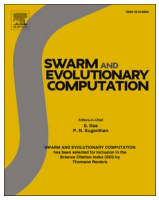
Swarm and Evolutionary Computation 75 (2022) 101165

Contents lists available at ScienceDirect

Swarm and Evolutionary Computation

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



An evolutionary correlation-aware feature selection method for classification problems

Motahare Namakin [a], Modjtaba Rouhani [a,*], Mostafa Sabzekar [b]

[a] Department of Computer Engineering, Ferdowsi University of Mashhad, Mashhad, Iran
[b] Department of Computer Engineering, Birjand University of Technology, Birjand, Iran

ARTICLE INFO

Keywords:
Feature selection
Correlated features
Estimation of distribution algorithms
Conditional probabilities

ABSTRACT

As global search techniques, population-based optimization algorithms have provided promising results in feature selection (FS) problems. However, their major challenge is high time complexity associated with the exploration of a large search space and consequently a large number of fitness function evaluations. Moreover, the interaction between features is another key issue in FS problems, directly affecting the classification performance through selecting correlated features. In this paper, an estimation of distribution algorithm (EDA)-based method is proposed with three important contributions. Firstly, as an extension of EDA, the proposed method in each iteration generates only two individuals competing based on a fitness function, evolving during the algorithm using our proposed update procedure. Secondly, we provide a guiding technique to determine the number of features to be selected for individuals in each iteration. As a result, the number of selected features in the final solution would be optimized during the evolution process. These two would lead to increasing the convergence speed of the algorithm. Thirdly, as the main contribution of the paper, in addition to considering the importance of each feature alone, the proposed method can consider the interaction between features, being able to deal with complementary features and consequently increase classification performance. To do this, we provide a conditional probability scheme that considers the joint probability distribution of selecting two features. The introduced probabilities successfully detect correlated features. Experimental results on a synthetic dataset with correlated features proved the performance of our proposed approach facing these types of features. Furthermore, the results on 13 real-world datasets obtained from the UCI repository showed the superiority of the proposed method in comparison with some state-of-the-art approaches. To evaluate the effectiveness of each feature subset, support vector machines are used as classifier. The efficiency analysis of the experimental results using two non-parametric statistical tests proved that the proposed method had significant advantages in comparison to other approaches.

1. Introduction

Classification as one of the major tasks in machine learning has been remarkably applied to a wide range of research topics such as Bioinformatics [1], intrusion detection systems (IDSs) [2], fraud detection [3], and prediction of different diseases (Parkinson [4], Cancers [5], COVID-19 [6], etc.). In classification tasks, feature selection (FS) is one of the critical preprocessing steps. This step, essential to improve the classification performance and to build a robust model, selects the optimal subset of features such that the selected features be as informative and small as possible. Thus, FS can lead to reducing the computational time and cost of building the model, preventing over-fitting, which in turn could increase the generalizability of the obtained classifier, and improving its accuracy by removing redundant and irrelevant features. However, without prior knowledge, it is difficult to discriminate between salient and common features. Moreover, FS is a challenging research problem not only due to the large search space, but also owning to the correlation among features. While the first one, which stems exponentially from the large number of features, would make the FS an NP-hard problem and consequently could make the exhaustive search an impractical solution, the latter can considerably decrease the classification accuracy.

There have been a large body of literature that aims to overcome those FS challenges. Based on the evaluation criteria, these efforts can be

* Corresponding author.
E-mail address: rouhani@um.ac.ir (M. Rouhani).

https://doi.org/10.1016/j.swevo.2022.101165
Received 10 October 2021; Received in revised form 20 July 2022; Accepted 20 August 2022
Available online 21 August 2022

generally classified into: *filter, wrapper*, and *embedded* approaches [7]. To begin with, filter feature selection methods are based merely on the inherent characteristics of data, generating candidate feature subsets without involving any classification algorithm in evaluating phase. Moreover, the wrapper approaches utilize a predetermined learning algorithm and consider their performance as the goodness criterion of feature subsets, generally leading to better performance than that of filter-based methods. However, these algorithms are computationally more expensive and need more time to execute compared to filter-based algorithms. Finally, embedded approaches, incorporating feature selection and training process of the learning model into a single procedure, combine the advantages of two other categories in order to make a good trade-off between computational time and accuracy, although applying just to specific learning models is the main limitation of this category.

Moreover, from the viewpoint of the searching techniques, the FS methods can be categorized into sequential and global search methods [8]. The most common methods in the former category are the sequential forward selection (SFS) and the sequential backward selection (SBS). The SFS (SBS) starts with the empty (full) set and progressively adds (removes) features until the performance of the classifier is improved (not decreased). However, they are typically stuck in local optima. On the other side, the search strategy of the latter category is based on the random search in the solution space to find the best feature subset. Nowadays, global search methods, as other population-based optimization approaches, report promising and successful results dealing with the FS problem [9]. The most common population-based optimization methods in solving the FS problem include genetic algorithm (GA) [10–12], particle swarm optimization (PSO) [13–16], ant colony optimization (ACO) [17,18,19], artificial bee colony (ABC) [20–22], and cuckoo search (CS) [23,24]. The main advantages of population-based optimization approaches are, firstly, they do not need domain knowledge or any assumption about the problem, and secondly, they can generate several solutions in a single run due to their population-based structure [8]. However, their main challenge is that they suffer from high time complexity because of the exploration of a large search space and consequently a large number of function evaluations. Apart from challenges mentioned, another limitation related to all FS methods is that they usually cannot consider the interaction between features. On the one hand, a feature may be weakly relevant to the class label individually although the classification accuracy can be improved by combining it with some complementary features. On the other hand, a feature may be relevant by itself, but it may decrease the classification accuracy or generalization when used with some other correlated features.

To overcome the limitations of feature selection methods utilizing population-based optimization, our main contribution is to present a new method based on the estimation of distribution algorithm (EDA) considering the correlated features. EDA [25,26] is one of the population-based optimization methods that generates the new candidate solutions by a probabilistic model. The optimization in this method is considered as a sequence of probabilistic model updates. With the new candidate solutions being generated by utilizing an implicit distribution (variation operators) in conventional population-based optimization methods, the EDA utilizes an explicit probability distribution such as multivariate interaction, Bayesian network, and etc.

This paper attempts to present a novel correlation-aware feature selection approach so that both the importance of each lone feature and the value of the interaction between features are considered. Thus, the proposed method would be able to deal with complementary features and consequently would improve the classification performance. For this purpose, we introduce a conditional probability scheme considering the joint probability distribution of selecting two features. The interaction between features is also considered by introducing the *interaction matrix* (IM). As will be discussed in Section 3.4, this matrix reflects the interaction between each pair of features. In fact, its elements represent the probabilities of mutually selecting each pair. Furthermore, the proposed method generates only two individuals in each iteration competing based on a fitness function and evolve during the algorithm execution. Moreover, the proposed approach would be quite fast in solving FS problems due to considering the interaction between features using *IM*. Finally, we propose a guiding technique that helps the algorithm select the appropriate number of features during the evolution process.

The rest of this paper has been organized in the following way. Section 2 provides a brief review of the state-of-the-art FS methods, which utilized population-based optimization algorithms. The architecture of the proposed method is illustrated in Section 3. Finally, the experimental results followed by some conclusions and areas for future research are discussed in Sections 4 and 5, respectively.

2. Literature review

As mentioned before, the population-based optimization approaches have been widely used to solve the FS problem in the literature. To categorize and study these efforts, there exist different ways one of which is employed here. We study them in light of two aspects which are the representation method and the number of objectives. Based on the representation method, the FS algorithms that utilized population-based optimization can be classified into binary and continuous representations. In binary representation, 1 s and 0 s are used as values of each individuals' vector element, indicating selecting or non-selecting features, the continuous representation consists of real values. Generally, in continuous representation, a threshold θ is considered to determine whether the corresponding feature should be selected or discarded. The element value relating to a feature will be selected by the FS method If it is greater than θ, otherwise it will be dropped.

Besides, based on the number of objectives, they can be categorized into single-objective (SO) and multi-objective (MO) algorithms. Whilst the SO methods usually consider only the classification accuracy as the objective function, other criteria are also accompanied in MO methods to evaluate the FS model. For instance, in many MO methods, the number of selected features is considered as the second objective function in addition to the primary accuracy of the classification algorithm. Although there appear to be many approaches in each category that can be reviewed, the remaining part of this section surveys only the state-of-the-art studies and compares their capabilities.

Genetic algorithm is the most common and likely the first population-based optimization method that has been adopted for the FS problem and has been applied in many studies [27,28]. GA finds an optimal solution using applying evolutionary operators such as crossover, mutation, and selection to the population. The authors in [28] firstly ranked features according to a filter criterion and then applied GA on the high-rank features in an attempt to reduce the search space. Moslehi and Haeri [29] used the GA along with the PSO to achieve better performance. Having integrated the populations obtained using these algorithms, the best solutions from the integrated population were selected. In [30], a bi-objective GA is used for an ensemble-based feature selection technique, and the boundary region analysis alongside the multivariate mutual information were considered as objective functions to select informative features.

Another population-based optimization method that has been widely used to solve FS problems in the literature is PSO. It was developed by Eberhart and Kennedy [31] in an attempt to deal with search and optimization problems. To increase the search capability of selecting distinctive features, a hybrid PSO-based FS algorithm with a local search strategy (called HPSO-LS) was proposed in [32]. The local search strategy provided by employing the correlation information of the features helps the search process select less the best features. Amoozegar and Minaei-Bidgoli [13] proposed a multi-objective FS algorithm, namely RFPSOFS, that ranked the features based on their occurrences in the archive set. Then, the archive set was refined according to these

rated features. Additionally, involved in updating the particle position vector, these ranks caused the particles to move purposefully. Most PSO-based FS methods have used fixed-length representations causing high computational costs for high dimensional data. However, the first variable, dynamic length representation for the PSO-based FS method (called VLPSO) was proposed in [9] enabling the particles to have different lengths. In this way, the swarm was divided into several divisions so that each division had maximum length and the length of divisions were different from each other. Besides, the features were sorted in descending order of relevance and subsequently a division by shorter length considered the top rank features for the selection process. To avoid getting stuck in the local optimum, the length of each division could be changed during the evolution process. This algorithm improved the performance of PSO by concentrating its search on reduced space and more fruitful areas.

ACO, as one of the most well-known population-based optimization approaches which was proposed in 1999 [33], showed promise in the FS problem. For example, it was used with ANN in [18] for application in text FS that contains high dimensional data. In this method, two global and local rules were presented to update the pheromone level. While the global updating rule helps the algorithm generate feature subsets with a low rate of classification error, the local updating rule gives a chance for unrelated features which have not been investigated formerly to be selected and thus prevents early convergence. Generally, in ACO-based FS methods, there are multiple paths for a specific subset causing an uneven distribution of pheromones sediments. To overcome this problem, Ghosh et al. [34] assigned pheromones sediments to nodes instead of edges between nodes. They proposed a wrapper-filter FS (WFACOFS) method to reduce the computational complexity. The algorithm generated the feature subsets using a filter method and then evaluates them by a classifier. Additionally, a fitness-based memory was presented to keep the best solutions. So, in this way, FS was performed in a multi-objective manner.

In comparison with mentioned optimization algorithms, ABC was proposed later by Kraboga [35]. It deals with the optimization tasks using a vector-based representation, appropriate for solving the FS problems. With utilizing both binary and continuous representations, the authors in [36] have applied GA crossover and mutation to their multi-objective ABC-based method that was incorporated with the non-dominated sorting process. Moreover, they utilized both binary and continuous representations. Kuo et al. [20] proposed an ABC-based method selecting relevant features and simultaneously optimizing the SVM parameters. A cost-sensitive ABC-based feature selection approach called TMABC-FS was proposed in [21]. With minimizing the feature cost and maximizing the classification accuracy in the multi-objective problem modeling stage, TMABC-FS contributes to introduce two new operators, namely *diversity-guiding* and *convergence-guiding* searches for the onlooker and employed bees, respectively. Furthermore, it considers two archive sets, including *leader* and *external* archives, in order to improve the search procedure of different kinds of bees.

Estimation of distribution algorithm (EDA) is another population-based optimization method used in solving the FS problems. In [37], the EDA was applied to the multi-objective feature selection phase in an intrusion detection system (IDS). The authors claimed that the proposed approach (MOEDAFS) had lower complexity and higher classification accuracy. The compact genetic algorithm (cGA) [38] is an EDA-based method that represents the population in keeping with the estimated probabilistic model over the set of solutions instead of traditional genetic operators (crossover and mutation) in the traditional genetic algorithm. This method was also applied to solve the FS problem in [39].

To improve the performance of the FS problems, other population-based optimization methods, some of which are proposed in recent studies like the Cuckoo search algorithm in [23], firefly optimization in [40], and bat algorithm in [41], have also been used. To summarize this section, Table 1 compares the specifications of the surveyed methods.

Despite all advantages, the population-based optimization methods in FS problems suffer from several limitations that can be discussed in two directions:

1) High time complexity: large search space and consequently the large number of fitness function evaluations can lead to this problem.
2) Interaction between features: a feature may be weakly relevant to the target class individually, but the classification performance can be improved using some complementary features. Moreover, a feature may be relevant by itself, but it causes decreased classification performance when used with some other features.

To tackle the mentioned challenges relating to feature selection methods utilizing population-based optimization, the main contribution of this paper is to propose a correlation-aware EDA-based method and to apply it in an attempt to solve the FS problems. The next section will detail the structure of the proposed method will be described in detail.

3. The proposed method

In this paper, we present a correlation-aware feature selection algorithm that not only considers the importance of each feature alone, but also can deal with the interaction between features. A good FS method should select a subset that features have minimum correlation and at the same time increase the classification performance. Therefore, the proposed method has the capability of considering the complementary features, and consequently, the classification performance will be improved. In addition to this, as we know, one of the main limitations of the wrapper-based FS methods, which utilized population-based optimization approaches is suffering from high complexity of time due to a large number of fitness function evaluations. Fortunately, the proposed method generates only two individuals in each iteration. Like the cGA, the generated individuals compete with each other in each iteration of the algorithm based on a fitness function to determine the *winner* and the *loser*. These two individuals evolve during the algorithm to find

Table 1
Comparison of the state-of-the-art FS methods that utilized population-based optimization.

Method	Population-based algorithm	Type	Representation method	The number of objectives	Classifier
HGA-NN [28]	GA	Wrapper	Binary	SO	ANN
HGP-FS [29]	GA, PSO	Hybrid	Continues	SO	ANN
Ensemble-FSGA [30]	GA	Filter	Binary	MO	–
HPSO-LS [32]	PSO	Hybrid	Continues	SO	KNN
RFPSOFS[13]	PSO	Wrapper	Continues	MO	KNN
VLPSO [9]	PSO	Hybrid	Continues	SO	KNN
ACO-ANN [18]	ACO	Wrapper	Continues	SO	ANN
WFACOFS [34]	ACO	Hybrid	Continues	SO	KNN/ANN
Hancer et al. [36]	ABC	Wrapper	Binary/ Continues	MO	KNN
ABC-SVM-DT [20]	ABC	Wrapper	Continues	SO	SVM/DT
TMABC-FS [21]	ABC	Wrapper	Continues	MO	KNN
MOEDAFS [37]	EDA	Wrapper	Binary	MO	–
cGA-FS [39]	cGA	Wrapper	Continues	SO	Naive Bayes

the best solution. The number of features for each individual is determined based on our guiding technique, which will be discussed in Section 3.5. In this technique, the number of features for each individual is determined randomly by a chi-square distribution with d degrees of freedom in each iteration, where d is the number of *winner*'s features. In this way, the best value of d is determined using evolution process, too.

To consider both the effects of each feature alone and the interaction between features in the proposed method, we define two data structures. The first one is the *significance vector (SV)* with size n, and the second one is the *interaction matrix (IM)* with size $n \times n$, where n is the number of features. While $SV(i)$represents the goodness of the corresponding feature i, $IM(i,j)$denotes the goodness of simultaneous presence of two features i and j in the final solution. All elements of SV and IM are initialized by one to provide the features an equal chance of selection. Then, having been generated using the conditional probabilities, one of the main advantages of the proposed method, the generated individuals

/*Step 1: Initialization*/

Initialize the value d, the sets A and B, the vector SV and the matrix $IM_{n\times n}$, as:

$$d = n/2;\ \ A = \varnothing;\ B = \varnothing;\ SV(i) = 1;\ IM(i,j) = 1;\ \ i,j = 1,\ldots,n.$$

The_best_subset = [];

for i = 1 to *iter*

/*Step 2: Generating two individuals *a* and *b/**

Select the first feature for each of a and b with roulette wheel using probabilities P:

$$P(a_j^1) = SV(j) / \sum_{k=1}^{n} SV(k);\ \ P(b_j^1) = SV(j) / \sum_{k=1}^{n} SV(k);$$

Select the number of features for each individual by chi-square distribution:

$$s_a \sim \chi^2(d);\ \ s_b \sim \chi^2(d);$$

for k =2 to s_a

Select k-th feature for individual a with roulette wheel using probabilities:

$$P(a_j^k \mid a_l \in A) = \frac{\left(\prod_{a_l \in A} IM(a_j, a_l)\right) \times SV(a_j)}{\left(\sum_{a_z \in A} \prod_{a_l \in A} IM(a_z, a_l) \times SV(a_z)\right)};$$

End for

for k =2 to s_b

Select k-th feature for individual b with roulette wheel using probabilities:

$$P(b_j^k \mid b_l \in B) = \frac{\left(\prod_{b_l \in B} IM(b_j, b_l)\right) \times SV(b_j)}{\left(\sum_{b_z \in B} \prod_{b_l \in B} IM(b_z, b_l) \times SV(b_z)\right)};$$

End for

/*Step 3: Competition*/

Calculate the fitness of each individual a and b;

if $fitness(a) \geq fitness(b)$

winner=*a*; *loser*=*b*;

else

winner=*b*; *loser*=*a*;

End if

if (fitness(*winner*) > fitness (*The_best_subset*))

The_best_subset = *winner*;

End if

/*Step 4: Update *SV* and *IM* using the update procedure*/

/*Step 5: Update the estimated number of features for individuals by our guiding technique*/

d=the number of features selected by the *winner*;

End for

Return *The_best_subset* as the final solution;

Fig. 1. Pseudocode of the proposed method.

compete in an attempt to determine the *winner* and the *loser*. Finally, *SV* and *IM* are updated using our update procedure that will be described in the following this section. The pseudocode of the proposed method is described in Fig. 1.

In step 1, we initialized all of our variables and data structures. Then, based on our introduced probabilities scheme, which will be discussed later, two individuals are created in step 2. The main advantage of these probabilities is that the correlated features are given less probability values. Thus, they will have little chances of being selected. In the next step, the created individuals compete based on a fitness function and the winner and loser are determined. In step 4, the introduced *SV* and *IM* data structures are updated based on our introduced update procedure, which will be discussed in Section 3.4. Finally, the guiding technique helps the algorithm to select the appropriate number of features during the evolution process. The number of features for each individual is determined randomly by chi-square distribution with d degrees of freedom in each iteration, where d is the number of *winner*'s features. These steps repeat until the algorithm is stopped. In the following, we will explain the full details of each step in separate subsections.

3.1. Initialization

In the first step, we initialize the number of selected features by *winner* by $n/2$. Moreover, the sets A and B, which indicate the selected features for individuals a and b are empty. Finally, all the elements of the significance vector *SV* and the interaction matrix *IM* are set to 1. It should be noted that the *SV* and *IM* data structures are used to determine the probability of selecting the features in the following steps. By assigning equal values to them, all features have the same chance to be selected at the beginning of the algorithm.

3.2. Generating two individuals

To select the best feature subset, the algorithm generates two independent individuals in each iteration. The probability of selecting the first feature of each individual a and b is determined by its associated significance value divided by the sum of the significant value of all features:

$$P\left(a_j^1\right)=\frac{SV(j)}{\sum_{k=1}^{n} SV(k)};\quad P\left(b_j^1\right)=\frac{SV(j)}{\sum_{k=1}^{n} SV(k)},\ \forall j=1,\ldots,n. \tag{1}$$

The first feature is selected using the roulette wheel mechanism for each individual based on the calculated probabilities in Eq. (1). Greater probability value gives more chance to jth feature to be selected as the first one. In the first iteration, the probability of selecting each feature is $1/n$. However, since the *SV* is updated at the end of each iteration, it will be different for each feature in the successive iterations.

Similarly, the k-th feature of each individual is selected using the roulette wheel mechanism based on the following conditional probabilities:

$$P\left(a_j^k\middle|a_l\in A\right)=\frac{\left(\prod_{a_i\in A} IM(a_j,a_i)\right)\times SV(a_j)}{\left(\sum_{a_z\in\bar{A}}\prod_{a_l\in A} IM(a_z,a_l)\times SV(a_z)\right)},\quad \forall k=2,\ldots,s_a, \tag{2}$$

$$P\left(b_j^k\middle|b_l\in B\right)=\frac{\left(\prod_{b_i\in B} IM(b_j,b_i)\right)\times SV(b_j)}{\left(\sum_{b_z\in\bar{B}}\prod_{b_l\in B} IM(b_z,b_l)\times SV(b_z)\right)},\quad \forall k=2,\ldots,s_b, \tag{3}$$

where in Eq. (2), $P(a_j^k|a_l\in A)$denotes the probability of jth element of individual a (a_j) to be selected as the k-th feature, when a set of features A is selected in the previous iterations. The numerator represents the goodness of simultaneous presence of a_j and previously selected features and also the significance value of a_j. The denominator represents the goodness of simultaneous presence of unselected features $\bar{A}$ and previously selected features A and also the significance value of each unselected feature. Similarly, the probability of jth element of individual b (b_j) to be selected as the k-th feature is determined by Eq. (3). This process continues until s_a and s_b features are selected for individuals a and b, respectively. It should be noted that and s_b are random numbers that determined by chi-square distribution with d degrees of freedom in each iteration, where d is the number of *winner*'s features. We will discuss determining these variables in Section 3.5. It should be noted that however the *IM* reflects the interaction between only two features, the introduced probability in Eqs. (2) and (3) considers the goodness of selecting one feature given selecting a subset of selected features.

3.3. Competition

The generated individuals a and b from the previous step are then evaluated according to the following fitness function:

$$fitness=\frac{accuracy}{SFR},\qquad SFR=\frac{number\ of\ selected\ features}{total\ number\ of\ features}. \tag{4}$$

According to Eq. (4), the fitness value of each individual is calculated by the classification accuracy achieved by the corresponding selected features of the individual divided by the selected feature rate (SFR). In this way, we can deal with both increasing the classification performance and decreasing the number of selected features. In calculating the accuracy of each candidate solution, our algorithm benefits from the advantages of support vector machines (SVMs) as the classifier, including the generalization ability, strong theoretical foundations, absence of local minima, and robustness against noise. After fitness evaluation, the individual with greater (smaller) fitness is called the *winner* (*loser*). The *winner* and the *loser* are binary vectors with length n. Thus, w_i=1 indicates that the i th feature has been selected by the *winner*, while w_i=0 means that the i th feature has not been selected. Similarly, elements of the *loser* vector are considered as l_i, which can be either 0 or 1. They are used to update the *SV* and *IM* in the next step of the proposed algorithm. If the current winner's fitness is greater than the fitness of the best solution has been found so far, the best solution is replaced by the current *winner*.

3.4. Update procedure

In this step, *SV* and *IM* should be updated using the obtained *winner* and *loser*. We update these two data structures using Table 2 and Table 3, respectively.

As shown in Table 2, each element of the *SV* is updated based on the corresponding values of the *winner* and the *loser* vectors. In the case that a feature has been selected by both the winner and the loser or that a feature has not been selected by both, we cannot decide whether it would be good to select the corresponding feature or not. Therefore, the corresponding value in *SV* remains unchanged. In other case that a feature is selected by the loser but not by the *winner*, we decrease the chance of selecting it by a predefined value between zero and one, called

Table 2
The update procedure of *SV* based on the *winner* and the *loser* vectors.

loser / winner	$l_i=\mathbf{0}$	$l_i=\mathbf{1}$
$w_i=\mathbf{0}$	*SV*	*SV*−
$w_i=\mathbf{1}$	*SV*+	*SV*

Table 3
The update procedure of *IM* based on the *winner* and the *loser* vectors.

loser / *winner*	$l_i, l_j = \mathbf{0}, \mathbf{0}$	$l_i, l_j = \mathbf{0}, \mathbf{1}$	$l_i, l_j = \mathbf{1}, \mathbf{0}$	$l_i, l_j = \mathbf{1}, \mathbf{1}$
$w_i, w_j = \mathbf{0}, \mathbf{0}$	*IM*	*IM*	*IM*	*IM* −
$w_i, w_j = \mathbf{0}, \mathbf{1}$	*IM*	*IM*	*IM*	*IM* − −
$w_i, w_j = \mathbf{1}, \mathbf{0}$	*IM*	*IM*	*IM*	*IM* − −
$w_i, w_j = \mathbf{1}, \mathbf{1}$	*IM* +	*IM* + +	*IM* + +	*IM*

change factor which is used to strengthen or weaken the significance value of feature *i*. In the last case, when the *winner* selects a feature while the *loser* does not so, the chance of its selection is increased by the *change factor*.

In Table 3, the selecting status of each pair of features *i* and *j* is compared, and some updates on *IM* are performed based on differences between *winner* and *loser*. In the following, we discuss how to update the *IM* in some states. The update procedure of the remaining states is similar.

- w_i=l_i *and* w_j=l_j: for all four states with this condition, main diagonal of the table, no updates on *IM* are made because the *winner* and the *loser* has done the same for selecting features *i* and *j*. Therefore, we cannot decide whether selecting or not selecting features *i* and *j* together, is good or not.
- (w_i,w_j=0,0) and (l_i,l_j=0,1): in this state, both the *winner* and the *loser* have not selected feature *i*. Thus, we cannot decide about the goodness of selecting this feature. However, since the *loser* has selected the *j*th feature and the *winner* has not selected it, we decrease the chance of selecting it by the introduced *change factor*.
- (w_i,w_j=0,0) and (l_i,l_j=1,1): here, the *winner* has not selected any of *i* or *j*, while the *loser* has selected both of them. Therefore, we decrease the $IM(i.j)$by the *change factor*.
- (w_i,w_j=0,1) and (l_i,l_j=0,0): in this state, the *winner* has selected the *j*th feature, and none of the features *i* and *j* have been selected by the *loser*. Hence, the chance of simultaneous appearance of features *i* and *j* remains unchanged.
- (w_i,w_j=0,1) and (l_i,l_j=1,1): since the *winner* has not selected both features *i* and *j* together, and at the same time, the *loser* has selected both of them, and we decrease the $IM(i.j)$by a stronger value than the *change factor* (i.e., two or three times higher), since in this state, we are more confident in decreasing or increasing the chance of selecting the features.

3.5. Update the estimated number of features for individuals by our guiding technique

One of the advantages of the proposed algorithm is that the number of features for each of two individuals in each iteration is determined based on the number of *winner*'s features, a guide to select optimum features. To this aim, the number of each individual (s_a and s_b) in Fig. 1, are random numbers determined by chi-square distribution with *d* degrees of freedom, where *d* is the number of *winner*'s features. For the first iteration, *d* is initialized to $n/2$. The expected value for the chi-square distribution is equal to *d*, but it is possible to take the values more or less as well. Since the variable *d* is defined as the number of *winner*'s features, which will be updated in each iteration, the number of selected features of the final solution will be optimized during the evolution process. Moreover, this guiding mechanism ensures that the number of features for each individual does not exceed the value determined by the chi-square distribution. This can directly increase the convergence speed of the algorithm due to the limitation on the number of features for each individual, and can help the proposed algorithm select fewer features. In Section 4.5, we will discuss the results of applying this technique to different datasets.

4. Experimental results

In this section, comparing with state-of-the-art studies, we evaluate our method using different datasets. It should be noted that the proposed algorithm was implemented using MATLAB® 2018a. Besides, all the experiments were performed on a machine with 2.60 GHz Intel Core i7 processor and 6.0GB of DDR3 memory. We will compare our proposed method with GA-SVM, cGA-FS [39], WFACOFS [34], MOEDAFS [37], and RSVM-SBS [42]. To select the best feature subset, GA-SVM utilizes the genetic algorithm as the optimization technique and support vector machines as the fitness function. The way in which the following three approaches operate were described in Table 1, in Section 2. As for the last one, RSVM-SBS combines the sequential backward search (SBS) with noise-aware support vector machines, namely RSVM, to deals with the FS problem in the presence of outliers. It is worth bearing in mind that in all experiments, the datasets were firstly divided randomly into 75% training, and 25% testing sets, which to ensure a fair comparison, we used the same training and testing ones for all methods. As well as, the *change factor* value in our updating procedure was set to 0.01. Finally, since each method has some parameters to be tuned, we determined the best parameter values for each one using trial and error for a fair comparison. For this purpose, we ran each method with the same number of fitness function evaluation and determined the best values of the hyper parameters. Table 4 summarized the parameters utilized to set up all algorithms under comparison.

4.1. Datasets

The details of datasets used to assess the proposed approach compared with that of other approaches in the literature are summarized in Table 5. All datasets in our experiments are obtained from the UCI Repository [44]. These datasets are from various fields and can be categorized based on the number of features into three groups: small, medium, and large. A dataset with less than ten features is considered small, while it is placed in the large category if its number of features is more than 100, otherwise it would be a medium dataset. We tested our algorithm on two small, seven medium, and four large datasets.

4.2. Performance metrics

To measure the performance of different methods, some well-known metrics were used, including accuracy, precision, recall, F1-score, and a one introduced in [42], i.e., the product of accuracy (*ACC*) rate and the

Table 4
Parameter setting.

Method	Parameters	Settings
WFACOFS	Exploitation balance factor	1
	Exploration balance factor	1
	Weight of accuracy	100
	Weight of number of features	1
	Pheromone evaporation factor	0.15
	Pheromone evaluation factor	0.8
RSVM-SBS	Kernel function	RBF
	C	100
	σ	0.5
GA-SVM	Crossover rate	0.7
	Mutation rate	0.01
	Selection mechanism	Roulette wheel
cGA-FS	Np	0.6
C4.5	Confidence factor	0.25
	Min. instance per leaf	2
Random forest	The number of trees	200
	mtry[a]	$\sqrt{n}$
OblRF-H [43]	The number of trees	500
	Mtry	$\sqrt{n}$
The proposed Method	Change factor	0.01

[a] the number of the candidate features in each split.

Table 5
Details of datasets.

Dataset	Size	Number of samples	Number of features	Number of classes
Breast Cancer	Small	699	9	2
Glass	Small	214	9	6
Heart	Medium	270	13	2
Wine	Medium	178	13	3
Segmentation	Medium	2310	19	7
German	Medium	1000	24	2
Ionosphere	Medium	351	34	2
Soybean-small	Medium	47	35	4
Sonar	Medium	208	60	2
Hill-valley	Large	1212	100	2
Musk1	Large	476	167	2
Arrhythmia	Large	452	279	16
Isolet5	Large	1559	617	26

percentage of discarded features (*PDF*). These metrics are defined as follows:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}, \tag{5}$$

$$Precision = \frac{TP}{TP + FP}, \tag{6}$$

$$recall = \frac{TP}{TP + FN}, \tag{7}$$

$$F1 - score = \frac{2TP}{2TP + FP + FN}, \tag{8}$$

$$ACC \times PDF = Accuracy \times percentageofdiscardedfeatures, \tag{9}$$

where TP, TN, FP, and FN are the number of true positives, true negatives, false positives, and false negatives, respectively.

4.3. The results on a synthetic dataset with correlated features

One of the main contributions of the proposed method is that it can deal with the correlation between features. To investigate this, we generated a synthetic dataset. This dataset consisting of 10 correlated features and 250 samples with random values in two classes. There was no correlation between the first six features, but the values of other features are generated as follows:

$$f_7 = 10 \times f_1, \quad f_8 = (f_2 + 3 \times f_3), \quad f_9 = f_4, \quad f_{10} = f_5/1000, \tag{10}$$

where f_i denotes the i th feature. A good FS method should not select the correlated features. For example, based on Eq. (10), only one of f_1 or f_7 should appear in the final solution for our synthetic dataset. After running the proposed method on the synthetic dataset, it selects $\{f_1,f_2,f_3, f_4,f_{10}\}$ as the best feature set with 98.39% accuracy. The proposed method did not select any correlated feature. Table 6 summarizes the obtained results for different methods on the synthetic dataset.

As shown in Table 6, the proposed method has selected the least number of features. Meanwhile, it has reported higher accuracy, and most importantly, has not selected any correlated feature. However, the other methods failed to do so. For example, cGA-FS has selected f_4 and f_9, simultaneously. Similarly, WFACOFS has selected f_1 and f_7 as well as f_5 and f_{10}. The other methods also have behaved similarly. To better understand how the proposed approach obtained these results, we should review the role of the probability scheme, introduced in the previous section. Fig. 2 represents the heatmap matrix (HM) of our introduced conditional probabilities determined by Eqs. (2) and (3) after 1000 runs of the proposed method on the synthetic dataset. The correlated features are highlighted in bold.

Table 6
The accuracy (%) for different methods on the synthetic dataset.

Method	Accuracy	Selected features
GA-SVM	98.38	$\{f_1,f_5,f_6,f_7,f_8,f_9\}$
cGA-FS	80.76	$\{f_1,f_2,f_3,f_4,f_8,f_9\}$
WFACOFS	79.03	$\{f_1,f_3,f_5,f_6,f_7,f_9,f_{10}\}$
MOEDAFS	83.87	$\{f_1,f_4,f_5,f_7,f_8,f_9,f_{10}\}$
RSVM-SBS	81.69	$\{f_1,f_2,f_5,f_6,f_7,f_8,f_{10}\}$
The proposed Method	98.39	$\{f_1,f_2,f_3,f_4,f_{10}\}$

Each element *HM(i,j)* in Fig. 2 denotes the probability of selecting the *j*th feature given the *i* th one with the sum of the conditional probabilities at any column is equal to one, as expected. As we can conclude from the heatmap matrix, the probabilities of selecting the correlated features have decreased using our update procedure (see Section 3.4). For example, *HM(5,10)* and *HM(10,5)* have the lowest values in their rows. This means when we select f_5, the probability of selecting f_{10} is less than the other remaining features. Likewise, when f_{10} is selected by the proposed algorithm, the probability of selecting f_5 is less than the other features. Smaller values in each row indicate more correlation between corresponding features. As we described earlier, although the *IM* reflects the interaction between only two features, the proposed algorithm considers other interactions (see Section 3.2). For example, as shown in the first row of *HM* in Fig. 2, the value of *HM(1,6)* and *HM(1,5)* are lower than *HM(1,7)* but the algorithm has not selected f_1 and f_7, simultaneously.

4.4. The results on real-world datasets

Here, we compare our proposed method with other approaches on real-world datasets. For the first experiment in this subsection, we compare our results with some basic and also state-of-the-art decision tree-based classifiers. It should be mentioned that we chose them for comparison because they perform feature selection implicitly. Tables 7 and 8 show the obtained results of C4.5, random forests, and OblRF-H (with and without feature selection) [43] in comparison with the proposed method in terms of accuracy and F1-score.

As shown in Tables 7 and 8, the proposed method reported better performance in almost datasets in compared with other methods. To prove the performance of the proposed method in comparison to other methods, two non-parametric statistical tests, namely Wilcoxon's signed-rank test [45] and Friedman's test [46], with a significance level of 0.05, were performed. The Wilcoxon's signed-rank test was utilized for pairwise performance evaluation between the proposed approach and the other methods. Table 9 summarizes the Wilcoxon's signed-rank test results in terms of accuracy and F1-score metrics.

Results from Table 9 indicate that the proposed approach shows a significant difference from the compared methods on all datasets.

The Friedman's test was also applied to evaluate the performance of all compared methods. The *p*-values for this test in terms of accuracy and F1-score metrics were equal to 3.65E-5 and 2.06E-4, respectively, indicating that the overall performance of the proposed method is significantly better than the others.

As we know, the random forest method also provided a ranking of the features. For the next experiment, it would be interesting to compare the best features obtained by the proposed method and those reported by the random forest method. In Table 10, we show the best *s* features that selected by the proposed method based on our obtained probabilities and also the *s* top ranked features that were reported by the random forest. Moreover, the classification results using support vector machines for the best features are reported in Table 10.

As shown in Table 10, the selected features by the proposed method are more reliable and provide better performance than that of the random forest.

For the next experiment, it would be interesting to compare our proposed method with different classic feature selection methods that

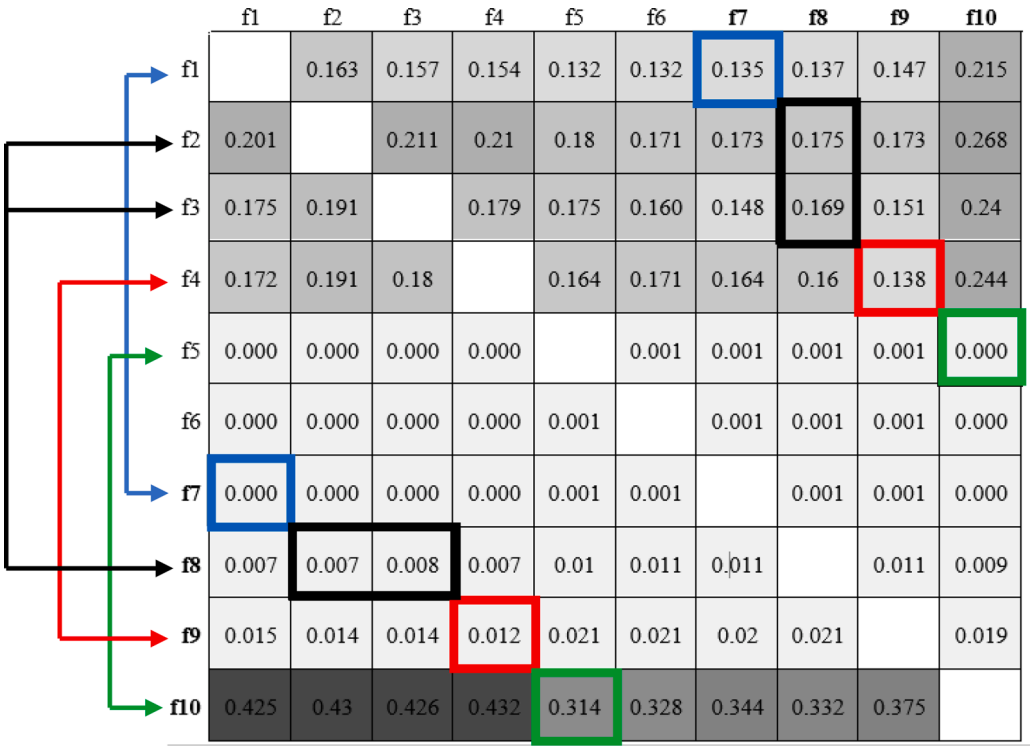


Fig. 2. The conditional probability heatmap matrix. The correlated features have been shown with different colors. Smaller values in each row indicate more correlated between corresponding features.

Table 7
The average accuracy (%) metric for different methods on different datasets.

	C4.5	Random forest	OblRF-H (without FS)	OblRF-H (with FS)	The proposed method
Breast Cancer	96.57	97.14	97.14	98.23	**98.85**
Glass	66.04	**75.47**	70.26	71.04	71.70
Heart	83.58	79.10	84.51	86.48	**91.04**
Wine	97.73	93.18	97.32	98.62	**99.36**
Segmentation	95.49	**96.17**	95.05	95.36	94.80
German	75.20	76.80	75.15	79.31	**82.80**
Ionosphere	93.18	94.32	94.16	**95.58**	95.45
Soybean-small	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**
Sonar	75.00	88.46	85.63	89.14	**92.30**
Hill-valley	52.15	54.13	59.46	61.74	**71.61**
Musk1	79.83	82.35	85.24	**86.62**	84.03
Arrhythmia	66.37	71.45	71.08	72.40	**72.66**
Isolet5	76.41	88.79	87.96	89.15	**90.00**

Table 8
The average F1-score (%) metric for different methods on different datasets.

	C4.5	Random forest	OblRF-H (without FS)	OblRF-H (with FS)	The proposed method
Breast Cancer	96.30	97.05	97.63	98.06	**98.89**
Glass	55.56	65.80	66.14	**67.21**	64.78
Heart	82.85	78.29	83.47	86.76	**92.25**
Wine	98.09	93.74	96.35	96.64	**99.46**
Segmentation	95.58	97.61	**98.51**	97.20	95.28
German	70.79	69.39	74.20	73.41	**77.91**
Ionosphere	92.84	93.12	93.62	95.00	**95.72**
Soybean-small	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**
Sonar	74.80	88.44	82.39	90.64	**93.12**
Hill-valley	52.17	54.08	60.24	69.10	**75.39**
Musk1	78.96	81.33	82.86	82.53	**83.14**
Arrhythmia	39.87	40.01	38.24	39.79	**40.10**
Isolet5	77.20	90.19	88.28	90.21	**90.36**

was utilized in [47] for intrusion detection application. The authors implemented Information Gain (IG), Chi-Square (CS), and Recursive Feature Elimination (RFE) feature selection techniques with different classifiers. We also used these three FS methods for different datasets. Being in filter-based category, the first two methods rank the features based on the chi-square score and the information gain score, respectively. For each dataset, we calculated the average obtained weighs of ranked features and selected those features with higher weights than the average. Since SVM reported the best results among different classifiers in [47], we used it to evaluate selected features that are obtained by different methods. Table 11 reports the obtained results after 10-fold cross-validation.

As shown in Table 11, the proposed method outperformed the other methods in almost all datasets. The statistical tests also prove that the proposed method was significantly better than the other methods. Table 12 summarizes the Wilcoxon's signed-rank test results in terms of both accuracy and F1-score metrics.

Table 9
P-value of Wilcoxon signed-rank test between the proposed approach and each other methods in terms of accuracy and F1-score.

Method	*p*-value (accuracy)	*p*-value (F1-score)
C4.5	0.00097	0.00146
Random forest	0.02539	0.01611
OblRF-H (without FS)	0.00244	0.02685
OblRF-H (with FS)	0.03417	0.04248

The results obtained from Table 12 show that the proposed approach indicates a significant difference compared with other methods on all datasets.

The Friedman's test was also applied to evaluate the performance of all compared methods. The *p*-values for this test in terms of accuracy and

Table 10
The best feature subsets and their performances.

	Random forest		The proposed method	
	Best features	Accuracy	Best features	Accuracy
Breast Cancer	{3,4,6,7}	96.00	{2,4,7,9}	**97.71**
Glass	{1,2,3,4,6,7}	**67.92**	{1,3,4,5,6,8}	**67.92**
Heart	{8,10,12,13}	88.06	{2,3,12,13}	**94.03**
Wine	{7,10,12,13}	97.73	{1,10,12,13}	**100.00**
Segmentation	{2,10,11,12,16,17,19}	90.12	{1,2,10,12,16,18,19}	**92.20**
German	{1,3,4,5,8,9,10,11,12,13}	76.80	{1,2,3,4,5,9,16,17,19,20}	**78.80**
Ionosphere	{3,4,5,6,7,8,16,27}	87.50	{1,4,5,7,8,12,15,31}	**92.05**
Soybean-small	{21,22}	**100.00**	{21,22}	**100.00**
Sonar	{9, 10,11,12,13,51}	86.54	{1,10,11,12,37,48}	**92.30**
Hill-valley	S1*	52.15	S2*	**60.73**
Musk1	S1*	**73.95**	S2*	**73.95**
Arrhythmia	S1*	68.14	S2*	**69.91**
Isolet5	S1*	89.23	S2*	**90.26**

* Those sets indicated by S1 and S2 have more than 15 members and are not shown.

Table 11
The average accuracy (Acc), F1-score (F1), and the number of selected features (SF) for different methods.

	IG-FS			CS-FS			RFE-FS			The proposed method		
	Acc	F1	SF	Acc	F1	SF	Acc	F1	SF	Acc	F1	SF
Breast Cancer	95.43	94.71	7.2	94.86	94.18	7.4	94.29	93.37	5.3	**98.85**	**98.89**	**4.3**
Glass	60.38	54.43	5.3	54.72	50.99	**4.7**	62.26	57.76	8.6	**71.70**	**64.78**	6.5
Heart	82.09	83.27	6.7	82.09	80.86	7.3	83.54	84.50	10.2	**91.04**	**92.25**	**4.6**
Wine	88.64	89.95	7.2	97.73	97.97	6.4	90.91	91.83	7.8	**99.36**	**99.46**	**4.3**
Segmentation	82.84	81.86	7.8	84.24	86.54	8.9	84.23	83.58	**5.6**	**94.80**	**95.28**	8.7
German	72.40	65.57	9.2	74.40	65.59	**9.1**	75.60	70.60	16.4	**82.80**	**77.91**	14.8
Ionosphere	81.82	81.03	15.6	87.50	86.67	19.4	88.64	88.35	28.1	**95.45**	**95.72**	**12.0**
Soybean-small	**100.00**	**100.00**	12.3	**100.00**	**100.00**	13.2	**100.00**	**100.00**	6.5	**100.00**	**100.00**	**3.1**
Sonar	76.92	76.48	28.1	71.15	71.30	22.6	78.85	78.65	52.4	**92.30**	**93.12**	**13.7**
Hill-valley	60.07	65.79	52.4	52.84	56.08	43.2	50.83	33.70	79.3	**71.61**	**75.39**	**33.4**
Musk1	72.63	75.52	90.6	70.59	70.56	75.1	82.35	82.16	95.1	**84.03**	**83.14**	**34.4**
Arrhythmia	61.06	28.45	121.4	59.29	39.65	101.5	55.75	30.40	98.2	**72.66**	**40.10**	**22.0**
Isolet5	84.12	80.24	305.1	79.45	80.64	276.8	87.18	85.42	241.4	**90.00**	**90.36**	**141.3**

Table 12
P-value of Wilcoxon signed-rank test between the proposed approach and each other methods in terms of accuracy and F1-score.

Method	*p*-value (accuracy)	*p*-value (F1-score)
IG-FS	0.0005	0.0005
CS-FS	0.0005	0.0005
RFE-FS	0.0005	0.0005

F1-score metrics were equal to 4.01E-5 and 2.88E-05, respectively, indicating that the overall performance of the proposed method is significantly better than the others. Furthermore, it is interesting to investigate the performance of the proposed method on a real application. Thus, we compared our method with [47] for intrusion detection on NSL-KDD dataset [48]. This dataset is developed to address the limitations of KDD CUP 99 dataset. It contains 41 features, 149,470 instances, and five classes (DoS, Probe, R2L, U2R, and Normal). Table 13 shows the results of different feature selection methods for each of four attack classes. It should be noted that SVMs are used as the classifier.

As indicated in Table 13, the proposed method outperforms the others in almost all attacks.

In the rest of this subsection, we compare our proposed method with state-of-the-art feature selection methods. Tables 14–19 summarize the obtained results in terms of different performance metrics after ten independent runs. The best results in each table are highlighted in boldface.

The proposed method, As shown in Table 14, reported the best accuracies in eight datasets. GA-SVM obtained better accuracies on four datasets in comparison with the proposed method. Finally, WFACOFS and RSVM-SBS also had better results in Segmentation and Glass datasets, respectively. It should be emphasized that high accuracy does not necessarily indicate the high efficiency of a feature selection method. For example, considering the hill-valley dataset, GA-SVM selects 92 features from all 100 features, whilst the proposed method selects only 24 ones. In the rest of this section, we will report the best number of selected features in each method. Table 15 summarizes the average precisions of different methods on different datasets.

As reported in Table 15, our proposed approach outperformed the

Table 13
The average accuracy (Acc) and F1-score (F1) for different FS methods on NSL-KDD dataset.

	DoS		Probe		R2L		U2R	
	Acc	F1	Acc	F1	Acc	F1	Acc	F1
IG-FS	**98.88**	98.40	99.20	98.78	98.65	98.41	97.68	92.72
CS-FS	97.31	97.12	97.31	96.52	93.63	96.39	90.34	91.45
RFE-FS	98.76	98.42	98.75	98.72	98.45	98.68	**99.12**	**99.01**
The proposed method	**98.88**	**99.24**	**99.35**	**99.39**	**99.12**	**99.32**	98.27	98.87

Table 14
The average accuracy (%) metric for different methods on different datasets.

	All-features	GA-SVM	cGA-FS	WFACOFS	MOEDAFS	RSVM-SBS	The proposed Method
Breast Cancer	70.29	96.57	94.85	97.71	96.00	96.58	**98.85**
Glass	56.60	62.26	67.92	69.81	69.81	**72.56**	71.70
Heart	82.09	88.06	85.07	83.58	89.55	83.75	**91.04**
Wine	93.18	97.89	97.22	97.72	98.83	98.04	**99.36**
Segmentation	93.07	93.58	87.69	**95.32**	94.28	91.76	94.80
German	73.60	82.40	79.20	77.60	80.40	78.91	**82.80**
Ionosphere	86.36	92.05	88.63	94.31	89.77	90.69	**95.45**
Soybean-small	**100.00**	**100.00**	90.00	**100.00**	**100.00**	**100.00**	**100.00**
Sonar	73.08	78.84	82.69	90.38	90.38	89.03	**92.30**
Hill-valley	48.18	**92.73**	59.07	51.48	53.46	61.79	71.61
Musk1	82.35	**92.43**	68.06	86.55	86.55	82.74	84.03
Arrhythmia	51.32	**74.33**	61.94	60.17	61.06	63.97	72.66
Isolet5	76.40	**95.38**	90.76	82.05	84.10	88.45	90.00

Table 15
The average precision (%) metric for different methods on different datasets.

	All-features	GA-SVM	cGA-FS	WFACOFS	MOEDAFS	RSVM-SBS	The proposed Method
Breast Cancer	71.53	96.67	94.56	97.81	95.98	96.80	**98.80**
Glass	43.53	63.30	54.19	61.42	59.59	66.73	**68.75**
Heart	81.91	86.22	84.85	83.88	88.51	86.46	**91.70**
Wine	93.89	97.87	97.43	98.14	98.43	97.52	**99.86**
Segmentation	93.28	93.35	88.03	**95.40**	94.46	73.19	94.81
German	76.26	72.89	70.80	70.29	69.23	71.37	**77.24**
Ionosphere	82.38	89.45	77.27	93.73	85.61	90.07	**94.29**
Soybean-small	**100.00**	**100.00**	88.12	**100.00**	**100.00**	**100.00**	**100.00**
Sonar	76.67	80.00	82.68	90.29	90.74	91.61	**92.15**
Hill-valley	48.13	**92.79**	56.25	52.72	54.43	63.71	70.43
Musk1	82.44	**92.10**	68.43	86.50	87.00	87.11	83.75
Arrhythmia	29.87	**55.14**	38.76	13.61	18.18	33.08	36.20
Isolet5	67.96	**95.82**	89.73	81.75	83.83	80.57	90.07

Table 16
The average recall (%) metric for different methods on different datasets.

	All-features	GA-SVM	cGA-FS	WFACOFS	MOEDAFS	RSVM-SBS	The proposed Method
Breast Cancer	68.88	96.31	94.24	96.80	94.38	95.35	**98.80**
Glass	33.85	**70.58**	63.31	67.60	63.40	70.35	59.58
Heart	84.10	89.95	83.88	84.18	90.05	89.06	**90.95**
Wine	92.90	96.94	98.61	96.96	97.66	98.68	**98.81**
Segmentation	93.93	93.43	88.45	95.76	94.89	81.50	**95.90**
German	70.01	75.50	81.25	75.40	**82.77**	75.96	80.30
Ionosphere	85.42	93.24	93.42	94.24	86.74	94.85	**96.49**
Soybean-small	**100.00**	**100.00**	86.22	**100.00**	**100.00**	**100.00**	**100.00**
Sonar	80.56	78.44	82.44	90.47	91.67	89.91	**92.65**
Hill-valley	47.91	**92.81**	60.59	53.21	54.65	66.76	80.26
Musk1	82.33	**92.35**	68.88	86.50	87.32	90.31	83.58
Arrhythmia	24.12	**54.27**	40.08	16.16	38.77	43.34	46.78
Isolet5	71.81	**95.48**	89.72	83.36	85.58	89.10	89.90

Table 17
The average F1-score (%) metric for different methods on different datasets.

	All-features	GA-SVM	cGA-FS	WFACOFS	MOEDAFS	RSVM-SBS	The proposed Method
Breast Cancer	70.58	96.79	93.87	97.58	95.47	96.24	**98.89**
Glass	36.48	66.42	56.91	64.32	60.84	**67.58**	64.78
Heart	83.09	88.34	83.97	84.03	89.07	86.72	**92.25**
Wine	92.83	97.82	96.67	97.04	98.65	98.29	**99.46**
Segmentation	93.91	92.89	88.39	95.16	94.28	78.13	**95.28**
German	73.26	73.97	75.58	72.59	75.21	73.39	**77.91**
Ionosphere	83.13	90.43	84.15	93.21	86.73	93.05	**95.72**
Soybean-small	**100.00**	**100.00**	87.08	**100.00**	**100.00**	**100.00**	**100.00**
Sonar	79.28	79.93	83.28	91.10	91.92	91.47	**93.12**
Hill-valley	47.37	**93.02**	58.71	53.33	54.91	65.57	75.39
Musk1	81.85	**91.69**	68.12	85.97	86.79	88.15	83.14
Arrhythmia	25.98	**54.31**	38.70	14.07	24.04	36.81	40.10
Isolet5	70.21	**96.02**	90.10	82.92	85.07	84.99	90.36

Table 18
The average and the best (in parentheses) number of selected features for different methods on different datasets.

	All-features	GA-SVM	cGA-FS	WFACOFS	MOEDAFS	RSVM-SBS	The proposed Method
Breast Cancer	9 (9)	7.0 (5)	6.9 (6)	8.0 (7)	7.3 (6)	6.5 (6)	**4.3** (**4**)
Glass	9 (9)	7.5 (7)	8.0 (7)	6.6 (4)	6.5 (6)	6.8 (6)	**6.5** (**6**)
Heart	13 (13)	11.8 (11)	7.0 (6)	9.2 (8)	6.6 (5)	11.0 (10)	**4.6** (**4**)
Wine	13 (13)	11.5 (10)	8.0 (6)	7.1 (6)	9.7 (9)	10.3 (9)	**4.3** (**4**)
Segmentation	19 (19)	10.7 (9)	10.7 (**7**)	12.4 (9)	8.9 (**7**)	15.8 (14)	**8.7** (8)
German	24 (24)	17.9 (12)	17.1 (13)	16.9 (13)	18.2 (15)	18.9 (16)	**14.8** (**10**)
Ionosphere	34 (34)	26.3 (21)	12.7 (**5**)	**10.8** (6)	16.0 (6)	18.5 (12)	12.0 (8)
Soybean-small	35 (35)	17.2 (10)	5.5 (4)	16.8 (12)	15.5 (14)	16.3 (10)	**3.1** (**2**)
Sonar	60 (60)	54.3 (44)	42.0 (27)	39.1 (33)	19.3 (8)	30.7 (17)	**13.7** (**6**)
Hill-valley	100 (100)	68.1 (56)	39.7 (26)	56.9 (45)	65.3 (46)	63.4 (53)	**33.4** (**24**)
Musk1	167 (167)	143.2(122)	59.0 (47)	42.5 (29)	82.5 (71)	56.0 (41)	**34.4** (**22**)
Arrhythmia	279 (279)	188.9 (171)	120.9 (95)	44.7 (24)	134.9 (115)	87.2 (70)	**22.0** (**15**)
Isolet5	617 (617)	326.6 (273)	297.2 (247)	142.4 (97)	392.8 (327)	236.3 (181)	**141.3** (**94**)

Table 19
The average ACC × PDF (%) metric for different methods on different datasets.

	All-features	GA-SVM	cGA-FS	WFACOFS	MOEDAFS	RSVM-SBS	The proposed Method
Breast Cancer	0.00	32.19	26.87	16.28	25.06	29.50	**53.26**
Glass	0.00	12.10	11.31	**28.69**	21.33	20.95	21.90
Heart	0.00	10.83	42.53	28.28	49.59	16.10	**53.22**
Wine	0.00	17.18	45.00	48.47	27.92	25.56	**67.86**
Segmentation	0.00	45.06	46.84	41.63	54.82	19.80	**55.13**
German	0.00	31.07	29.53	29.25	24.79	21.53	**40.02**
Ionosphere	0.00	28.01	65.55	66.00	60.72	50.01	**67.37**
Soybean-small	0.00	45.42	75.06	58.85	53.56	62.42	**92.71**
Sonar	0.00	14.25	35.13	36.07	70.56	53.63	**77.14**
Hill-valley	0.00	35.19	39.66	25.24	23.70	25.82	**51.05**
Musk1	0.00	19.03	46.45	68.02	46.77	58.70	**69.84**
Arrhythmia	0.00	26.38	37.96	52.75	33.71	45.94	**67.84**
Isolet5	0.00	49.03	50.73	66.13	35.03	58.53	**72.83**

other methods in eight datasets. However, GA-SVM showed the better results on four datasets as well as WFACOFS on Segmentation in comparison with the proposed method. Table 16 reports the average recall of different methods on different datasets.

As can be seen from Table 16, our proposed algorithm achieved the best recall on seven datasets. However, GA-SVM and MOEDAFS methods reported improved performance compared with our algorithm on five and two datasets, respectively. The F1-score results of different methods are represented in Table 17.

Table 17 results that our proposed approach obtained the best results on eight datasets, whilst GA-SVM achieved better outcomes in terms of F1-score on four datasets, and RSVM-SBS showed improved performance on Glass dataset.

For the next experiment, we investigated the number of features selected by each method on different datasets, comparing in Table 18 the average and the best number of selected features of different methods in the final solution after ten runs.

It can be seen from the data in Table 18 that the average number of selected features achieved by the proposed algorithm was less than that of the other methods on all datasets except for Ionosphere. Additionally, the proposed algorithm outperformed the other methods in term of the best number of selected features except for Ionosphere dataset.

As discussed earlier, higher performance metrics which were summarized in Tables 14–18, does not guarantee higher efficiency in a feature selection method. In addition to the reported performance metrics, a good FS method should optimize the number of selected features. The reason is that minimizing the number of selected features increases the generalizability of the model and decreases its complexity. Thus, we provided a criterion which is introduced in [42], namely the product of accuracy (*ACC*) rate and the percentage of discarded features (*PDF*). Table 19 summarizes the obtained results for different methods in term of this new performance metric.

The obtained results in Table 19 proved that the proposed approach increased the accuracy and simultaneously decreased the selected number of features. Our proposed approach achieved the best results on 12 out of 13 datasets.

To say more about the superiority of the proposed method, we ranked different methods based on each performance metric on all datasets, representing the number of best ranks achieved by each method in Table 20.

From Table 20, it can be found that the proposed method reported more best ranks compared with other methods. To exemplify, it achieved the best rank on eight datasets in terms of the accuracy metric. Additionally, for some metrics, the sum of the best ranks exceeds the total number of datasets, indicating more than one method achieves the best result on some datasets. It can be observed from Table 20 that the proposed method obtained promising results in terms of ACC × PDF. It also proves that the proposed method considered increasing the accuracy and simultaneously decreasing the number of selected features.

For the next experiment, Fig. 3 represents the average ranks of different with regard to different performance metrics.

Fig. 3 illustrates that the proposed method reported lower average ranks in all terms of performance metrics. It is worth bearing in mind that the lower rank indicates better results.

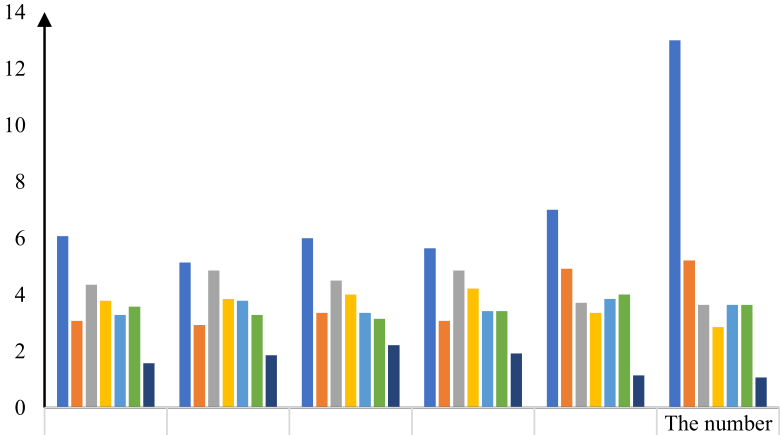
Once-again, to analyze the efficiency analysis of the experimental results obtained by different methods, Wilcoxon's signed-rank test [45] and Friedman's test [46] with a significance level of 0.05, were performed. Table 21 summarizes the Wilcoxon's signed-rank test results regarding ACC × PDF metric.

Based on the obtained results from Table 21, the proposed approach shows a significant difference from the compared methods on all datasets.

The Friedman's test was also applied to evaluate the performance of all compared methods concerning the ACC × PDF metric. This test having been performed, the *p*-value was equal to 1.38E-9, which indicates that the overall performance of the proposed method is

Table 20
The number of reported best results for different methods in terms of each performance metric on different datasets.

	All-features	GA-SVM	cGA-FS	WFACOFS	MOEDAFS	RSVM-SBS	The proposed Method
Accuracy	1	5	0	2	1	2	**8**
Precision	1	5	0	2	1	2	**8**
Recall	1	6	0	1	3	1	**7**
F1-score	1	5	0	1	2	2	**8**
ACC × PDF	0	0	0	1	1	0	**12**
# of Selected Features	0	0	0	1	2	0	**13**



	Accuracy	Precision	Recall	F1-score	ACC×PDF	The number of Selected Features
All-features	6.07	5.14	6	5.64	7	13
GA-SVM	3.07	2.92	3.35	3.07	4.92	5.21
cGA-FS	4.35	4.85	4.5	4.85	3.71	3.64
WFACOFS	3.78	3.85	4	4.21	3.35	2.85
MOEDAFS	3.28	3.78	3.35	3.42	3.85	3.64
RSVM-SBS	3.57	3.28	3.14	3.42	4	3.64
The proposed Method	1.57	1.85	2.21	1.92	1.14	1.07

All-features GA-SVM cGA-FS WFACOFS
MOEDAFS RSVM-SBS The proposed Method

Fig. 3. The average ranks of different methods in terms of each performance metric on different datasets.

Table 21
P-value of Wilcoxon signed-rank test between the proposed approach and each other methods in terms of ACC × PDF.

Method	*p*-value
All-features	0.0002
GA-SVM	0.0002
cGA-FS	0.0002
WFACOFS	0.0017
MOEDAFS	0.0002
RSVM-SBS	0.0002

significantly better than the other methods.

Finally, it is interesting to compare our proposed methos with a correlation-aware feature selection method. In [49], a correlation-aware feature selection method (CBFS) was proposed which further improves feature selection speed while preserving classification accuracy. The authors claimed that their method shortens computations compared to the pairwise feature selection method and produces classification errors that are not worse than those produced by existing methods. Thus, we decided to compare our method with this technique. Table 22 compares the best classification accuracy of this method with the proposed method testing on three datasets which were used in [49]. All datasets are available at the UCI repository of machine learning databases [44]. The mushroom dataset has 8124 samples with 22 features belonging to two classes. The waveform dataset contains 21 features while the waveform with noise dataset has additional 19 all-noise features. Both of them contain 5000 samples with three classes. The number of training samples is denoted by $|V_{train}|$.

Table 22
The average accuracy (%) metric for different methods on different datasets.

	$\lvert V_{train}\rvert$	CBFS with C4.5	CBFS with MLP	The proposed Method
Mushroom	4000	**99.60**	88.60	90.26
Waveform	3500	88.40	84.02	**89.46**
Waveform with noise	3500	89.27	83.63	**90.80**

As shown in Table 22, in two of datasets the proposed method reports better results.

4.5. Guiding technique analysis

A simple way to determine the number of selected features is to use a greedy approach. Thus, in each iteration, we randomly draw *s* and select

the best s features according to the measure used in line 7 of the pseudocode. However, with this technique, the number of selected features will not converge to an optimal value. Fig. 4 illustrates the average number of selected features for the *winner* after ten separate runs of the algorithm with 500 function evaluations for the Hill-valley dataset.

As shown in Fig. 4, the number of selected features did not converge after 500 function evaluations. In the following, we introduce our guiding strategy technique that would solve this problem and can converge to a stable and good solution in the same number of function evaluation.

As discussed in Section 3.5, our introduced guiding technique ensures that the number of features for each individual follows the chi-square distribution with d degrees of freedom, where d is the number of *winner*'s features. This can increase the convergence speed of the algorithm due to the limitation on the number of features for each individual. It can also help the proposed method to select fewer features. Similarly, we repeat the experiment with 500 function evaluations. However, after 100 function evaluations, the number of selected features converged. Thus, Fig. 5 shows the average number of selected features for the *winner* after ten separate runs of the algorithm with 100 function evaluations. We performed this experiment on the Hill-valley dataset with 100 features. Since the other datasets showed similar trends, reporting them was ignored.

As shown in Fig. 5, the proposed method tends to select the minimum number of features almost at the beginning of the algorithm.

4.6. Time complexity

In this section, we discuss the time complexity of the proposed method. Based on our pseudocode of the proposed algorithm, the time complexity analysis of the proposed method should be studied in five steps. Generally, the time complexity of the entire evolution process in evolutionary-based FS methods mainly depends on the fitness function [50]. In the first step, we initialized our variables and data structures. Step 2 generates two individuals using our introduced probabilities. There are two main loops in this step corresponding to each individual. The main operation in this step is calculation of the conditional probabilities using Eqs. (2) and (3). Let C be the cost of one-time calculation of each probability. Thus, the time complexity of step 2 is $O(\max(s_a, s_b) \times C \times |\overline{X}|)$, where X is equal to A if s_a is greater than s_b or equal to B, otherwise. It should be noted that $|\overline{X}|$ is the number of not-selected features. Since s_a, s_b, *and* $|\overline{X}|$ is less than or equal to n, where n is the number of features, the time complexity will be $O(C \times n^2)$. In the next step, the fitness of each individual is calculated. The SVM classifier is used for this purpose. As mentioned before, the time complexity of the evolution process mainly depends on the calculation of the fitness function. Therefore, the computation time in this step is dataset-dependent. The fourth step updates the vector SV with size n and the matrix IM with size $n \times n$. Hence, the time complexity of this step is $O(n^2)$. Finally, in the last step, the parameter d is updated.

For the last experiment, we investigated the convergence speed of the proposed algorithm and compared it with other population-based optimization FS methods. Fig. 6 shows the number of function evaluations to achieve the best average accuracy in the evolution process. The results are reported after ten separate runs of the algorithm with 500 function evaluations for the Heart dataset. It should be noted that the other datasets also had similar behavior and so we ignored reporting them.

As shown in Fig. 6, the proposed method converged to a higher accuracy with fewer number of function evaluations. The proposed method reported the best answer in 73-th function evaluation, but GA-SVM, cGA-FS.

As we know, there is a trade-off between computation speed and classification accuracy in each FS method. The main advantage of our method is that it considers interdependencies between features but it does not perform in a pairwise manner for selecting features. In pairwise feature selection strategy all of the features are evaluated in a pairwise manner. The procedure begins with an empty set of selected features. Then, features are added in pairs. In each selection step, the pair that maximizes a predefined criterion together with the already selected features is added. A pairwise selection strategy has higher time-complexity but it can take into account any possible interdependencies between features, which leads to higher classification accuracy [49]. Thus, this selection strategy has a quadratic complexity. However, in our proposed method, for each individual a and b, in each iteration, only one feature is selected based on our introduced probabilistic model (Eqs. (1)–(3)) using roulette wheel mechanism. The introduced probabilities were calculated using the goodness of selecting

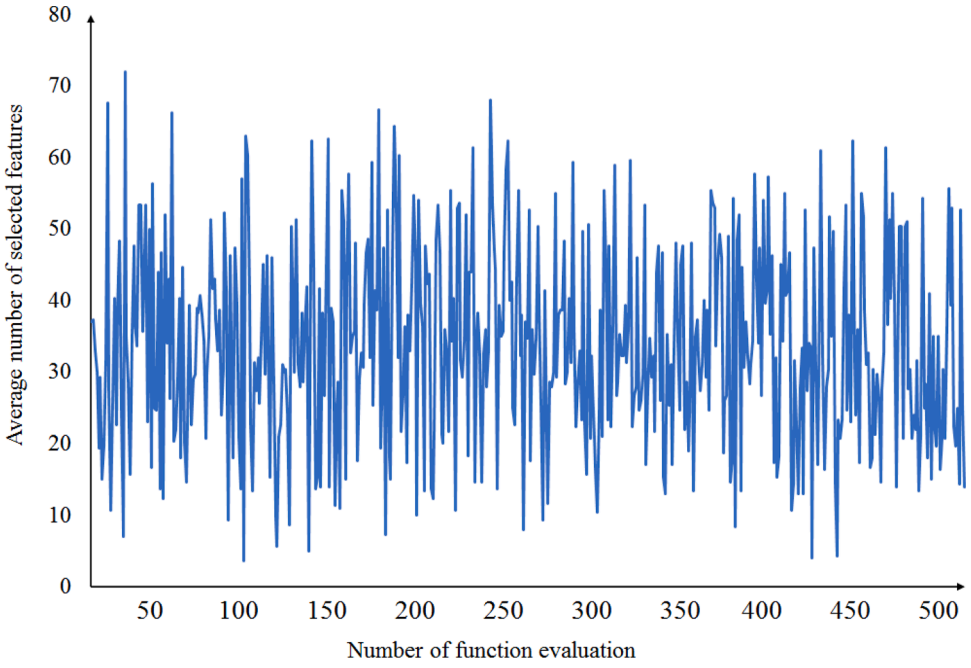


Fig. 4. The average number of selected features of the *winner*, determined by the greedy approach for the Hill-valley dataset.

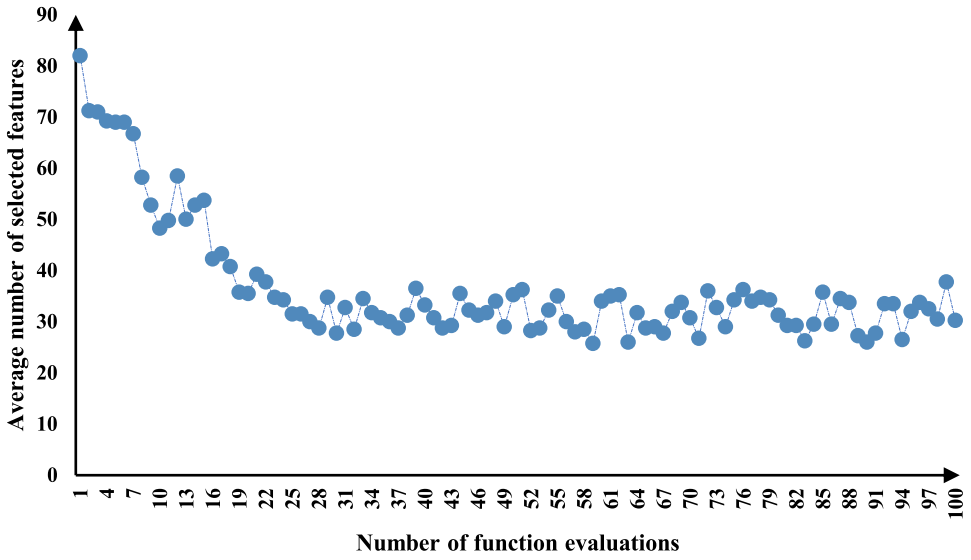


Fig. 5. The average number of selected features of the *winner*, determined by the chi-square distribution for the Hill-valley dataset.

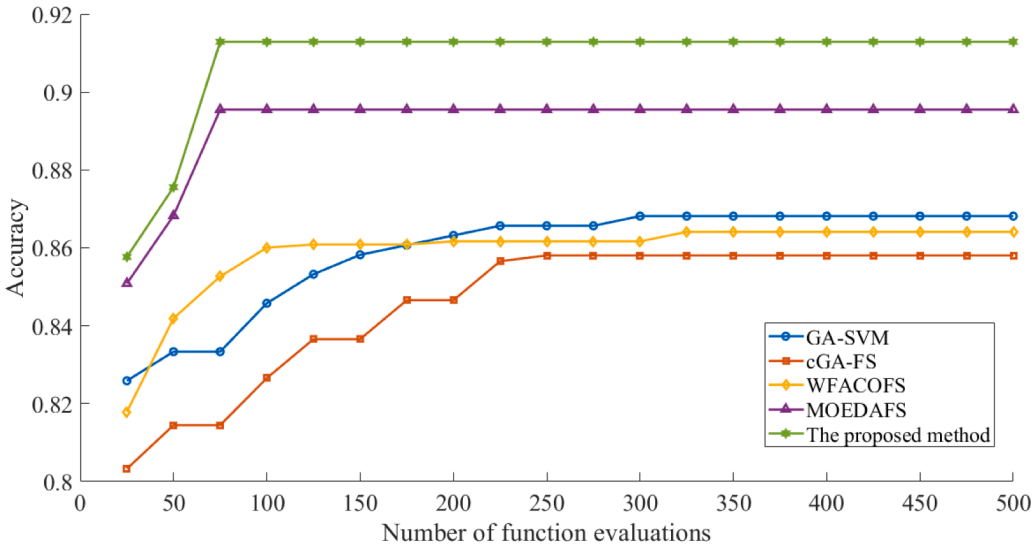


Fig. 6. The number of function evaluations to achieve the best average accuracies for the Heart dataset.

each feature alone (using the vector *SV*) and the goodness of selecting each feature along with others (using the matrix *IM*). Thus, we do not perform pairwise selection. In fact, we compare each pair of the winner and the loser for updating the matrix IM. Thus, as presented in experimental results, the proposed method not only improves the classification accuracy, but also it converges to the optimum solution with fewer function evaluations.

5. Conclusion and future works

Feature selection is one of the critical preprocessing steps in each machine learning application. It tries to find the optimal subset of informative features and consequently remove irrelevant ones. The main advantages of FS are reducing the time complexity and the model building cost, preventing over-fitting, increasing the generalizability of the trained classifier, and increasing its accuracy. However, an FS task suffers from a large search space and correlated features that severely affects its performance. In recent years, there have been several studies in the literature as for each of these challenges. Among them, evolutionary algorithms have provided more promising results, dealing with the FS problem. In this paper, we proposed a correlation-aware FS approach based on the estimation of distribution algorithms (EDAs), which is categorized in population-based optimization methods. The EDA methods use an explicit probability distribution to generate new candidate solutions, and a sequence of probabilistic model updates is utilized to optimize the problem. The main contribution of our proposed method is to consider both the importance of each feature alone and the interaction between features. To do this, a conditional probability scheme that examines the joint probability distribution of selecting two features is considered. The other advantage of the proposed method is that it generates only two individuals in each iteration. However, the obtained results in the experiments justified that this does not result in weak search in the entire space. The generated individuals compete in each iteration and evolve during the algorithm to find the best solution. Finally, we proposed a guiding mechanism that ensures that the number of features for each individual does not exceed the value determined by the chi-square distribution. This can directly increase the convergence speed of the algorithm due to the limitation on the number of features for each individual. The experimental results on synthetic and real-world datasets and the statistical analysis proved that the proposed

method was quite successful in both considering the correlated features and increasing the classification accuracy. It should be mentioned that our proposed method does not make any assumption about datasets, being able to deal with any feature selection for classification problems. For example, for a given imbalanced dataset, it is necessary to firstly balance it and then run our algorithm on the balanced dataset. Moreover, since our proposed method is an evolutionary-based algorithm that uses only two individuals in each generation, it requires more time to converge for high-dimensional data. As future works, it is interesting to extend our method in order to deal with multi-objective problems, receiving the advantages of the proposed method in more real-world applications. Moreover, one limitation about the proposed method is dealing with high-dimensional data. This limitation of the proposed method stems from the $n \times n$ interaction matrix (*IM*), where n is the number of features and increasing n would result in increasing the time complexity as well as the space required to deal with it. Thus, it can be considered as a direction for future studies. Our proposed has not been as good as other methods in some datasets. We should try to improve its performance to produce better results on more applications and datasets. Finally, although our method is originally developed in the case that we generate only two individuals in each iteration, it can be extended to all evolutionary-based FS methods as an intriguing future study.

Declaration of Competing Interest

None.

References

[1] Y. Cao, T.A. Geddes, J.Y.H. Yang, P. Yang, Ensemble deep learning in bioinformatics, Nat. Mach. Intell. 2 (9) (2020) 500–508, https://doi.org/10.1038/s42256-020-0217-y.

[2] I.F. Kilincer, F. Ertam, A. Sengur, Machine learning methods for cyber security intrusion detection: datasets and comparative study, Comput. Netw. 188 (Apr. 2021), 107840, https://doi.org/10.1016/j.comnet.2021.107840.

[3] M. Dong, L. Yao, X. Wang, B. Benatallah, C. Huang, X. Ning, Opinion fraud detection via neural autoencoder decision forest, Pattern Recognit. Lett. 132 (Apr. 2020) 21–29, https://doi.org/10.1016/j.patrec.2018.07.013.

[4] Z.K. Senturk, Early diagnosis of Parkinson's disease using machine learning algorithms, Med. Hypotheses 138 (2020), 109603, https://doi.org/10.1016/j.mehy.2020.109603.

[5] N. Maleki, Y. Zeinali, S.T.A. Niaki, A k-NN method for lung cancer prognosis with the use of a genetic algorithm for feature selection, Expert Syst. Appl. 164 (2021), 113981, https://doi.org/10.1016/j.eswa.2020.113981.

[6] S. Lalmuanawma, J. Hussain, L. Chhakchhuak, Applications of machine learning and artificial intelligence for Covid-19 (SARS-CoV-2) pandemic: a review, Chaos Solitons Fractals 139 (2020), 110059, https://doi.org/10.1016/j.chaos.2020.110059.

[7] J. Kim, J. Kang, M. Sohn, Ensemble learning-based filter-centric hybrid feature selection framework for high-dimensional imbalanced data, Knowl. Based Syst. 220 (2021), 106901, https://doi.org/10.1016/j.knosys.2021.106901.

[8] B. Xue, M. Zhang, W.N. Browne, X. Yao, A survey on evolutionary computation approaches to feature selection, IEEE Trans. Evol. Comput. 20 (4) (2016) 606–626, https://doi.org/10.1109/TEVC.2015.2504420.

[9] B. Tran, B. Xue, M. Zhang, Variable-length particle swarm optimization for feature selection on high-dimensional classification, IEEE Trans. Evol. Comput. 23 (3) (2019) 473–487, https://doi.org/10.1109/TEVC.2018.2869405.

[10] S. Sayed, M. Nassef, A. Badr, I. Farag, A Nested Genetic Algorithm for feature selection in high-dimensional cancer Microarray datasets, Expert Syst. Appl. 121 (2019) 233–243, https://doi.org/10.1016/j.eswa.2018.12.022.

[11] C.B. Gokulnath, S.P. Shantharajah, An optimized feature selection based on genetic approach and support vector machine for heart disease, Clust. Comput. 22 (6) (2019) 14777–14787, https://doi.org/10.1007/s10586-018-2416-4.

[12] F. Amini, G. Hu, A two-layer feature selection method using Genetic Algorithm and Elastic Net, Expert Syst. Appl. 166 (2021), 114072, https://doi.org/10.1016/j.eswa.2020.114072.

[13] M. Amoozegar, B. Minaei-Bidgoli, Optimizing multi-objective PSO based feature selection method using a feature elitism mechanism, Expert Syst. Appl. 113 (2018) 499–514, https://doi.org/10.1016/j.eswa.2018.07.013.

[14] R.K. Huda, H. Banka, Efficient feature selection and classification algorithm based on PSO and rough sets, Neural Comput. Appl. 31 (8) (2019) 4287–4303, https://doi.org/10.1007/s00521-017-3317-9.

[15] Y. Xue, T. Tang, W. Pang, A.X. Liu, Self-adaptive parameter and strategy based particle swarm optimization for large-scale feature selection problems with multiple classifiers, Appl. Soft. Comput. J. 88 (Mar. 2020), 106031, https://doi.org/10.1016/j.asoc.2019.106031.

[16] X. fang Song, Y. Zhang, D. wei Gong, X. yan Sun, Feature selection using bare-bones particle swarm optimization with mutual information, Pattern Recognit. 112 (Apr. 2021), 107804, https://doi.org/10.1016/j.patcog.2020.107804.

[17] H. Peng, C. Ying, S. Tan, B. Hu, Z. Sun, An improved feature selection algorithm based on ant colony optimization, IEEE Access 6 (2018) 69203–69209, https://doi.org/10.1109/ACCESS.2018.2879583.

[18] R. Joseph Manoj, M.D. Anto Praveena, K. Vijayakumar, An ACO–ANN based feature selection algorithm for big data, Clust. Comput. 22 (2) (2019) 3953–3960, https://doi.org/10.1007/s10586-018-2550-z.

[19] W. Ma, X. Zhou, H. Zhu, L. Li, L. Jiao, A two-stage hybrid ant colony optimization for high-dimensional feature selection, Pattern Recognit. 116 (Aug. 2021), 107933, https://doi.org/10.1016/j.patcog.2021.107933.

[20] R.J. Kuo, S.B.L. Huang, F.E. Zulvia, T.W. Liao, Artificial bee colony-based support vector machines with feature selection and parameter optimization for rule extraction, Knowl. Inf. Syst. 55 (1) (2018) 253–274, https://doi.org/10.1007/s10115-017-1083-8.

[21] Y. Zhang, S. Cheng, Y. Shi, D. Gong, X. Zhao, Cost-sensitive feature selection using two-archive multi-objective artificial bee colony algorithm, Expert Syst. Appl. 137 (2019) 46–58, https://doi.org/10.1016/j.eswa.2019.06.044.

[22] X. Wang, Y. Zhang, X. Sun, Y. Wang, C. Du, Multi-objective feature selection based on artificial bee colony: an acceleration approach with variable sample size, Appl. Soft Comput. 88 (2020), 106041, https://doi.org/10.1016/j.asoc.2019.106041.

[23] M.A. El Aziz, A.E. Hassanien, Modified cuckoo search algorithm with rough sets for feature selection, Neural Comput. Appl. 29 (4) (2018) 925–934, https://doi.org/10.1007/s00521-016-2473-7.

[24] A.C. Pandey, D.S. Rajpoot, M. Saraswat, Feature selection method based on hybrid data transformation and binary binomial cuckoo search, J. Ambient Intell. Humaniz. Comput. 11 (2) (2020) 719–738, https://doi.org/10.1007/s12652-019-01330-1.

[25] M. Hauschild, M. Pelikan, An introduction and survey of estimation of distribution algorithms, Swarm Evol. Comput. 1 (3) (2011) 111–128, https://doi.org/10.1016/j.swevo.2011.08.003.

[26] H. Karshenas, R. Santana, C. Bielza, P. Larrañaga, Multiobjective estimation of distribution algorithm based on joint modeling of objectives and variables, IEEE Trans. Evol. Comput. 18 (4) (2014) 519–542, https://doi.org/10.1109/TEVC.2013.2281524.

[27] F. Tan, X. Fu, Y. Zhang, A.G. Bourgeois, A genetic algorithm-based method for feature subset selection, Soft Comput. 12 (2) (2008) 111–120, https://doi.org/10.1007/s00500-007-0193-8.

[28] S. Oreski, G. Oreski, Genetic algorithm-based heuristic for feature selection in credit risk assessment, Expert Syst. Appl. 41 (4) (2014) 2052–2064, https://doi.org/10.1016/j.eswa.2013.09.004. Part 2.

[29] F. Moslehi, A. Haeri, A novel hybrid wrapper–filter approach based on genetic algorithm, particle swarm optimization for feature subset selection, J. Ambient Intell. Humaniz. Comput. 11 (3) (2020) 1105–1127, https://doi.org/10.1007/s12652-019-01364-5.

[30] A.K. Das, S. Das, A. Ghosh, Ensemble feature selection using bi-objective genetic algorithm, Knowl. Based Syst. 123 (May 2017) 116–127, https://doi.org/10.1016/j.knosys.2017.02.013.

[31] J. Kennedy, R. Eberhart, Particle swarm optimization 4 (1995) 1942–1948, https://doi.org/10.1109/ICNN.1995.488968.

[32] P. Moradi, M. Gholampour, A hybrid particle swarm optimization for feature subset selection by integrating a novel local search strategy, Appl. Soft. Comput. 43 (2016) 117–130, https://doi.org/10.1016/j.asoc.2016.01.044.

[33] M. Dorigo and G. Di Caro, "Ant colony optimization: a new meta-heuristic," in *Proceedings of the 1999 Congress on Evolutionary Computation-CEC99 (Cat. No. 99TH8406)*, 1999, vol. 2, pp. 1470–1477, doi: 10.1109/CEC.1999.782657.

[34] M. Ghosh, R. Guha, R. Sarkar, A. Abraham, A wrapper-filter feature selection technique based on ant colony optimization, Neural Comput. Appl. 32 (12) (2020) 7839–7857, https://doi.org/10.1007/s00521-019-04171-3.

[35] D. Karaboga, An idea based on honey bee swarm for numerical optimization, in: Technical report-tr06, Erciyes University, Engineering Faculty, Computer Engineering Department, 2005.

[36] E. Hancer, B. Xue, M. Zhang, D. Karaboga, B. Akay, Pareto front feature selection based on artificial bee colony optimization, Inf. Sci. (Ny). 422 (2018) 462–479, https://doi.org/10.1016/j.ins.2017.09.028.

[37] S. Maza, M. Touahria, Feature selection for intrusion detection using new multi-objective estimation of distribution algorithms, Appl. Intell. 49 (12) (2019) 4237–4257, https://doi.org/10.1007/s10489-019-01503-7.

[38] G.R. Harik, F.G. Lobo, D.E. Goldberg, The compact genetic algorithm, IEEE Trans. Evol. Comput. 3 (4) (1999) 287–297.

[39] I. Inza, P. Larrañaga, B. Sierra, Feature Subset Selection by Estimation of Distribution Algorithms BT - Estimation of Distribution Algorithms: a New Tool for Evolutionary Computation, Springer US, Boston, MA, 2002, pp. 269–293. P. Larrañaga and J. A. Lozano, Eds.

[40] L. Zhang, K. Mistry, C.P. Lim, S.C. Neoh, Feature selection using firefly optimization for classification and regression models, Decis. Support Syst. 106 (2018) 64–85, https://doi.org/10.1016/j.dss.2017.12.001.

[41] M.A. Laamari, N. Kamel, A Hybrid Bat Based Feature Selection Approach for Intrusion Detection, BT- Bio-Inspired Computing - Theories and Applications (2014) 230–238, https://doi.org/10.1007/978-3-662-45049-9_38.

[42] M. Sabzekar, Z. Aydin, A noise-aware feature selection approach for classification, Soft. Comput. 25 (8) (2021) 6391–6400, https://doi.org/10.1007/s00500-021-05630-7.

[43] R. Katuwal, P.N. Suganthan, L. Zhang, Heterogeneous oblique random forest, Pattern Recognit. 99 (2020), 107078, https://doi.org/10.1016/j.patcog.2019.107078.
[44] D. Dua, C. Graff, Machine Learning Repository, Universityof California, Irvine, School of Information and Computer Sciences (2017). http://archive.ics.uci.edu/ml.
[45] F. Wilcoxon, Individual Comparisons by Ranking Methods, in: S. Kotz, N. L. Johnson (Eds.), Breakthroughs in Statistics, Springer Series in Statistics, Springer, New York, NY, 1992, pp. 196–202, https://doi.org/10.1007/978-1-4612-4380-9_16.
[46] M. Friedman, The use of ranks to avoid the assumption of normality implicit in the analysis of variance, J. Am. Stat. Assoc. 32 (200) (1937) 675–701, https://doi.org/10.1080/01621459.1937.10503522.
[47] A. Thakkar, R. Lohiya, Attack classification using feature selection techniques: a comparative study, J. Ambient Intell. Humaniz. Comput. 12 (1) (2021) 1249–1266, https://doi.org/10.1007/s12652-020-02167-9.
[48] M. Tavallaee, E. Bagheri, W. Lu, A.A. Ghorbani, A detailed analysis of the KDD CUP 99 data set, in: Procdeedings of the IEEE Symposium on Computational Intelligence for Security and Defense Applications, IEEE, 2009, pp. 1–6.
[49] K. Michalak, H. Kwasnicka, Correlation based feature selection method, Int. J. Bio-Inspired Comput. 2 (5) (2010) 319–332, https://doi.org/10.1504/IJBIC.2010.036158.
[50] H. Tian, S.-C. Chen, M.-L. Shyu, Evolutionary programming based deep learning feature selection and network construction for visual data classification, Inf. Syst. Front. 22 (5) (2020) 1053–1066, https://doi.org/10.1007/s10796-020-10023-6.