other feature selection methods. The main advantage of our approach is having high performance in high dimensional data and low computational complexity which can be applied in the search-based platform to solve other optimization problems. These advantages cause the presented method to be practical in processing the big data.

There are several future directions for improving the proposed method.

- First, when the feature clustering is soft, the sensitivity of the proposed method to the number of clusters and feature selectivity is strongly reduced. Therefore, fuzzy clustering can be suitable.
- Second, when both general and specialized features are selected so that at first, a few general features are selected from the input space, and then the representative features are selected from each cluster. This policy will make the learning model more powerful. Because in this case at first the model learns with generic features and then with more specialized features. This idea is taken from "self-paced learning"(Kumar, Packer, & Koller, 2010).
- Third, an ensemble of feature selection algorithms can be used to provide a feature selection algorithm that is appropriate for different dataset and to support the "No free lunch theorem"(Wolpert & Macready, 1997). As a result, the proposed hybrid method can be developed toward an ensemble method.
- Fourth, a general way to improve the computation complexity and performance of the proposed method in confront with big data is to present a novel scheme of local feature selection in which the classification is performed by a parallel processing and data distribution. The main idea is gaining simultaneous benefits of Local Feature Selection and ACO search capability to improve results and reduce execution time.
- Finally, the proposed method can be extended for the multi-label applications.

References

Almuallim, H., & Dietterich, T. G. (1991). *E cient algorithms for identifying relevant features.* Paper presented at the Proc. of the 9th Canadian Conference on Artificial Intelligence.

Amiri, F., Yousefi, M. R., Lucas, C., Shakery, A., & Yazdani, N. (2011). Mutual information-based feature selection for intrusion detection systems. *Journal of Network and Computer Applications, 34*(4), 1184-1199.

Amoozegar, M., & Minaei-Bidgoli, B. (2018). Optimizing Multi-objective PSO based feature selection method using a feature elitism mechanism. *Expert Systems with Applications*.

Asuncion, A., & Newman, D. (2007). UCI repository of machine learning datasets. http://archive.ics.uci.edu/ml/datasets.html

Aynaud, T., & Guillaume, J. L. (2010, May 31 2010-June 4 2010). *Static community detection algorithms for evolving networks.* Paper presented at the 8th International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks.

Balabin, R. M., & Smirnov, S. V. (2011). Variable selection in near-infrared spectroscopy: benchmarking of feature selection methods on biodiesel data. *Analytica chimica acta, 692*(1-2), 63-72.

Benesty, J., Chen, J., Huang, Y., & Cohen, I. (2009). Pearson Correlation Coefficient *Noise Reduction in Speech Processing* (pp. 1-4). Berlin, Heidelberg: Springer Berlin Heidelberg.

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

Brkić, K. (2013). *Structural analysis of video by histogram-based description of local space-time appearance.* Fakultet elektrotehnike i računarstva, Sveučilište u Zagrebu.

Cai, J., Luo, J., Wang, S., & Yang, S. (2018). Feature selection in machine learning: A new perspective. *Neurocomputing, 300*, 70-79.

Che, J., Yang, Y., Li, L., Bai, X., Zhang, S., & Deng, C. (2017). Maximum relevance minimum common redundancy feature selection for nonlinear data. *Information Sciences, 409*, 68-86.

Chen, B., Chen, L., & Chen, Y. (2013). Efficient ant colony optimization for image feature selection. *Signal processing, 93*(6), 1566-1576.

Chen, Y., Miao, D., & Wang, R. (2010). A rough set approach to feature selection based on ant colony optimization. *Pattern Recognition Letters, 31*(3), 226-233.

Chen, Y., Miao, D., Wang, R., & Wu, K. (2011). A rough set approach to feature selection based on power set tree. *Knowledge-Based Systems, 24*(2), 275-281.

Chuang, L.-Y., Tsai, S.-W., & Yang, C.-H. (2011). Improved binary particle swarm optimization using catfish effect for feature selection. *Expert Systems with Applications, 38*(10), 12699-12707.

Chuang, L.-Y., Yang, C.-H., & Li, J.-C. (2011). Chaotic maps based on binary particle swarm optimization for feature selection. *Applied Soft Computing, 11*(1), 239-248.

Das, A. K., Goswami, S., Chakrabarti, A., & Chakraborty, B. (2017). A new hybrid feature selection approach using feature association map for supervised and unsupervised classification. *Expert Systems with Applications, 88*, 81-94.

Dessì, N., Pascariello, E., & Pes, B. (2013). A comparative analysis of biomarker selection techniques. *BioMed research international, 2013*.

Dettling, M., & Bühlmann, P. (2004). Finding predictive gene groups from microarray data. *Journal of Multivariate Analysis, 90*(1), 106-131.

Dhillon, I. S., Mallela, S., & Kumar, R. (2003). A divisive information-theoretic feature clustering algorithm for text classification. *Journal of Machine Learning Research, 3*(Mar), 1265-1287.

Dorigo, M., & Gambardella, L. M. (1997). Ant colony system: a cooperative learning approach to the traveling salesman problem. *IEEE Transactions on evolutionary computation, 1*(1), 53-66.

Ferreira, A. J., & Figueiredo, M. A. (2012). Efficient feature selection filters for high-dimensional data. *Pattern Recognition Letters, 33*(13), 1794-1804.

Ferreira, A. J., & Figueiredo, M. A. T. (2012). An unsupervised approach to feature discretization and selection. *Pattern recognition, 45*(9), 3048-3060. doi:http://dx.doi.org/10.1016/j.patcog.2011.12.008

Friedman, M. (1940). A comparison of alternative tests of significance for the problem of m rankings. *The Annals of Mathematical Statistics, 11*(1), 86-92.

Gangeh, M. J., Zarkoob, H., & Ghodsi, A. (2017). Fast and Scalable Feature Selection for Gene Expression Data Using Hilbert-Schmidt Independence Criterion. *IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB), 14*(1), 167-181.

García-Torres, M., Gómez-Vela, F., Melián-Batista, B., & Moreno-Vega, J. M. (2016). High-dimensional feature selection via feature grouping: A Variable Neighborhood Search approach. *Information Sciences, 326*, 102-118.

Ghaemi, M., & Feizi-Derakhshi, M.-R. (2016). Feature selection using forest optimization algorithm. *Pattern recognition, 60*, 121-129.

Ghareb, A. S., Bakar, A. A., & Hamdan, A. R. (2016). Hybrid feature selection based on enhanced genetic algorithm for text categorization. *Expert Systems with Applications, 49*, 31-47.

Ghazavi, S. N., & Liao, T. W. (2008). Medical data mining by fuzzy modeling with selected features. *Artificial Intelligence in Medicine, 43*(3), 195-206.

Gheyas, I. A., & Smith, L. S. (2010). Feature subset selection in large dimensionality domains. *Pattern Recognition, 43*(1), 5-13. doi:http://dx.doi.org/10.1016/j.patcog.2009.06.009

Ghosh, M., Begum, S., Sarkar, R., Chakraborty, D., & Maulik, U. (2018). Recursive Memetic Algorithm for Gene Selection in Microarray Data. *Expert Systems with Applications*.

Haindl, M., Somol, P., Ververidis, D., & Kotropoulos, C. (2006). Feature selection based on mutual correlation. *Progress in pattern recognition, image analysis and applications*, 569-577.

Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., & Witten, I. H. (2009). The WEKA Data Mining Software: An Update. *SIGKDD Explorations, 11*(1).

He, X., Cai, D., & Niyogi, P. (2005a). Laplacian Score for Feature Selection. *Advances in Neural Information Processing Systems, 18*.

He, X., Cai, D., & Niyogi, P. (2005b). *Laplacian score for feature selection.* Paper presented at the NIPS.

Herman, G., Zhang, B., Wang, Y., Ye, G., & Chen, F. (2013). Mutual information-based method for selecting informative feature sets. *Pattern recognition, 46*(12), 3315-3327.

Hu, L., Gao, W., Zhao, K., Zhang, P., & Wang, F. (2018). Feature selection considering two types of feature relevancy and feature interdependency. *Expert Systems with Applications, 93*, 423-434.

Hu, Z., Bao, Y., Xiong, T., & Chiong, R. (2015). Hybrid filter–wrapper feature selection for short-term load forecasting. *Engineering Applications of Artificial Intelligence, 40*, 17-27.

Huang, C.-L., & Huang, W.-L. (2009). Handling sequential pattern decay: Developing a two-stage collaborative recommender system. *Electronic Commerce Research and Applications, 8*(3), 117-129.

Ienco, D., & Meo, R. (2008). *Exploration and reduction of the feature space by hierarchical clustering.* Paper presented at the Proceedings of the 2008 SIAM International Conference on Data Mining.

Jang, S.-H., Roh, J.-H., Kim, W., Sherpa, T., Kim, J.-H., & Park, J.-B. (2011). A novel binary ant colony optimization: Application to the unit commitment problem of power systems. *Journal of Electrical Engineering and Technology, 6*(2), 174-181.

Jenatton, R., Audibert, J.-Y., & Bach, F. (2011). Structured variable selection with sparsity-inducing norms. *Journal of Machine Learning Research, 12*(Oct), 2777-2824.

Jiang, S., Chin, K.-S., Wang, L., Qu, G., & Tsui, K. L. (2017). Modified genetic algorithm-based feature selection combined with pre-trained deep neural network for demand forecasting in outpatient department. *Expert Systems with Applications, 82*, 216-230.

Jiang, Y., & Ren, J. (2011). *Eigenvalue sensitive feature selection.* Paper presented at the Proceedings of the 28th International Conference on Machine Learning (ICML-11).

Jörnsten, R., & Yu, B. (2003). Simultaneous gene clustering and subset selection for sample classification via MDL. *bioinformatics, 19*(9), 1100-1109.

Kabir, M. M., Shahjahan, M., & Murase, K. (2011). A new local search based hybrid genetic algorithm for feature selection. *Neurocomputing, 74*(17), 2914-2928.

Kadri, O., Mouss, L. H., & Mouss, M. D. (2012). Fault diagnosis of rotary kiln using SVM and binary ACO. *Journal of mechanical science and technology, 26*(2), 601-608.

Kashef, S., & Nezamabadi-pour, H. (2015). An advanced ACO algorithm for feature subset selection. *Neurocomputing, 147*, 271-279.

Kim, S., & Xing, E. P. (2012). Feature selection via block-regularized regression. *arXiv preprint arXiv:1206.3268*.

Kim, Y., & Kim, J. (2004). *Gradient LASSO for feature selection.* Paper presented at the Proceedings of the twenty-first international conference on Machine learning.

Kong, M., & Tian, P. (2005). A binary ant colony optimization for the unconstrained function optimization problem. *Computational intelligence and security*, 682-687.

Kong, M., & Tian, P. (2006). *Introducing a binary ant colony optimization.* Paper presented at the International Workshop on Ant Colony Optimization and Swarm Intelligence.

Krier, C., François, D., Rossi, F., & Verleysen, M. (2007). *Feature clustering and mutual information for the selection of variables in spectral data.* Paper presented at the ESANN.

Kumar, M. P., Packer, B., & Koller, D. (2010). *Self-paced learning for latent variable models.* Paper presented at the Advances in Neural Information Processing Systems.

Lai, C., Reinders, M. J. T., & Wessels, L. (2006). Random subspace method for multivariate feature selection. *Pattern Recognition Letters, 27*(10), 1067-1076. doi:http://dx.doi.org/10.1016/j.patrec.2005.12.018

Latkowski, T., & Osowski, S. (2015). Data mining for feature selection in gene expression autism data. *Expert Systems with Applications, 42*(2), 864-872.

Lin, C.-H., Chen, H.-Y., & Wu, Y.-S. (2014). Study of image retrieval and classification based on adaptive features using genetic algorithm feature selection. *Expert Systems with Applications, 41*(15), 6611-6621.

Liu, H., & Motoda, H. (2012). *Feature selection for knowledge discovery and data mining* (Vol. 454): Springer Science & Business Media.

Liu, H., Wu, X., & Zhang, S. (2011). *Feature selection using hierarchical feature clustering.* Paper presented at the Proceedings of the 20th ACM international conference on Information and knowledge management.

Liu, H., & Yu, L. (2005). Toward integrating feature selection algorithms for classification and clustering. *IEEE Transactions on knowledge and data engineering, 17*(4), 491-502.

López, F. G., Torres, M. G., Batista, B. M., Pérez, J. A. M., & Moreno-Vega, J. M. (2006). Solving feature subset selection problem by a parallel scatter search. *European Journal of Operational Research, 169*(2), 477-489.

Loscalzo, S., Yu, L., & Ding, C. (2009). *Consensus group stable feature selection.* Paper presented at the Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining.

Ma, Z., & Tavares, J. M. R. (2017). Effective features to classify skin lesions in dermoscopic images. *Expert Systems with Applications, 84*, 92-101.

Martens, D., Baesens, B., & Fawcett, T. (2011). Editorial survey: swarm intelligence for data mining. *Machine Learning, 82*(1), 1-42.

Martínez Sotoca, J., & Pla, F. (2010). Supervised feature selection by clustering using conditional mutual information-based distances. *Pattern recognition, 43*(6), 2068-2081. doi:http://dx.doi.org/10.1016/j.patcog.2009.12.013

Mohamed, N. S., Zainudin, S., & Othman, Z. A. (2017). Metaheuristic approach for an enhanced mRMR filter method for classification using drug response microarray data. *Expert Systems with Applications, 90*, 224-231.

Moradi, P., & Rostami, M. (2015). Integration of graph clustering with ant colony optimization for feature selection. *Knowledge-Based Systems, 84*, 144-161. doi:http://dx.doi.org/10.1016/j.knosys.2015.04.007

Muštra, M., Grgić, M., & Delač, K. (2012). Breast density classification using multiple feature selection. *automatika, 53*(4), 362-372.

Narendra, P. M., & Fukunaga, K. (1977). A branch and bound algorithm for feature subset selection. *IEEE Transactions on computers*(9), 917-922.

Novovicová, J., Pudil, P., & Kittler, J. (1996). Divergence based feature selection for multimodal class densities. *IEEE Transactions on pattern analysis and machine intelligence, 18*(2), 218-223.

Oh, I.-S., Lee, J.-S., & Moon, B.-R. (2004). Hybrid genetic algorithms for feature selection. *IEEE Transactions on pattern analysis and machine intelligence, 26*(11), 1424-1437.

Pacheco, F., Cerrada, M., Sánchez, R.-V., Cabrera, D., Li, C., & de Oliveira, J. V. (2017). Attribute clustering using rough set theory for feature selection in fault severity classification of rotating machinery. *Expert Systems with Applications, 71*, 69-86.

Padungweang, P., Lursinsap, C., & Sunat, K. (2012). A discrimination analysis for unsupervised feature selection via optic diffraction principle. *IEEE transactions on neural networks and learning systems, 23*(10), 1587-1600.

Peng, H., Long, F., & Ding, C. (2005). Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy. *IEEE Transactions on pattern analysis and machine intelligence, 27*(8), 1226-1238.

Peng, Y., Li, W., & Liu, Y. (2006). A hybrid approach for biomarker discovery from microarray gene expression data for cancer classification. *Cancer informatics, 2*, 117693510600200024.

Saeys, Y., Inza, I., & Larrañaga, P. (2007). A review of feature selection techniques in bioinformatics. *bioinformatics, 23*(19), 2507-2517.

Shen, X., & Huang, H.-C. (2010). Grouping pursuit through a regularization solution surface. *Journal of the American Statistical Association, 105*(490), 727-739.

Shi, L., Wan, Y., Gao, X., & Wang, M. (2018). Feature Selection for Object-Based Classification of High-Resolution Remote Sensing Images Based on the Combination of a Genetic Algorithm and Tabu Search. *Computational intelligence and neuroscience, 2018*.

Silva, B., & Marques, N. C. (2010). *Feature Clustering with Self-organizing Maps and an Application to Financial Time-series for Portfolio Selection.* Paper presented at the IJCCI (ICFC-ICNC).

Somol, P., Pudil, P., & Kittler, J. (2004). Fast branch & bound algorithms for optimal feature selection. *IEEE Transactions on Pattern Analysis & Machine Intelligence*(7), 900-912.

Song, Q., Ni, J., & Wang, G. (2013). A fast clustering-based feature subset selection algorithm for high-dimensional data. *IEEE Transactions on knowledge and data engineering, 25*(1), 1-14.

Sotoca, J. M., & Pla, F. (2010). Supervised feature selection by clustering using conditional mutual information-based distances. *Pattern recognition, 43*(6), 2068-2081.

Tabakhi, S., & Moradi, P. (2015). Relevance–redundancy feature selection based on ant colony optimization. *Pattern recognition, 48*(9), 2798-2811.

Tabakhi, S., Moradi, P., & Akhlaghian, F. (2014). An unsupervised feature selection algorithm based on ant colony optimization. *Engineering Applications of Artificial Intelligence, 32*, 112-123.

Tabakhi, S., Najafi, A., Ranjbar, R., & Moradi, P. (2015). Gene selection for microarray data classification using a novel ant colony optimization. *Neurocomputing, 168*, 1024-1036.

Tibshirani, R., Saunders, M., Rosset, S., Zhu, J., & Knight, K. (2005). Sparsity and smoothness via the fused lasso. *Journal of the Royal Statistical Society: Series B (Statistical Methodology), 67*(1), 91-108.

Tran, B., Xue, B., & Zhang, M. (2018). A new representation in PSO for discretization-based feature selection. *IEEE transactions on cybernetics, 48*(6), 1733-1746.

Unler, A., Murat, A., & Chinnam, R. B. (2011). mr2PSO: A maximum relevance minimum redundancy feature selection method based on swarm intelligence for support vector machine classification. *Information Sciences, 181*(20), 4625-4641. doi:http://dx.doi.org/10.1016/j.ins.2010.05.037

Wan, Y., Wang, M., Ye, Z., & Lai, X. (2016). A feature selection method based on modified binary coded ant colony optimization algorithm. *Applied Soft Computing, 49*, 248-258.

Wang, L., Chu, F., & Xie, W. (2007). Accurate cancer classification using expressions of very few genes. *IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB), 4*(1), 40-53.

Wei, J., Zhang, R., Yu, Z., Hu, R., Tang, J., Gui, C., & Yuan, Y. (2017). A BPSO-SVM Algorithm based on Memory Renewal and Enhanced Mutation Mechanisms for Feature Selection. *Applied Soft Computing*.

Witten, D. M., & Tibshirani, R. (2010). A framework for feature selection in clustering. *Journal of the American Statistical Association, 105*(490), 713-726.

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

Xiang, S., Nie, F., Meng, G., Pan, C., & Zhang, C. (2012). Discriminative least squares regression for multiclass classification and feature selection. *IEEE transactions on neural networks and learning systems, 23*(11), 1738-1754.

Xu, Z., King, I., Lyu, M. R.-T., & Jin, R. (2010). Discriminative semi-supervised feature selection via manifold regularization. *IEEE transactions on neural networks, 21*(7), 1033-1047.

Yang, J.-B., & Ong, C.-J. (2011). Feature selection using probabilistic prediction of support vector regression. *IEEE transactions on neural networks, 22*(6), 954-962.

Yang, Y., Shen, H. T., Ma, Z., Huang, Z., & Zhou, X. (2011). *l2, 1-norm regularized discriminative feature selection for unsupervised learning.* Paper presented at the IJCAI proceedings-international joint conference on artificial intelligence.

Yu, B., & Yuan, B. (1993). A more efficient branch and bound algorithm for feature selection. *Pattern recognition, 26*(6), 883-889.

Yu, L., Ding, C., & Loscalzo, S. (2008). *Stable feature selection via dense feature groups.* Paper presented at the Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining.

Yu, L., & Liu, H. (2004). Efficient feature selection via analysis of relevance and redundancy. *Journal of Machine Learning Research, 5*(Oct), 1205-1224.

Zeng, Z., Wang, X., Zhang, J., & Wu, Q. (2016). Semi-supervised feature selection based on local discriminative information. *Neurocomputing, 173*, 102-109.

Zhang, H., & Sun, G. (2002). Feature selection using tabu search method. *Pattern recognition, 35*(3), 701-711.

Zhang, Y., Li, S., Wang, T., & Zhang, Z. (2013). Divergence-based feature selection for separate classes. *Neurocomputing, 101*, 32-42.

Zhang, Y., Li, X., Gao, L., & Li, P. (2018). A new subset based deep feature learning method for intelligent fault diagnosis of bearing. *Expert Systems with Applications*.

Zhang, Y., Yang, A., Xiong, C., Wang, T., & Zhang, Z. (2014). Feature selection using data envelopment analysis. *Knowledge-Based Systems, 64*, 70-80.

Zhang, Y., & Zhang, Z. (2012). Feature subset selection with cumulate conditional mutual information minimization. *Expert Systems with Applications, 39*(5), 6078-6088.

Zhao, J., Lu, K., & He, X. (2008). Locality sensitive semi-supervised feature selection. *Neurocomputing, 71*(10), 1842-1849.

Zhao, X., Deng, W., & Shi, Y. (2013). Feature selection with attributes clustering by maximal information coefficient. *Procedia Computer Science, 17*, 70-79.

Zhao, Z., & Liu, H. (2007). *Spectral feature selection for supervised and unsupervised learning.* Paper presented at the Proceedings of the 24th international conference on Machine learning.

Zhao, Z., Wang, L., Liu, H., & Ye, J. (2013). On similarity preserving feature selection. *IEEE Transactions on knowledge and data engineering, 25*(3), 619-632.

Zhou, J.-H., Pang, C. K., Lewis, F. L., & Zhong, Z.-W. (2011). Dominant feature identification for industrial fault detection and isolation applications. *Expert Systems with Applications, 38*(8), 10676-10684.

Zorarpacı, E., & Özel, S. A. (2016). A hybrid approach of differential evolution and artificial bee colony for feature selection. *Expert Systems with Applications, 62*, 91-103.