Applied Soft Computing 43 (2016) 117–130

Contents lists available at ScienceDirect

Applied Soft Computing

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

A hybrid particle swarm optimization for feature subset selection by integrating a novel local search strategy

Parham Moradi*, Mozhgan Gholampour

Department of Computer Engineering, University of Kurdistan, Sanandaj, Iran

ARTICLE INFO

Article history:
Received 10 December 2013
Received in revised form 7 August 2015
Accepted 28 January 2016
Available online 4 February 2016

Keywords:
Feature selection
Local search
Correlation information
Particle swarm optimization

ABSTRACT

Feature selection has been widely used in data mining and machine learning tasks to make a model with a small number of features which improves the classifier's accuracy. In this paper, a novel hybrid feature selection algorithm based on particle swarm optimization is proposed. The proposed method called HPSO-LS uses a local search strategy which is embedded in the particle swarm optimization to select the less correlated and salient feature subset. The goal of the local search technique is to guide the search process of the particle swarm optimization to select distinct features by considering their correlation information. Moreover, the proposed method utilizes a subset size determination scheme to select a subset of features with reduced size. The performance of the proposed method has been evaluated on 13 benchmark classification problems and compared with five state-of-the-art feature selection methods. Moreover, HPSO-LS has been compared with four well-known filter-based methods including information gain, term variance, fisher score and mRMR and five well-known wrapper-based methods including genetic algorithm, particle swarm optimization, simulated annealing and ant colony optimization. The results demonstrated that the proposed method improves the classification accuracy compared with those of the filter based and wrapper-based feature selection methods. Furthermore, several performed statistical tests show that the proposed method's superiority over the other methods is statistically significant.

1. Introduction

In recent years, with the advance of science and technology, datasets with large numbers of features and relatively few patterns were produced. A large number of irrelevant and redundant features may significantly degrade the accuracy of learned models as well as reduce the learning speed of the models. This problem is known as curse of dimensionality in data mining methods and increases the computational complexity of building the model. Feature selection is one of the most feasible solutions to reduce the dimensionality of the datasets by selecting the most informative features and still retains sufficient information for the classification task. The main idea behind the feature selection is to choose a subset of salient features, by eliminating irrelevant features with little or no predictive information, as well as redundant features that are strongly correlated. On the other hand, this reduction helps to speed up the learning process, leads to a simple and understandable predictor model and avoids overfitting [1–3]. Feature selection has been successfully applied to many fields such as text categorization [4,5], face recognition [6,7], cancer classification [8,9], gene classification[10], finance [11,12], recommender systems[13] and customer relationship management [14].

To find the optimal feature subset one needs to enumerate and evaluate all the possible subsets of features. The entire search space contains all the possible subsets of features, which means that the search space size is 2^n where n is the number of the original features. Therefore, the problem of finding the optimal feature subset is a NP-hard problem [15,16]. Thus, evaluating the entire feature subset is computationally expensive and also impractical even for a moderate-sized feature set. Therefore, many feature selection algorithms involve heuristic or random search strategies to find the optimal or near optimal subset of features in order to reduce the computational time. These types of feature selection methods can be categorized into filter [17–26], wrapper [27–37], hybrid [33,38–42] and embedded [43–45] approaches. The filter approach looks for features that maximize a criterion which does not rely on any specific learning model. The wrapper approach utilizes a given learning algorithm to evaluate a candidate feature subset and thus the feature selection process is wrapped around the learning

* Corresponding author. Tel.: +98 8733668513.
E-mail addresses: p.moradi@uok.ac.ir (P. Moradi), mjgn.gholampour@gmail.com (M. Gholampour).

http://dx.doi.org/10.1016/j.asoc.2016.01.044
1568-4946/

model. Although the wrapper approach uses learning algorithms to evaluate a feature subset, it requires a high computational cost for high-dimensional datasets. On the other hand, the filter-based methods are typically faster than the wrapper ones, but due to the lack of a learning model in their search process, the quality of the final result will be low compared with those of the wrapper methods. Therefore, the goal of the hybrid-based methods is to use the computational efficiency of the filter model and the proper performance of the wrapper model. However, the hybrid model may suffer in terms of accuracy because the filter and wrapper models are considered as two separate steps. Finally, the embedded approach seeks to subsume feature selection as part of the model building process and is thus associated with a specific learning model.

The wrapper approach uses a predetermined learning model to evaluate features subsets [3]. Although this approach achieves a good result compared with that of the filter approach, it requires more computational resources. The wrapper-based methods can be classified into sequential and global search methods [15,45,46]. The sequential search methods in turn can be categorized into forward (SFS) and backward (SBS) search methods. The forward search method starts from an empty set of features and at each step a feature is added to the feature set to increase the classifier performance, while the backward search method starts from the full set of features and greedily removes a feature at each step according to the classifier performance. In other words, the salient (redundant) features are added (deleted) sequentially during the training of the classifier using the forward (backward) search strategy. A number of algorithms have been proposed in the sequential search strategy, the search processes of which are guided by specific learning models [28,30,47,48]. The sequential search strategies involve the local search rather than the global search; thus, these algorithms try to find solutions that range between sub-optimal and near optimal regions. Therefore, the sequential-based feature selection methods still suffer from a variety of problems, such as stagnation in local optima and high computational cost.

On the other hand, the global search methods apply randomness into their search strategy to explore a large portion of the solution space in order to better address the feature selection problems. Recently, metaheuristic algorithms have attracted a lot of attention due to their good performance in solving the feature selection problem. These algorithms include genetic algorithm (GA) [36,49,50], particle swarm optimization (PSO) [21,51–63], ant colony optimization (ACO) [10,18,24,26,33,64], simulated annealing (SA) [65–67] and bacterial foraging optimization (BFO) [68]. The attention of researchers upon GA is due to its simplicity while PSO and ACO have higher accuracy in similar tasks [69]. It has been shown that PSO is computationally less expensive and can converge more quickly compared with the other metaheurestic algorithms such as GA and genetic programming (GP). Moreover, the PSO is an easy-to-implement algorithm, has less adjustable parameters and is also computationally inexpensive in both speed and memory requirements. Therefore, the PSO has been used as an effective technique in many fields, including feature selection.

Up to now, several PSO-based feature selection methods have been proposed in the literature.

[21,51,56,70–72]. Although the particle swarm optimization has been shown as an effective approach for finding optimal (or near optimal) feature subsets, it suffers from several shortcomings. One of the problems with existing PSO-based feature selection methods is that they ignore the number of features in their search processes while they only emphasize minimizing the classification error rate. Another limitation is selecting similar features in the final feature subset. Moreover, the existing PSO-based approaches do not use correlation information of the features to guide the search process. Therefore, similar features have a high probability to be selected in the final subset, which reduces the classifier performance. To overcome these problems, in this paper, a novel hybrid particle swarm optimization is proposed for feature selection. The proposed method tries to hybridize the PSO by integrating new local search operations. Such operations embedded in the PSO fine-tune the search process for feature selection in an organized fashion. The proposed method called HPSO-LS (i.e. hybridized PSO with local search operations) uses correlation information to guide the search process in PSO in such a way that relatively less correlated (dissimilar) features are selected with a high probability than more correlated (similar) features. In the proposed method, the correlation-based search strategy is used as a local search operation in the PSO. Moreover, HPSO-LS selects the reduced number of salient features by using a specific subset size determination scheme. This scheme works upon a bounded region and tries to provide a subset with a smaller number of features. Such utilizations ultimately lead to a significant performance gain for the feature selection in HPSO-LS.

The rest of this paper is organized as follows; in Section 2, related works on feature selection are reviewed, and the basic PSO method and the feature selection task are also discussed. Moreover, details about the proposed method are introduced in Section 3. In Section 4, the proposed algorithm is compared with the other existing feature selection methods. Finally, Section 5 summarizes the present study.

2. Background and related works

2.1. Related works

Feature selection is a fundamental research topic in data mining with a long history since the 1970s. The main goal of the feature selection is to select a subset of features, by removing redundant features that are strongly correlated as well as irrelevant features with little or no predictive information. To find the optimal feature subset, it is required to enumerate and evaluate all the possible subsets of the features. The entire search space contains all the possible subsets of features, meaning that the search space size is 2^n where n denotes the number of original features. Therefore, evaluating the entire feature subset would be computationally expensive, time consuming and also impractical even for a moderate-sized dataset. Thus, the final solution should be found in a feasible computational time with a reasonable trade-off between the quality of the found solution and time–space cost. To this end, many feature selection methods which employed random search strategies were proposed in order to reduce the computational time.

The feature selection methods can be classified into four categories including filter, wrapper, embedded and hybrid models. The filter approach applies a statistical analysis to a feature set for solving the feature selection problem without utilizing any learning models. Therefore, the methods in this approach are typically fast. The filter-based methods can be classified into univariate and multivariate methods. In the univariate methods, each feature is evaluated independently according to a specific criterion. This means that each feature is considered separately, thereby ignoring feature dependencies, which may lead to worse classification. Several univariate methods have been proposed in the literature including Information gain (IG) [73], Gain ratio [74], Term variance (TV) [75], Gini index (GI) [76], Laplacian Score (L-Score) [77] and Fisher Score (F-Score) [78]. On the other hand, in the multivariate methods, to evaluate the relevance of the features their dependencies are taken into consideration. Minimal-redundancy-maximal-relevance (mRMR) [79], Random subspace method (RSM) [43], Relevance-redundancy feature selection (RRFS) [80], UFSACO [23], RRFSACO[24], GNCC[25] and GCACO[26] are well-known multivariate feature selection methods.

In the wrapper model, a given learning algorithm is used to evaluate subsets of features through the search process. In other words, the wrapper model is an iterative search process such that the results of the learning algorithm at each iteration are used to guide the search process [2]. The wrapper-based methods include the interaction with the learning algorithm and thus they outperform filter-based methods in term of prediction accuracy. However, these methods continuously use the learning algorithm in the search process and they are computationally more expensive, especially for high-dimensional datasets. Generally, the wrapper-based methods can be classified into greedy and random search approaches [15,46]. The greedy search approach is based on the hill-climbing algorithm in which a single feature is added or removed iteratively in a greedy way. Sequential backward selection and sequential forward selection are two well-known greedy search methods [15]. On the other hand, the random search approach applies randomness into its search strategy to explore a large portion of the solution space. Examples of random methods include ant colony optimization (ACO) [64], particle swarm optimization (PSO) [21], genetic algorithm (GA) [81,82], random mutation hill-climbing [83,84], simulated annealing (SA) [66] and Artificial bee colony (ABC) [85,86].

Among the many existing metaheuristic methods, GA, PSO and ACO are widely used for the feature selection problem. The attention of researchers upon GA is due to its simplicity while PSO and ACO have higher accuracy in similar tasks [69,87,88]. In the ACO algorithm, a number of ants are used to construct a feature subset based on the pheromone update and heuristic information. In most cases, the heuristic information in ACO is determined by applying field-based methods and the pheromone value is measured by using a learning model [10,18,23,24,26,33,64]. The ACO algorithm offers better accuracy because of its robustness but suffers from powerless rules for pheromone update and heuristic information. On the other hand, compared with GA, PSO has some attractive characteristics. It has memory, so knowledge of good solutions is retained by all particles; whereas in GA, previous knowledge of the problem is destroyed once the population changes. It has constructive cooperation between particles to share their information [58,59,89]. Moreover, the PSO is an easily implemented algorithm, has less adjustable parameters and is also computationally inexpensive in both speed and memory requirements. Various works [58,59,61,89–91] show that particle swarm optimization is equally well suited or even better than genetic algorithms for solving global optimization problems [58,91]. Therefore, PSO has been used as an effective technique in many fields, including feature selection [2,21,52,54,92].

The particle swarm optimization is a powerful swarm-based metaheuristic method, proposed by Kennedy and Eberhart in 1995 [89]. PSO is motivated by social behaviors such as bird flocking and fish schooling. The PSO method has recently gained more attention for solving the feature subset selection problem [2,57,92–94]. Wang et al. [21] proposed a PSO-based method to find optimal feature subsets based on rough sets and concluded that the PSO-based methods would be more efficient than GA-based methods for the feature selection problem. Moreover, Talbi et al. [70] proposed a hybrid feature selection method called GPSO based on PSO and GA to classify high-dimensional microarray data. Furthermore, another hybrid method has been proposed in Lin et al. [71] by combining the PSO with support vector machine (SVM). In addition, in Li Chuang et al. [51] introduced a binary optimization algorithm, called catfish BPSO, by employing the catfish effect to improve the performance of the binary PSO for the feature selection problem. In Jiang et al. [95], the PSO was hybridized with the artificial fish swarm algorithm to overcome the local optimization of the PSO and improve its searching ability. Likewise, in Chang et al. [96], a hybrid model was developed by integrating a case-based reasoning approach and a particle swarm optimization model for medical data classification. In addition, Unler et al. [2] proposed a hybrid method called maximum relevance minimum redundancy PSO (mr2PSO), which integrated the mutual information-based filter model within the PSO-based wrapper model. Another hybrid approach was proposed by Inbarani [94] to handle the selection of appropriate features for medical diagnosis problems. They proposed two supervised methods, PSO-based Relative Reduct (HPSO-RR) and PSO-based Quick Reduct (HPSO-QR), for feature reduction in order to increase the efficiency of the features selection method. Moreover, Huang and Dun [57] proposed a PSO–SVM model that hybridized the PSO and support vector machines (SVM) to improve the classification accuracy with a small and appropriate feature subset. Furthermore, Xue et al. [93] proposed three initialization strategies and several updating mechanisms in PSO to develop feature selection approaches. The goal of the method was to select a smaller number of features as well as to achieve better classification performance. Recently, in Xue et al. [56] a PSO-based feature selection method was proposed to select a smaller number of features and achieve better classification performance than using all features. So these goals were obtained by developing forward and backward word selection, and new pbest and gbest updating mechanisms.

On the other hand, the hybrid model is a combination of the filter and the wrapper models and attempts to take advantage of both approaches. The hybrid model mainly focuses on combining the filter and the wrapper-based methods to achieve the best possible performance with a particular learning algorithm with time complexities similar to those of the filter-based methods. Moreover, the embedded model tries to include the feature selection as a part of the classifier training process, like inherently binary decision tree classifiers do. In other words, the feature selection process is embedded into the training of the learning algorithm.

2.2. Feature selection in classification

Supervised Learning algorithms generate a function from labeled training data that maps inputs to desired outputs. For example, in a classification problem, the learner approximates a function to map a vector into its corresponding class using input–output examples. In supervised learning, each example is a pair which consists of an input object and a desired output value. These methods are also known as learning with teacher because the training examples are labeled by human experts. Artificial neural networks, decision trees, Bayesian networks, support vector machines and learning automata are well known supervised learning algorithms [97].

The main goal of the feature selection is to reduce the number of features by eliminating redundant features or by selecting the most informative ones to improve the generalization capability of the classifiers. The important benefits of the features selection are reducing the amount of data required for the learning process, reducing the memory requirements for storage of data, improving computation time and increasing speed and accuracy of classification. Feature selection is an important preprocess step for those of datasets with large numbers of features.

The dataset can be defined as $s = \left\{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\right\}$ where $x_i \cong [x_{i1}, x_{i2}, \ldots, x_{id}]$ is a multi-dimensional vector sample, d denotes the number of features, n is the number of samples and $y_i \in \gamma$ denotes the label of the sample x_i. Furthermore, the main goal of the supervised learning is to approximate a function $f: \chi \mapsto \gamma$ to predict the class label for a new input vector x_i.

In the learning problems, choice of the optimization method is very important to reduce the dimensionality of the feature space and to improve the convergence speed of the learning model. Thus, an optimization method would be preferred that can handle a

large number of features. To this end, the PSO global optimization method was chosen because it has been shown through simulation that PSO offers better results compared with those of the other stochastic optimization methods [98].

2.3. Particle swarm optimization

The particle swarm optimization (PSO) algorithm is a stochastic optimization method in the global search methods family. Similar to the GA and evolutionary algorithm (EA), PSO is a search process based on the idea of swarm intelligence in biological populations, which searches for the global optima by updating its generations. The PSO algorithm was originally developed by Kennedy and Eberhart [89] in 1995 and is based on the bird's flocking or fish schooling paradigm. Each potential solution is called a particle, and the set of particles in any iteration is called a population. The particles fly through a multi-dimensional search space with their corresponding velocities which are updated by the previous best performance of the particle and its neighbors. The first population is typically initialized using a random number generator to spread the particles uniformly in the search space.

Suppose that D represents the dimension of the search space, $x_{id}(t)$ represents the position of the i-th particle at d-th dimension and $v_i(t)$ is the velocity of the i-th particle. The best previously visited position (up to time t) of the i-th particle is represented by x_i^{best} and the global best position of the swarm is denoted by x_g^{best}. The particle's velocity and its new position are updated as follows:

$$\bar{v}_{id}(t+1) = v_{id}(t) + c_1.r_1.\left(x_i^{best} - x_{id}(t)\right) v_i(t) + c_2.r_2.\left(x_g^{best} - x_{id}(t)\right) \quad (1)$$

where c_1 and c_2 are learning factors, normally set as $c_1 = c_2 = 2$ and r_1 and r_2 are random numbers between 0 and 1. The position of each particle is modified by adding its velocity to the current position.

$$x_{id}(t+1) = x_{id}(t) + v_{id}(t+1),\ i = 1, 2, \ldots, N,\ d = 1, 2, \ldots, D \quad (2)$$

In this work, the binary PSO was used since the position of each individual particle can be given in binary form (0 or 1), that shows whether a feature needs to be selected or not. The changes in particle velocity can be explained as changes in the probability of finding the particle in one state [99]. PSO has poor search ability in the near local optimum region, and premature convergence seems to be difficult to avoid [51,52].

2.4. k-nearest neighbor classifier

In pattern recognition, the k-nearest neighbors algorithm (k-NN) is a method used for classification and regression. The input of the k-NN algorithm is the training examples, and the output is a class membership. An object is classified by a majority vote of its k closest neighbors according to some distance or similarity function, where k is a positive integer, typically small, and is a user-defined constant. If $k = 1$, then the object is simply assigned to the class of that single nearest neighbor [98]. The k-NN is a type of instance-based learning and is among the simplest of all machine learning algorithms. The training examples are vectors in a multi-dimensional feature space, each associated with a class label. Unlike many pattern recognition algorithms, instance-based learners do not abstract any information from the training data during the learning phase and that is, at the time of classification. In the classification phase, an unlabeled vector is classified by calculating the distance between this vector and all training examples, then assigning the majority label of the k training samples nearest to that unlabeled vector. A commonly used distance metric is the Euclidean distance.

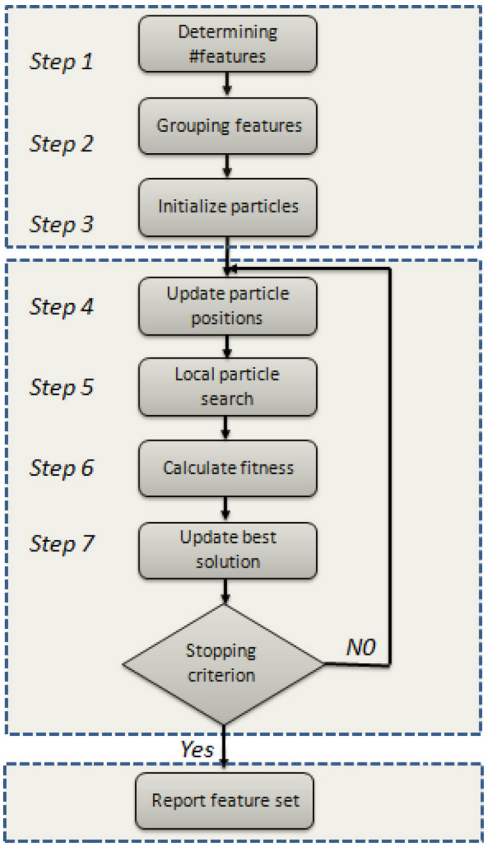

Fig. 1. Flowchart of the proposed HPSO-LS algorithm.

3. Proposed method

Although PSO provides a global search strategy for find a better solution in the feature selection task, it is infected with two shortcomings, premature convergence and weakness in fine-tuning near local optimum points [100]. To overcome such weaknesses, a novel hybrid feature selection method based on PSO called HPSO-LS was proposed. In the HPSO-LS, the final feature subset is selected through seven steps. In the first step, the size of the feature subset is determined automatically. Then in the second step, all the features are classified into similar and dissimilar groups using correlation information of the features. Then through the steps three to eight, the binary PSO method is hybridized with a specific local search strategy which considers the local information of the features into the search process. In the third step, the predetermined numbers of particles are created. These particles are moved to their new positions in step 4 according to their local best positions and the global best of the swarm. Next, in step 5, each particle searches in its local area considering the features correlation information. Continuously in the sixth step, the fitness of each particle is calculated and then in step 7, the local and the global best particles are replaced with those of the previous ones. Moreover, steps 4–7 are repeated until the stopping criterion is satisfied, otherwise the algorithm is stopped and the best feature set is reported. Fig. 1 shows the HPSO-LS steps and the additional details are described in the corresponding sections.

3.1. Step 1- Determining the number of features

The subset size determination mechanism employs a probabilistic random function that tries to provide a random number for determining the number of selected features in a bounded region

[101]. To this end, a probabilistic formula (e.g. Eq. 4) is used to define the initial size of the feature subset ($sf \leq f$) [31]

$$l_{sf} = \frac{f - sf}{\sum_{i=1}^{l} (f - i)} \tag{4}$$

where f denotes the number of the original features in a given dataset, sf is the number of the selected features, l represents the difference between f and k (i.e. $l = f - sf$) and l_{sf} is the probability value of determining sf as an initial number of features in the proposed method. It is clear that l_{sf} is maximized when sf is minimized. The initial number of features (i.e. sf) is determined using the roulette wheel procedure based on the probability value l_{sf}. The sf is randomly selected in the range $[x, M]$ where $M = \varepsilon . f$ and x depends on a given dataset and generally is set to 3. Moreover, ε is an adjustable parameter that controls M. If ε is set to about 1, then M is close to the number of the original features f, so the search space becomes larger, which clearly causes the high computational cost. Thus, ineffective feature subsets might be generated.

It should be noted that such a scheme works upon a bounded region and tries to provide the size of the subset smaller in number. Moreover, the function of such a scheme is an automatic identification of the initial number of features rather than hand-tuning and it only depends on a value of ε that is determined by the user based on the knowledge of the dataset.

3.2. Step 2- Grouping of the features

In this step, the features are divided into similar and dissimilar groups. The goal of the grouping in HPSO-LS is to find relationships between features in which the most distinct and informative features can be distributed into the newly generated particles [47]. To this end, HPSO-LS uses the well-known Pearson correlation coefficient [102] to measure correlation between different features as follows:

$$c_{ij} = \frac{\sum_{k=1}^{m} (x_i(k) - \bar{x}_i)\left(x_j(k) - \bar{x}_j\right)}{\sqrt{\sum_{k=1}^{m} (x_i(k) - \bar{x}_i)^2} \sqrt{\sum_{k=1}^{m} (x_i(k) - \bar{x}_i)^2}} \tag{4}$$

where c_{ij} is the correlation coefficient between two features i and j, m is the number of the samples, $x_i(k)$ and $x_j(k)$ denote the values of the feature vectors i and j for the k-th sample, respectively, and $\bar{x}_i$ and $\bar{x}_j$ represent the mean values of x_i and x_j vectors over all of the m samples, respectively. According to Eq. (4), the correlation coefficient between two features computes the similarity between the features; thus, higher values mean that the two features have high similarity to each other. On the other hand, lower values indicate that the two features have low similarity. After computing the correlation coefficient for all possible combinations of features, the correlation value for feature i is calculated as follows:

$$\mathrm{cor}_i = \frac{\sum_{j=1}^{f} \left|c_{ij}\right|}{f - 1} \text{ if } \quad i \neq j \tag{6}$$

where f is the number of all features and c_{ij} denotes the Pearson correlation value between features i and j. A higher correlation value for a feature means that the feature has a high value of similarity to the other features, while a lower value means that the feature is more distinct among the others. In order to create two groups of features, HPSO-LS attempts to divide the original feature set into two equal groups. To this end, HPSO-LS sorts all of the features in ascending order according to their correlation values. Those in the first half of the features have the lowest correlation values and they are put into the dissimilar group called D, while the rest of the features have higher correlation values and they are included in the second group called the similar group, S. The features available in the dissimilar group D are less correlated than those in the similar group S.

3.3. Step 3- Initializing particles

In the proposed method, each particle is represented by a binary vector. The length of the vector is equal to the number of the original features. In this type of representation if the value of a cell in the vector is set to 1, it denotes that the corresponding feature is selected and when the value is set to 0, it means that the corresponding feature is not selected. On the other hand, in HPSO-LS, the desired number of features k is decided by the subset size determination step (i.e. step 1). Then, for each particle a velocity vector is generated by using a random float number generator. The length of the velocity vector is equal to the length of the particle vectors. Each cell of the velocity vector is set to a random value in the range of [01].

3.4. Step 4- Updating the particle positions

In PSO, each particle changes its position according to its velocity as follows (Eq. 7):

$$\begin{aligned} v_{id}(t+1) &= v_{id}(t) + c_1 . r_{i,1} . \left(x_i^{\text{best}} - x_{id}(t)\right) v_i(t) \\ &\quad + c_2 . r_{i,2} . \left(x_g^{\text{best}} - x_{id}(t)\right) \end{aligned} \tag{7}$$

where c_1 and c_2 are learning factors and are generally set to $c_1 = c_2 = 2$, $r_{1,i}$ and $r_{2,i}$ are random numbers in the range of [01], and $x_{id}(t)$ represents the position of the d-th dimension of the i-th particle and $v_i(t)$ denotes the velocity of the i-th particle. The best previously visited position (up to time t) of the i-th particle is represented by x_i^{best} and the global best position of the swarm is denoted by x_g^{best}.

It should be noted that if the sum of accelerations causes the velocity of that dimension to exceed $V_{\max}$, then the velocity of that dimension is limited to $V_{\max}$ according to the following equation (Eq. 8):

$$\begin{aligned} &\text{if } v_{id}(t+1) \notin (v_{\min}, v_{\max}) \text{ then } v_{id}(t+1) \\ &= \max\left(\min(v_{\max}, v_{id}(t+1)), v_{\min}\right) \end{aligned} \tag{8}$$

where $v_{\max}$ and $v_{\min}$ are user specific parameters (in this paper $v_{\max} = 4, v_{\min} = -4$). The particles take the binary values and their corresponding velocities define the probability of each bit to take the values 0 or 1. The position of a particle is changed based on the following equations:

$$\begin{aligned} &s(v_{id}(t+1)) = \frac{1}{1 + e^{-v_{id}}} \\ &\text{if } rand < s(v_{id}(t+1)) \text{ then } x_{id}(t+1) = 1 \\ &\qquad \text{else } x_{id}(t+1) = 0 \end{aligned} \tag{9}$$

The position of particles after updating is calculated by the function $s(v_{id}(t+1))$ (Eq. (9)). If $v_{id}(t+1)$ is larger than a random value, then its position value is represented by 1 (meaning the corresponding feature is selected for the next update). On the other hand, if $v_{id}(t+1)$ is smaller than a random number, then its position value is represented by 0 which means that its corresponding feature is not selected for the next update.

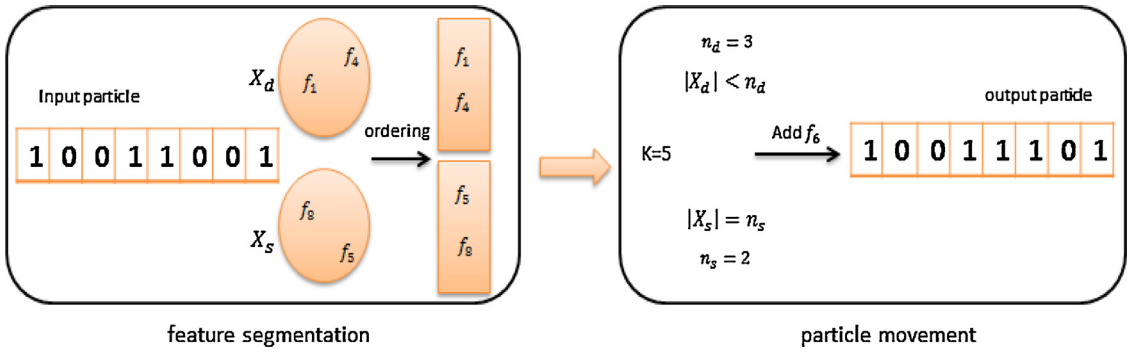

Fig. 2. Illustration of the local search strategy of the proposed method.

3.5. Step 5- Local search operations

In the proposed method, local search operations provide the special and general characteristics of a given dataset to the new position of a particle. Therefore, a classifier can learn necessary information about the dataset, which leads to better generalization ability [47]. Therefore, to carry out the local search in the proposed method, two steps are considered, including (1) feature segmentation, and (2) particle movement [101]. In these steps for a given particle the "*Add*" and "*Delete*" operators are employed to improve the local search of a particle. In short, a particle utilizes the *Add* operator to select a desired number of features, and the *Delete* operator is employed to remove a desired number of existing features from the position of the particle. In other words, the *Add* operator inserts the dissimilar features into the particle and the similar features are deleted from the particle using the *Delete* operator. Therefore, the main idea behind the local search is to select distinct features with the lowest correlation. The additional details of the local search strategy are illustrated with an example as shown in Fig. 2.

In the local search operator first of all the features selected by the particle are extracted. Therefore, in each practice, the algorithm distinguishes the number of 1-bits from a newly generated particle, e.g. 10011001, and puts them into a subgroup X in the form of feature numbers, i.e., $X = \{f_1, f_4, f_5, f_8\}$. Each element of X is then compared with those of D and. Then the elements of X are segmented into X_d and X_s. Specifically, X_d includes the dissimilar features that are included in D and their corresponding values in the newly generated particle [77], while X_s includes the similar features of X which are included in both X and S. Then in the next step all the features of X_d and X_s are rearranged in ascending order according to their correlation values (Fig. 2). Thus, the first feature and last feature of X_d and X_s denote the most distinct and the most similar feature, respectively.

Particle movement is the most important step of the local search process. In this step, it is needed to control the number of 1-bits in the newly generated particle. To this end, the numbers of similar and dissimilar features are given by calculating the values of n_s and n_d, respectively. Here, $n_s = \alpha . sf$ and $n_d = (1 - \alpha) . sf$ where α is a user specific parameter and sf is the size of the initial subset of the features which is determined in the first step. Actually, the number of features should be kept constant during the improvement of a particle by adding the most dissimilar features and deleting the least dissimilar ones. In other words, when the number of dissimilar features in the particle is smaller than n_d, then $(n_d - X_d)$ features in $(D\text{-}X_d)$ are added to the particle, otherwise $(X_d - n_d)$ features in X_d should be deleted from the particle. Moreover, if the number of similar features in the newly generated particle is larger than n_s, $(X_s\text{-}n_s)$ similar features in X_s are removed from the particle. On the other hand, when the number of similar features in the generated particle is smaller than n_s, $(n_s\text{-}X_s)$ features in $(S\text{-}X_s)$ are added to the particle.

3.6. Step 6- Calculating fitness

The proposed method (i.e. HPSO-LS) employs the k-NN classifier to evaluate a candidate feature subset solution. Before the evaluation process, first of all each feature is normalized being scaled between -1 and 1. The normalization process changes the dominating features with greater numeric values to those with bounded numeric ranges. Experimental results demonstrated that scaling the feature values can help improve the classification accuracy. Therefore, in this paper, a linear normalization method was used to scale the datasets as follows:

$$x^{\text{new}} = l + \left[(u - l) * \left(\frac{x - x_{\min}}{x_{\max} - x_{\min}}\right)\right] \tag{10}$$

where l and u are the lower bound and upper bound of the normalization process, respectively, $x_{\max}$ and $x_{\min}$ show the maximum and minimum values of the feature x, respectively. After the normalization process, the new dataset was extracted from the (normalized) original dataset with the features that were present in the solution of the particle. Then the new dataset was divided into training and test parts (70% for training sets and 30% for the testing set). Afterwards, the *10*-fold cross validation method [103] was employed to evaluate each particle using the k-NN classifier; thus, for each dataset the training set was split into 10 parts, where each part shared the same proportion for each class of the data. Therefore, for each dataset, the first nine data parts were applied in the training process to build the learning model, while the last one was utilized in the validation process to evaluate the particle's fitness value. If the accuracies of two solutions were the same, then the solution using the smaller number of features was selected.

The pseudo code of the HPSO-LS algorithm is shown in Fig. 3.

4. Experiments

4.1. Datasets

In this paper in order to evaluate the performance of the proposed method, several experiments were conducted. The experiments were performed on 12 datasets taken from the UCI machine learning repository [104], including *Glass*, *Vowel*, *Wisconsin Breast Cancer* (WBC), *Wine*, *Heart*, *Segment*, *Twonorm*, *Sonar, Arrhythmia, LSVT Voice Rehabilitation, Colon Cancer, Lymphoma* and *Leukemia*. These datasets cover examples of small, medium and large-dimensional datasets and they have been used in many studies of the machine learning research area. Table 1 shows the additional details of the mentioned datasets such as dimensionality, the number of the classes and the number of samples.

Algorithm 1. HPSO-LS- Hybrid particle swarm optimization with local search

Input $Feature = \{f_1, f_2, \dots, f_D\}$
NC_{max}: the maximum cycles that algorithm repeated
K<<D
NP: number of particles
Output $Feature' = \{f_1', \dots, f_k'\}$
Begin algorithm
Begin initialize
Run Determining number of features to determine k
for i=1 to NP **do**
X_{ij}=create random particle ∈ {0,1}
V_{ij}=create random velocity ~ $\mathbb{U}[0,1]$
end for
for all feature I **do**

$$c_{ij} = \sum (x_i - \bar{x}_i)(x_j - \bar{x}_j) \Big/ \sqrt{(x_i - \bar{x}_i)^2} \sqrt{(x_j - \bar{x}_j)^2}$$

$$cor_i = \sum_{i=1}^{f} |c_{ij}|/f - 1 \quad i \neq j$$

end for
$Similar\ feature \leftarrow cor_i \geq cor_{mid}$
$Dissimilar\ feature \leftarrow cor_i < cor_{mid}$
end initialize
for i =1 to NC_{max} **do**

$$\hat{v}_i(t+1) = v_i(t) + c_1.r_1.rand\left(x_i^{best}(t) - x_i(t)\right) v_i(t) + c_2.r_2.rand\left(x_g^{best}(t) - x_i(t)\right)$$

$$v_i(t+1) = sign\left((\hat{v}_i(t+1))min\{|\hat{v}_i(t+1)|, v_{max}\}\right) \ j = 1, \dots, v$$

$$x_i(t+1) = \mathbb{I}_{\{rand<s(v_i(t+1))\}}$$

$X_s = Similar\ feature$
$X_d = Dissimilar\ feature$
$x' = x$
Remove all feature in X_d that is 0 in particle x
Remove all feature in X_s that is 0 in particle x
Calculate the value of n_s and n_d
Perform "particle movement" on each position of particle and replace it
$f_i = fitness(x_i')$
If $f_i > fitness\left(x_i^{best}\right)$
$x_i^{best} = x_i'$
if$f_i > fitness\left(x_g^{best}\right)$
$x_g^{best} = x_i'$
end for
Return x_g^{best}
end algorithm

Fig. 3. Pseudo code of the proposed feature selection method.

Table 1
List of real world datasets from UCI used for comparison between algorithms.

Name	#features	#patterns	#classes
Glass	9	214	6
Vowel	10	528	11
Wisconsin Breast Cancer (WBC)	10	699	2
Wine	13	175	3
Heart	13	270	2
Segment	19	2310	7
Two norm	20	7400	2
Sonar	60	208	2
Arrhythmia	279	452	16
LSVT Voice Rehabilitation	309	126	2
Colon cancer	2000	62	2
Lymphoma	4026	47	2
Leukemia	7129	72	2

4.2. Parameters setting

In all the experiments in this section, the values of the common parameters such as the maximum cycle number and the population size were set as follows. The maximum number of cycles was set to 50 (i.e. NC_{max} = 50) and the swarm size was set to 20 (i.e. NP = 20). Furthermore, for each function all the methods were run 20 times with random seed on the PC with Intel Core 2 Due, 3 GHz CPU and 4 GHz of RAM. The other specific parameters of the algorithms were set as follows:

For PSO, CBPSO [105], HPSO-STS [106], SPSO-QR [94], PSO(4-2) [93] and HPSO-LS the cognitive and social components, c_1 and c_2, were set to be 2. Moreover, the upper and lower bounds for v were set to 4 and -4, respectively (i.e. v_{max} = 4 and v_{min} = -4). If the sum of accelerations caused the velocity on that dimension $v(t+1)$ to exceed v_{mix} or v_{max}, then the velocity on that dimension was limited to v_{mix} or v_{max}, respectively. Furthermore, for HPSO-LS, the value of ε which is used to determine the number of the initial features (i.e. Step 1) was set randomly in the range of [0.15,0.7] and the similar/dissimilar control parameter α was set as 0.65. This parameter is used by the local search operation of the proposed method (i.e. Step 5). For GA and HGAFS [101], the probability of crossover operation and probability of mutation operation were set to 0.6 and 0.02, respectively and the number of ants for ACOFS [64] was set to 20.

4.3. Results

The proposed method (HPSO-LS) is compared with basic wrapper methods as well as well-known metaheuristic algorithms such as simulated annealing (SA), genetic algorithm (GA), particle swarm optimization (PSO) and also ant colony optimization (ACO) which are applied for the feature selection problem.

Table 2
Average classification accuracy (in %) and standard deviation of HPSO-LS, PSO, ACO, SA and GA feature selection methods over 20 independent runs using 1-NN classifier. Moreover, the results of 1-NN classifier on the datasets with original features are reported in '*Without FS*' column. In each dataset, the best method is marked by the boldface.

Dataset	Without FS	GA	SA	ACO	PSO	HPSO-LS
Glass	71.875	64.51 ± 8.86	62.64 ± 4.7	70.77 ± 6.54	70.31 ± 3.68	**74.91** ±3.63
Vowel	97.60	84.109 ± 12.47	84.03 ± 12.53	74.857 ± 14.06	95.872 ± 0.26	**99.87** ± 0.20
WBC	97.12	96.361 ± 2.42	96.88 ± 1.03	95.833 ± 1.31	97.69 ± 0.58	**98.27** ± 0.4
Wine	96.55	92 ± 8.02	89.18 ± 3.86	92.236 + 2.724	94.24 ± 4.18	**97.17** ± 1.39
Heart	76.54	72.464 ± 6.73	75.18 ± 5.47	73.69 ± 5.0	74.81 ± 5.04	**78.84** ± 2.07
Segment	89.03	83.11 ± 5.08	81.181 ± 9.74	92.17 ± 0.74	90.775 ± 0.93	**92.348** ± 0.669
Two norm	93.24	86.35 ± 3.9	85.13 ± 3.09	74.9 ± 6.88	89.92 ± 2.82	**94.25** ± 0.59
Sonar	79.03	83.28 ± 5.36	76.83 ± 3.48	66.9 ± 8.16	82.233 ± 3.71	**87.23** ± 3.40
Arrhythmia	49.62	49.69 ± 5.33	48.99 ± 3.61	47.99 ± 5.90	51.65 ± 4.51	**53** ± 2.28
LSVT	60.52	71.71 ± 4.93	71.45 ± 3.01	68.59 ± 6.26	73.21 ± 4.89	**77.06** ± 3.62
Cancer	72.22	79.56 ± 5.57	78.68 ± 4.98	75.23 ± 5.43	80.33 ± 5.98	**83.88** ± 4.09
Lymphoma	61.53	79.01 ± 6.34	77.81 ± 5.16	70.63 ± 4.37	80.65 ± 6.89	**82.85** ± 5.48
Leukemia	85.71	85.28 ± 4.74	82.21 ± 5.92	79.54 ± 5.44	87.39 ± 4.67	**89.28** ± 3.36
average	79.27	79.03	77.707	75.64	82.23	**85.30**
std						

Table 3
Compare subset feature selected by algorithm.

Dataset	GA	SA	PSO	ACO	HPSO-LS
Glass	2,3,4,5,6	3,4,5,6	4,5,6,7,9	3,4,5,6	3,4,5,6
Vowel	4,5,6,7	4,5,6,7,9	3,4,5,6,8	3,4,5	3,4
WBC	2,4,5,6,8	3,4,5,6,7	4,5,6,7,8	4,5,6,8	4,5,6,7,8
Wine	4,6,7,8,10	5,6,7,9	3,4,5,6,7,8,9	3,5,8,9	4,5,6,7,8
Heart	4,5,6,7,9	3,5,6,7,9	5,6,7,8,9,10	4,6,8,9,10	3,4,6,7,8
Segment	6,7,9,10,11,12	3,8,9,11,12	7,9,10,11,12	7,9,11,12,14,15	9,10,11,12,13,14,15
Two norm	9,10,11,12,14,16	7,9,10,11,13,14	11,12,13,14,15,17	7,8,11,13,14,15,16,18	16,17,18
Sonar	26,28,32,34,37	27,30,32,35,36	20,26,28,32,34,36	20,25,26,29,32,37,39	23,28,33,34, 36

Moreover, the proposed method is compared with filter-based feature selection methods including Information gain (IG) [73], term variance (TV) [75], Fisher score (F-Score) [78] and minimal-redundancy-maximal-relevance (mRMR) [79]. Furthermore, the proposed method is also compared with state-of-the-art feature selection methods including HGAFS [101], HPSO-STS [106], SPSO-QR [94], CBPSO [105] and PSO(4-2) [93].

In the experiments the k-nearest neighbor (k-NN) [107] was applied to evaluate the fitness of each particle in the wrapper-based methods. The k-NN method is a simple and commonly used learning algorithm and has been successfully applied in many research fields such as pattern recognition, statistical estimation, image processing and text mining. To estimate the performance of the proposed feature selection method, the one nearest neighbor (1-NN) is used as the evaluator.

4.3.1. Comparison of HPSO-LS and GA, SA, ACO and PSO

Several experiments were performed in order to compare the performance of the HPSO-LS method with those of GA, SA, PSO and ACO-based feature selection methods and the results have been reported in terms of means and standard deviations in Table 2. The 1-NN classifier accuracy which is obtained in the 20 independent runs is considered as the performance of the methods. Moreover, the accuracy of the 1-NN classifier on the original features (i.e. without FS) is also reported. According to the Table 2 results it can be seen that the proposed method produced the highest classification accuracy in comparison with the other feature selection methods on all of the datasets. This is due to the fact that the global search property of HPSO-LS originates from PSO and the local search operations of HPSO-LS result in selecting more discriminative features and removing irrelevant and redundant features. For example, the mean accuracy (standard deviation) of 1-NN classifier over the *Sonar* dataset was reported 87.23 (3.40) while the values for PSO, ACO, SA and GA were, respectively, reported 82.233 (3.71), 66.9 (8.16), 76.83 (3.48) and 83.28 (5.36). Moreover, in this case, the accuracy value for the case of using all of the original features without applying any feature selection methods was reported 79.03, that no feature selection methods had reported. Furthermore, the results show that in some cases the methods didn't select the appropriate features and thus their performance was reduced compared with the situation where all of the features were considered by the classifier. For example for the *Arrhythmia* dataset, the mean accuracy of ACO and SA were reported 47.99 and 48.99, respectively, while this value was reported 49.62 for the case of not using any feature selection methods.

In addition, the table results demonstrate that HPSO-LS produced robust results compared with those of PSO, ACO, SA and GA feature selection algorithms. For example, the standard deviation of the proposed method for the *Segment* (*Twonorm*) dataset was reported 0.669 (0.59) respectively, while in this case, the standard deviation of PSO, ACO, SA and GA were, respectively, reported 0.93(2.82), 0.74(6.88), 9.74(3.09) and 5.08(3.9). Consequently, from the results, it can also be concluded that the proposed method obtained the best results for the datasets with large numbers of features. For example, the differences between the obtained mean classification accuracy of the proposed method and that of the second best ones were reported 3.55 (i.e. 83.88–80.33), 2.2 and 1.89 for the *Cancer* (2000), *Lymphoma* (4026) and *Leukemia* (7129) datasets (number of features), respectively.

Moreover, Table 3 shows the features which are selected by the SA,GA, ACO, PSO and HPSO-LS methods. The numbers in this table show the indexes of the features in the dataset. The results show that each of the mentioned methods selected different subsets of features. It can also be seen from the Tables 2 and 3 results that the proposed method achieved the minimum number of features as well as obtained the highest classification accuracy compared with the other methods.

As another comparison, Tables 4 and 5 compare the classification error rates of the proposed method with those of the SA, GA, ACO and PSO feature selection methods on the *Arrhythmia* and *LSVT*

Table 4
Average classification accuracy (average over 20 runs, in %) of HPSO-LS, PSO, ACO, SA and GA feature selection methods considered over *Arrhythmia* dataset using 1-NN classifier. Moreover, mean (Average) and standard deviation(std) of the methods are also reported. The best result for each number of features is shown in bold face.

# feature	GA	SA	ACO	PSO	HPSO-LS
10	45.18	48.33	46.29	50.21	**54.07**
20	47.4	47.28	48.09	51.11	**53.33**
40	51.59	50.75	49.41	51.59	**53.14**
70	50.73	49.84	50.25	52.25	**55.18**
100	49.62	48.56	45.73	52.05	**54.31**
150	48.88	49.07	45.45	50.19	**52.59**
Average	48.9	48.97	47.53	51.23	**53.77**
Std	2.33	1.21	2.01	0.89	0.93

Table 5
Average classification accuracy (average over 20 runs, in %) of HPSO-LS, PSO, ACO, SA and GA feature selection methods considered over *LSVT* dataset using 1-NN classifier. Moreover, mean (Average) and standard deviation(std) of the methods are also reported. The best result for each number of features is shown in bold face.

# feature	GA	SA	ACO	PSO	HPSO-LS
10	60.03	58.49	57.01	60.52	**71.18**
20	69.89	65.93	60.10	71.053	**81.29**
40	63.76	68.29	62.23	65.789	**80.25**
70	63.09	63.98	63.81	63.158	**77.77**
100	62.21	60.37	61.29	63.15	**75.21**
150	58.56	59.90	55.18	60.526	**74.22**
Average	62.92	62.82	59.93	64.03	**76.7**
Std	3.92	3.85	3.26	3.96	3.9

datasets, respectively, when the final feature subsets with different sizes are selected. It is clear from the results that the performance of the proposed method is much better than the performance of the other feature selection algorithms. For example, when the number of selected features is set to 70 ($sf = 70$) for the *Arrhythmia* dataset, the accuracy of HPSO-LS was reported 55.18 while the accuracy values for PSO, ACO,SA and GA were reported 52.25, 50.25, 49.84 and 50.73, respectively. Furthermore, Table 5 reported the same results for the LSVT dataset. For example, the obtained average values of the proposed method over different numbers of selected features were accordingly reported 53.77 and 76.7 for *Arrhythmia* and *LSVT*. Consequently, it can be concluded from the Tables 2–5 results that the performance of the proposed method is much better than the performance of the other feature selection algorithms.

4.3.2. Comparison of HPSO-LS and state-of-the-art wrapper methods

The performance of the solution accuracy of HPSO-LS is compared with those of state-of-the-art wrapper-based methods including HGAFS [101], HPSO-STS [106], SPSO-QR [94], CBPSO [105] and PSO(4-2) [93]. The comparison results are shown in Table 6 in terms of means and standard deviations of classification accuracy of the solutions in the 20 independent runs. In the table, the best results are bolded and underlined and the second best results are only bolded. The results show that HPSO-LS offered the highest accuracy for most of the datasets and achieved better performance. For example, the proposed method achieved the mean value (standard deviation) 92.348 (0.669) for the *Segment* dataset while in this case PSO(4-2), CBPSO, SPSO-QR, HPSO-STS and HGAFS, respectively, obtained 85.42 (4.81), 90.03 (1.36), 81.33 (9.08), 88.19 (7.02) and 88.49 (1.11). Moreover, the results show that the HPSO-LS method obtained the best average results compared with the other methods, while the difference between the average mean accuracy value of the best (i.e. HPSO-LS) and the second best (i.e. CBPSO) is 5.7 (i.e. 85.30–79.53). Furthermore, the table results show that the proposed method produced robust results compared with those of the state-of-the-art feature selection algorithms. For example, the mean standard deviation of the proposed method over all of the datasets was reported 2.39, while in this case the mean standard deviations of PSO(4-2), CBPSO, SPSO-QR, HPSO-STS and HGAFS were, respectively, reported 4.70, 3.09, 5.42, 4.91 and 3.58. Consequently, it can be concluded from the results that the HPSO-LS is superior to the state-of-the-art methods.

Furthermore, several experiments were performed to compare the proposed method with the other feature selection methods based on the different numbers of selected features. Fig. 4a–d plots the classification accuracy (average over 20 independent runs) curves of the 1-NN classifier on the *Arrhythmia*, *LSVT*, *Leukemia* and *Lymphoma* datasets, respectively. In all the plots, the x-axis denotes the subset of selected features, while the y-axis is the average classification accuracy. Fig. 3a shows that the HPSO-LS is superior to the other methods for almost all numbers of selected features. For example, when the number of selected features was set to 100 (i.e. $sf = 100$) for the *Arrhythmia* dataset, the accuracy of HPSO-LS was reported 54.31 while in this case the accuracy values for HPSO-STS, HPSO-QR, PSO(4-2), CBPSO and HGAFS were reported 50.37, 49.88, 51.01, 51.33 and 46.66 respectively. In addition, Fig. 4b represents similar results when the HPSO-LS was applied on the LSVT dataset. For example, when *150* features were selected the difference between the accuracy values of the proposed method and those of HPSO-STS, HPSO-QR, PSO(4-2), CBPSO and HGAFS were respectively reported 13.694 (i.e. 74.22−60.525), 9.75, 4.97,7 and 3.02. Fig. 4c and 4d also report similar results for the *Leukemia* and *Lymphoma* datasets, respectively. Consequently, it can be concluded from the Table 6 and Fig. 4 results that the proposed method is superior to the state-of-the-art wrapper-based feature selection methods.

4.3.3. Comparison of HPSO-LS and filter-based methods

The performance of the proposed method was also compared with those of filter-based methods including Information gain (IG) [73], term variance (TV) [75], Fisher score (F-Score) [78] and minimal-redundancy-maximal-relevance (mRMR) [79]. The IG, TV and F-Score are univariate filter-based methods and in these methods, the relevance of a feature is measured individually using an evaluation criterion, while mRMR is a multivariate filter method and the dependencies between features is considered in the evaluation of the relevance of features. On the other hand IG, F-Score and mRMR are supervised methods, so they use class labels of the training patterns to compute the relevance of each feature while TV is an unsupervised method and it does not employ any class labels to evaluate the feature relevancy. Table 7 compares the performance of the proposed method with those of filter-based methods for different numbers of selected features in terms of the means of the rates of accuracy of the solutions in the 20 independent runs. The results show that the proposed method is superior to the mentioned supervised univariate feature selection methods (i.e. IG and F-Score). Furthermore, the results demonstrate that the performance of the proposed method is higher than that of a supervised multivariate feature selection method (i.e. mRMR). From the results, it can be concluded that the overall performance of the proposed method is much better than those of the mentioned unsupervised method (i.e. TV) over different datasets. Moreover, the HPSO-LS method outperformed unsupervised methods for different numbers of selected features. This is due to the fact that in these methods, the possible dependency between features as well as the class label information is ignored in the feature selection process. Moreover, the proposed method is a wrapper-based method and a learning model is used to evaluate the feature subsets, so both class labels and feature dependencies are considered in the selection of relevant feature subsets. Therefore, the proposed method was able to outperform the mentioned filter-based methods.

Table 6
Average classification accuracy (in %) and standard deviation of HPSO-LS, PSO(4-2), CBPSO, SPSO-QR, HPSO-STS and HGAFS feature selection methods over 20 independent runs using 1-NN classifier. In each dataset, the best method is marked by the boldface and underlined and the second best result is also boldfaced.

Dataset	HGAFS	HPSO-STS	SPSO-QR	CBPSO	PSO(4-2)	HPSO-LS
Glass	65.16 ± 2.30	67.57 ± 6.06	70.75 ± 7.25	68.74 ± 6.54	**73.21** ± 6	**74.91 ± 3.63**
Vowel	87.34 ± 2.43	92.96 ± 13.09	89.68 ± 11.90	**98.47 ± 1.61**	96.01 ± 5.37	**99.87 ± 0.20**
WBC	96.78 ± 0.74	95.73 ± 3.70	96.38 ± 1.43	**97.93 ± 0.30**	97.04 ± 1.33	**98.27 ± 0.4**
Wine	90.68 ± 7.80	**96.72 ± 0.97**	91.59 ± 7.28	95.17 ± 4.04	93.27 ± 3.27	**97.17 ± 1.39**
Heart	68.11 ± 7.72	73.65 ± 7.05	74.22 ± 7.12	**76 ± 5.60**	73.78 ± 4.13	**78.84 ± 2.07**
Segment	88.49 ± 1.11	88.19 ± 7.02	81.33 ± 9.08	**90.03 ± 1.36**	85.42 ± 4.81	**92.348 ± 0.669**
Two norm	87.14 ± 4.48	**91.43 ± 1.97**	85.92 ± 3.23	89.59 ± 2.53	89.29 ± 2.67	**94.25 ± 0.59**
Sonar	79.20 ± 4.17	82.65 ± 4.29	80.28 ± 5.02	**84.30 ± 3.38**	83.24 ± 3.75	**87.23 ± 3.40**
Arrhythmia	47.93 ± 2.83	**50.37 ± 1.04**	50.05 ± 0.722	48.96 ± 2.99	48.14 ± 3.87	**53.00 ± 2.28**
LSVT	68.05 ± 7.16	63.15 ± 3.72	65.20 ± 2.87	65.78 ± 1.99	**67.36 ± 7.11**	**77.06 ± 3.62**
Cancer	**74.74 ± 1.31**	69.436 ± 5.37	66.66 ± 5.861	70.36 ± 2.87	67.59 ± 5.46	**83.88 ± 4.09**
Lymphoma	66.66 ± 2.82	**70.32 ± 6.05**	66.86 ± 5.71	58.11 ± 4.05	66.41 ± 7.57	**82.85 ± 5.48**
Leukemia	82.24 ± 1.76	86.49 ± 3.54	84.91 ± 3.08	90.47 ± 3.01	**85.03 ± 5.78**	**89.28 ± 3.36**
Average	77.11 ± 3.58	79.12 ± 4.91	77.21 ± 5.42	**79.53 ± 3.09**	78.90 ± 4.70	**85.30 ± 2.39**

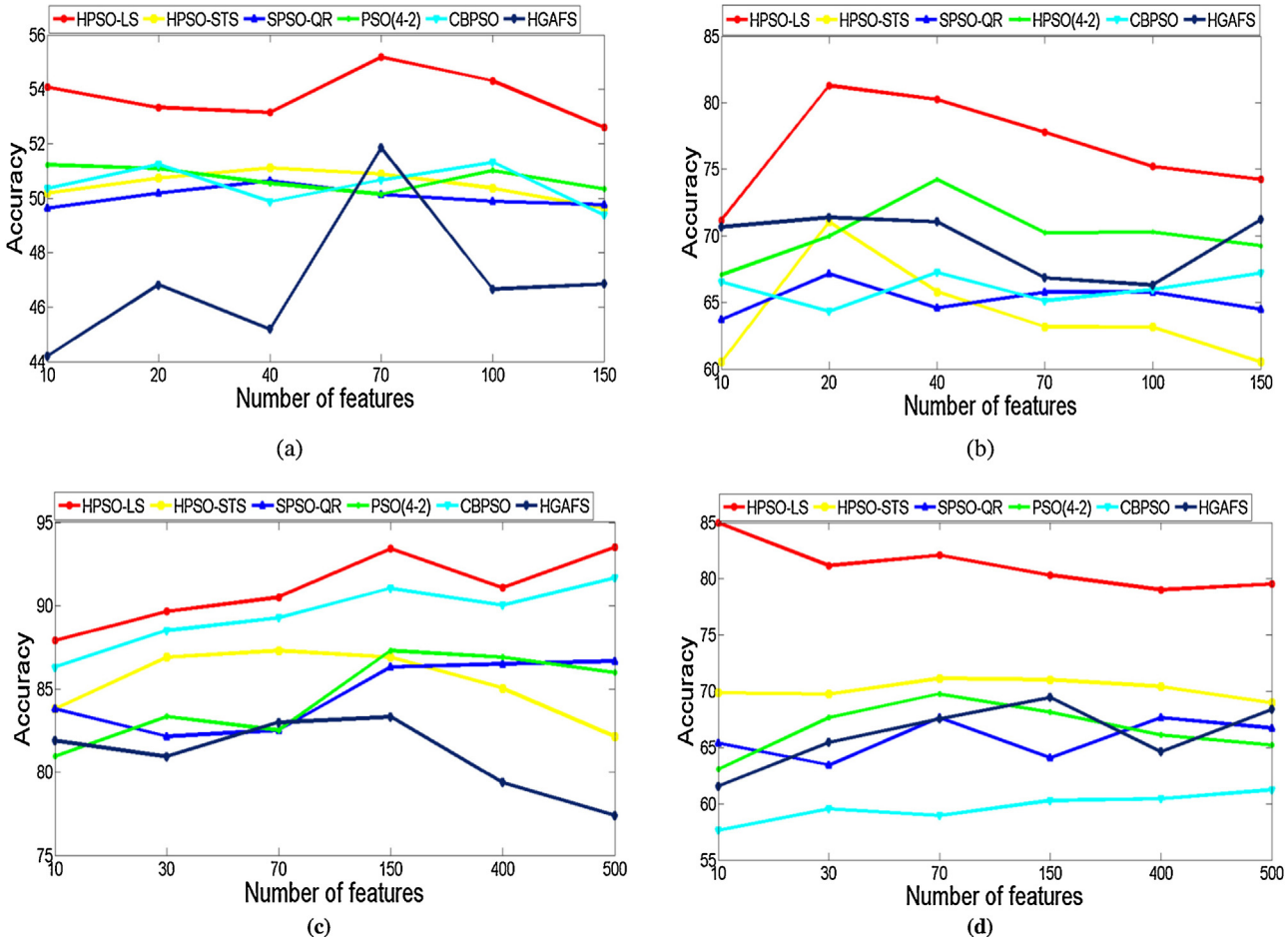

Fig. 4. Average classification accuracies for different features subset sizes of HPSO-LS, HPSO-STS, SPSO-QR, PSO(4-2), CBPSO and HGAFS feature selection methods over the (a) *Arrhythmia*, (b) *LSVT*, (c) *Leukemia* and (d) *Lymphoma* datasets.

4.3.4. Statistical analysis of HPSO-LS

In order to perform a comprehensive comparison of the proposed method and the other algorithms, the independent *t*-test and Friedman test methods were used to measure the statistical significance of the results from all methods. Statistics offers more powerful procedures to test the significance of differences between multiple methods. The independent *t*-test, also called the two sample *t*-test or student's *t*-test (in short T-test) is an inferential statistical test that determines whether there is a statistically significant difference between the means in two unrelated groups. In the experiments, the significance level in the T-tests is selected as 0.05 (or the confidence interval is 95%). Table 8 shows the *p*-value results of T-test between the classification performance achieved by the HPSO-LS over 20 independent runs and those of PSO(4-2), CBPSO, SPSO-QR, HPSO-STS and HGAFS feature selection methods. In this table, a *p*-value which is less than 0.05 indicates that the classification performance of the HPSO-LS significantly better than its corresponding feature selection algorithm. The reported results show that in all cases, the proposed method obtained the best results. For example, the *p*-value of statistical T-test between the proposed method (i.e. HPSO-LS) and the other algorithms including HGAFS, HPSO-STS, HPSO-QR, CBPSO and

Table 7
Average classification accuracy (in %) of HPSO-LS and those of filter based methods including information gain (IG), fisher score (F-Score), term variance (TV) and mRMR over 20 independent runs using 1-NN classifier. In each dataset, the best method is marked by the boldface and underlined and the second best results are also boldfaced.

Dataset	#features	IG	F-Score	TV	mRMR	HPSO-LS
Glass	4	63.66	59.37	58.74	**64.37**	**72.39**
	6	**70.31**	65.93	69.37	67.80	**76.73**
Vowel	3	99.15	57.56	**99.57**	86.39	**100**
	6	**98.73**	80.50	98.06	97.97	**99.87**
WBC	4	96.64	74.15	**97.32**	96.40	**98.08**
	6	**98.56**	96.48	97.22	96.01	**98.80**
Wine	4	**96.26**	70.69	84.47	89.65	**97.18**
	8	**97.41**	87.93	97.12	97.12	**98.21**
Heart	4	**80.06**	65.68	53.08	75.72	**79.98**
	8	76.53	67.48	73.24	**77.74**	**78.38**
Segment	8	**89.24**	79.65	81.81	85.27	**91.73**
	12	**90.86**	88.30	88.98	89.89	**91.97**
	15	**90.76**	89.75	89.75	90.37	**92.46**
Two norm	8	84.78	**85.31**	82.50	83.66	**93.12**
	12	88.44	86.98	89.20	**89.39**	**94.97**
	15	91.34	91.62	**92.78**	92.2	**95.34**
Sonar	10	**79.83**	70.15	71.60	56.04	**83.24**
	25	**83.06**	73.38	72.57	73.86	**89.12**
	35	**83.87**	74.72	78.70	77.81	**88.36**
Arrhythmia	10	47.4	38.2	45.92	**47.95**	**54.07**
	30	**51.17**	41.72	40.59	47.98	**53.24**
	50	49.39	42.58	**52.22**	48.28	**55.31**
LSVT	10	**71.05**	47.36	62.27	65.26	**71.18**
	30	**70.66**	57.89	61.04	54.38	**80.86**
	50	64.52	52.63	**65.78**	60.52	**78.32**
Cancer	30	70.22	72.21	**79.18**	78.69	**80.17**
	60	74.44	52.77	**80.55**	73.60	**84.38**
	100	80.33	63.88	72.21	**80.55**	**82.74**
Lymphoma	10	**76.46**	42.12	64.60	57.62	**82.33**
	50	77.28	38.46	75.38	**78.74**	**87.71**
	100	79.01	46.15	**84.61**	82.33	**83.21**
Leukemia	10	81.16	60.31	**83.66**	63.48	**84.32**
	60	**83.45**	68.83	83.33	64. 28	**87.76**
	100	**87.96**	71.42	74.6	71.42	**89.86**

Table 8
The *p*-value results of T-test between the classification performance achieved by the HPSO-LS over 20 independent runs and those of PSO(4-2), CBPSO, SPSO-QR, HPSO-STS and HGAFS feature selection methods. The results lower that 0.005 are bolded.

Dataset	HGAFS	HPSO-STS	HPSO-QR	CBPSO	PSO(4-2)
Glass	0.006	**0.009**	0.018	0.021	0.034
Vowel	0.010	0.042	0.027	0.023	0.032
WBC	**0.005**	0.023	0.011	0.037	0.031
Wine	0.022	0.020	0.034	0.032	0.017
Heart	0.020	0.024	0.021	0.032	0.010
Segment	0.013	0.044	0.030	0.029	0.025
Two norm	0.030	0.032	0.032	0.031	0.011
Sonar	**0.003**	0.012	0.018	**0.000**	**0.003**
Arrhythmia	**0.004**	0.047	0.023	0.010	0.024
LSVT	0.028	**0.000**	**0.001**	0.007	0.021
Cancer	0.029	**0.003**	**0.004**	**0.003**	**0.002**
Lymphoma	0.006	**0.001**	**0.000**	**0.001**	**0.005**
Leukemia	0.017	**0.001**	**0.001**	0.023	0.033
Average	0.014	0.019	0.016	0.012	0.018

PSO(4-2) over Wine dataset were, respectively, reported 0.0022, 0.020,0.034, 0.032 and 0.017. While all of these values are less than 0.05 then with confidence level of 95% we can claim that the proposed method statistically performed better than the other feature selection methods.

The Friedman test [108] is a non-parametric method which is used to compare different methods over multiple datasets by ranking each algorithm on each dataset. The Friedman test has been used in several pieces of research to statistically analyze feature selection methods [25,26]. For each subset of features, the different accuracies are ranked from one to the number of methods. The method with the highest classification accuracy will have rank 1, and the second best gets ranks 2, and so on. The Friedman estimator or F_F follows a Fisher distribution that allows analyzing the statistical significance of the results. This estimator is as follows (i.e. Eq. (11)):

$$F_F = \frac{(N_B - 1)X_f^2}{N_B(N_M - 1) - X_f^2} \tag{11}$$

where N_B and N_M are the number of datasets and the number of methods, respectively, and X_f^2 is Fisher distribution which is defined in Eq. (12) as follows:

$$X_f^2 = \frac{12N_B}{N_M(N_M+1)}\left(\sum_{j=1}^{N_M} R_j - \frac{N_M(N_M+1)^2}{4}\right) \tag{12}$$

where Rj is the average rank for each method.

F_F follows a Fisher distribution with N_M-1 and $(N_M\text{-}1)(N_B\text{-}1)$ degrees of freedom. In the experiments, the critical value of the Fisher distribution is set to $\alpha = 0.05$, 95% confidence. If the value of F_F is less than the corresponding critical value, the null hypothesis is retained; otherwise the null hypothesis is rejected. If the null hypothesis is rejected, the Holm test [16] is applied to compare all classifiers with a control classifier. In the Holm test, the value of z is used to compute the statistical difference between two methods as follows (i.e. Eq. (13)):

$$z = \frac{R_i - R_j}{\sqrt{\frac{N_M(N_M-1)}{4}}} \tag{13}$$

where R_i and R_j denote the average ranks of the i-th and j-th methods. For each value of z, a corresponding p-value can be obtained according to the table of standard normal distribution. In the Holm test, the value of p is compared to $\alpha/(N_B - i)$ where i is the order of the hypothesis. If the p_i is less than $\alpha/(N_B - 1)$, the corresponding hypothesis is rejected, and the difference between the two classifiers is significant.

Table 9 shows the results of the Friedman statistical test of the proposed method compared with those of filter-based and wrapper-based methods. The table shows the statistical test results in three parts. In the first part HPSO-LS is compared with PSO(4-2), CBPSO, STPSO-QR, HPSO-STS and HGAFS. In the second part, the proposed method is compared with filter-based methods including IG, F-Score, TV and mRMR, and in the last part the method is compared with the PSO, ACO, SA and GA methods.

Fisher distribution for the first part of the table for five methods, $N_M = 6$ that appear in the comparison and 13 databases, $N_B = 13$, with degrees of freedom 5 and 60 (i.e. $N_M - 1 = 5$, $(N_M - 1)(N_B - 1) = 60$), obtains a critical value of the Fisher distribution $F_{(5,60)} = 2.254$. On the other hand, Fisher distribution for the second part and the third part with five methods is $F_{(4,48)} = 2.565$. Because the critical value of $F_{(5,60)}$ is less than F_F=14.8264 for the first part of the results, the significant differences between HPSO-LS and PSO(4-2), CBPSO, SPSO-QR, HPSO-STS and HGAFS are detected. As another comparison, the z-value is obtained for the mentioned methods. The results show that the corresponding p-values for PSO(4-2), CBPSO, SPSO-QR, HPSO-STS and HGAFS are less than $\alpha/(N_M - 1)$, so the significant differences between the proposed method and the state-of-the-art methods is detected. Furthermore, from the second part and the third part results of the table, it can be seen that there is a significant difference between the proposed method and the filter-based methods as well as classical wrapper-based methods.

Table 9
The results of Friedman-test between the obtained rank of the HPSO-LS and rank of the wrapper and the filter-based feature selection methods. The hypotheses ordered by p-value and adjusting of α by Holm procedure, considering an initial $\alpha = 0.05$.

i	Algorithms	mean rank	$z = \frac{R_0 - R_i}{SE}$	p	$\alpha = \frac{0.05}{NB-i}$	F_F	Significant	Significant
1	HGAFS	4.81	5.1922	0.00001	0.0100	14.8264	$>F_{(5,60)} = 2.254$	Positive
2	SPSO-QR	4.65	4.9741	0.00001	0.0125			
3	HPSO-STS	3.85	3.8839	0.000103	0.0167			
4	PSO(4-2)	3.62	3.5705	0.000356	0.0250			
5	CBPSO	3.08	2.8346	0.004588	0.0500			
1	F-Score	4.46	5.4501	0.00001	0.0125	22.9864	$>F_{(4,48)} = 2.565$	Positive
2	TV	3.62	4.0956	0.000042	0.0167			
3	mRMR	3.31	3.5958	0.000323	0.0250			
4	IG	2.54	2.3542	0.018563	0.0500			
1	ACO	4.15	5.0792	0.00001	0.0125	9.8894	$>F_{(4,48)} = 2.565$	Positive
2	SA	3.92	4.7084	0.00001	0.0167			
3	GA	3.31	3.7248	0.000195	0.0250			
4	PSO	2.23	1.9833	0.047334	0.0500			

Table 10
Average of running time (in seconds). In each dataset, the best method is marked by the boldface.

Dataset	HGAFS	HPSO-STS	HPSO-QR	CBPSO	PSO(4_2)	IG	F-Score	TV	mRMR	HPSO-LS
Glass	0.33	5.10	15.45	0.38	0.22	0.02	**0.01**	0.04	0.03	0.39
Vowel	1.06	8.04	2:02.66	2.09	0.65	0.07	**0.06**	0.09	0.10	1.82
WBC	2.31	9.43	0.19	3.90	1.07	0.10	**0.08**	0.14	0.15	3.24
Wine	0.34	0.80	22.30	0.34	0.27	0.3	0.1	**0.03**	0.06	0.34
Heart	0.68	7.06	0.11	0.76	0.30	0.3	**0.03**	0.06	0.06	0.54
Segment	1:03.23	55:43.17	4:12:33	1:06.66	17.51	**1.34**	1.84	1.35	2.57	58.91
Twonorm	9:38.55	22:14.91	19.87	12:41.02	2:30.5	**13.8**	18.50	14.4	16.91	14:02.23
Sonar	1.69	4.60	**0.04**	1.57	0.47	**0.04**	0.07	0.07	2.03	1.41
Arrhythmia	46.42	4:55.37	5:56:06	34.46	7.72	1.12	0.96	**0.45**	9.32	24.94
LSVT	8.39	7.48	2.32	5.33	3.33	2.63	3.30	**2.12**	16.97	6.08
Cancer	1:54.30	19.54	2.43	5.82	2.81	0.34	0.28	**0.24**	6:59.51	1:14.28
Lymphoma	4:22.34	15.80	7.48	7.34	4.37	0.46	1.04	**0.26**	9:52.95	3:44.15
Leukemia	20:07.5	54.42	42.46	28.88	14.52	1.40	0.82	**0.68**	22:17.4	18:04.09

4.3.5. Experiments to analyze the execution time

The results of various performed experiments show that the HPSO-LS method is a well performed feature selection method. However, in this section, the computational time requirement of the proposed method is compared with those of wrapper-based methods including HGAFS, HPSO-STS, SPSO-QR, CBPSO and PSO(4_2). Moreover, the execution time of the HPSO-LS is also compared with those of filter-based methods including IG, F-Score, TV and mRMR. All of the algorithms were implemented using the C#.NET programming language and these methods were run under the same conditions on the PC with Intel Core 2 Due, 3 GHz CPU and 4 GHz of RAM. Table 10 shows the mean execution times (in seconds) obtained by the mentioned algorithms to select a feature subset from the benchmark datasets described in Table 1. As seen from the Table 10 results, univariate filter-based methods (e.g. IG, F-Score and TV) were able to obtain shorter computational times, while multivariate filter-based methods (e.g. mRMR) and wrapper-based methods required considerably long computational times. For example for the WBC dataset, the run time of HPSO-LS was 3.24 s while the run times were reported 7.19, 13.8, 90.80 (1:30.80) and 3.05 s for SA, GA, ACO and PSO. This is due to the fact that the wrapper-based methods employ a learning model in their search operations and thus in each iteration of these algorithms the learning model is needed to compute the fitness of each solution. On the other hand from the Tables 7–10 results, it is clear that univariate filter-based methods obtained worse quality results compared with those of the HPSO-LS method. Furthermore, the HPSO-LS, HGAFS and mRMR methods obtained the worst computational times compared with those of the other methods. This is because of the fact that the search operations of these methods need to compute the correlation values between each two features. Consequently, according to the results reported in Tables 2–10 and Fig. 4, the quality of the HPSO-LS is superior to those of the other methods. Therefore, there is a trade-off between the goodness of the results and the corresponding computational time.

5. 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 particle swarm optimization called HPSO-LS was proposed for feature selection. The proposed method integrated new local search operations with the global search process of PSO. The local search operations of HPSO-LS employ correlation information of the features to guide the search process in PSO in such a way that relatively less correlated features are selected with a high probability compared with more correlated ones. Moreover, HPSO-LS selects the reduced number of salient features by using a subset size determination scheme. The performances of the proposed methods were compared with those of the state-of-the-art wrapper-based feature selection methods including PSO(4-2), CBPSO, SPSO-QR, HPSO-STS and HGAFS and filter-based feature selection methods including IG, F-Score, TV and mRMR. Moreover, the HPSO-LS was compared with well-known wrapper-based feature selection methods such as GA, SA, ACO and PSO. The experimental results were reported from the three different aspects of classification accuracy, size of subset of selected features and execution time. The results of the experiments performed on the low- and high-dimensional datasets indicated that the proposed method effectively removed the irrelevant and redundant features. The joint use of the correlation-based local search operations and the global search strategy of PSO led to classification results superior to those of the wrapper and filter-based feature selection. Furthermore, the T-test and Friedman test were used to assess the statistical significance

of the differences between the proposed method and the other methods. The performed statistical tests revealed that the differences between the HPSO-LS and the other methods are statistically significant.

In future work, the local search strategy can be integrated with the multi-objective feature selection optimization framework to enhance the classification accuracy as well as reduce the number of selected features. Another perspective is to improve the local search operations by clustering the features into several groups.

References

[1] H. Liu, L. Yu, Toward integrating feature selection algorithms for classification and clustering, IEEE Trans. Know. Data Eng. 17 (2005) 491–502.
[2] A. Unler, A. Murat, R.B. Chinnam, mr2PSO: a maximum relevance minimum redundancy feature selection method based on swarm intelligence for support vector machine classification, Inform. Sci. 181 (2011) 4625–4641.
[3] I. Guyon, Andr, #233, Elisseeff, An introduction to variable and feature selection, J. Mach. Learn. Res. 3 (2003) 1157–1182.
[4] J. Yang, Y. Liu, Z. Liu, X. Zhu, X. Zhang, A new feature selection algorithm based on binomial hypothesis testing for spam filtering, Know. -Based Syst. 24 (2011) 904–914.
[5] H. Uğuz, A two-stage feature selection method for text categorization by using information gain, principal component analysis and genetic algorithm, Know. -Based Syst. 24 (2011) 1024–1032.
[6] H.R. Kanan, K. Faez, An improved feature selection method based on ant colony optimization (ACO) evaluated on face recognition system, Appl. Math. Comput. 205 (2008) 716–725.
[7] Z. Yan, C. Yuan, Ant colony optimization for feature selection in face recognition, in: In: Biometric Authentication, Springer Berlin Heidelberg, 2004, pp. 221–226.
[8] H. Yu, G. Gu, H. Liu, J. Shen, J. Zhao, A modified ant Colony optimization algorithm for tumor marker gene selection, genomics, Proteomics Bioinformatics 7 (2009) 200–208.
[9] A. Zibakhsh, M.S. Abadeh, Gene selection for cancer tumor detection using a novel memetic algorithm with a multi-view fitness function, Eng. Appl. Artif. Intell. 26 (2013) 1274–1281.
[10] S. Tabakhi, A. Najafi, R. Ranjbar, P. Moradi, Gene selection for microarray data classification using a novel ant colony optimization, Neurocomputing (2015).
[11] C.-L. Huang, C.-Y. Tsai, A hybrid SOFM-SVR with a filter-based feature selection for stock market forecasting, Expert Syst. Appl. 36 (2009) 1529–1539.
[12] Y. Marinakis, M. Marinaki, M. Doumpos, C. Zopounidis, Ant colony and particle swarm optimization for financial classification problems, Expert Syst. Appl. 36 (2009) 10604–10611.
[13] M. Ramezani, P. Moradi, F.A. Tab, Improve performance of collaborative filtering systems using backward feature selection, in: In: Information and Knowledge Technology (IKT), 2013 5th Conference on, IEEE, 2013, pp. 225–230.
[14] A. Kuri-Morales, F. Rodríguez-Erazo, A search space reduction methodology for data mining in large databases, Eng. Appl. Artif. Intell. 22 (2009) 57–65.
[15] I.A. Gheyas, L.S. Smith, Feature subset selection in large dimensionality domains, Pattern Recogn. 43 (2010) 5–13.
[16] H. Liu, H. Motoda, Computational Methods of Feature Selection, Chapman & Hall/CRC, 2007.
[17] M.A. Hall, Correlation-based feature selection for discrete and numeric class machine learning, in: In: Proceedings of the Seventeenth International Conference on Machine Learning, Morgan Kaufmann Publishers Inc, 2000, pp. 359–366.
[18] L. Ke, Z. Feng, Z. Ren, An efficient ant colony optimization approach to attribute reduction in rough set theory, Pattern Recogn. Lett. 29 (2008) 1351–1357.
[19] J. Yang, V. Honavar, Feature subset selection using a genetic algorithm, intelligent systems and their applications, IEEE 13 (1998) 44–49.
[20] Y. Sun, Iterative RELIEF for feature weighting: algorithms, theories, and applications, IEEE Trans. Pattern Anal. Mach. Intell. 29 (2007) 1035–1051.
[21] X. Wang, J. Yang, X. Teng, W. Xia, R. Jensen, Feature selection based on rough sets and particle swarm optimization, Pattern Recogn. Lett. 28 (2007) 459–471.
[22] A.M.P. Canuto, K.M.O. Vale, A. Feitos, A. Signoretti, ReinSel: a class-based mechanism for feature selection in ensemble of classifiers, Appl. Soft Comput. 12 (2012) 2517–2529.
[23] S. Tabakhi, P. Moradi, F. Akhlaghian, An unsupervised feature selection algorithm based on ant colony optimization, Eng. Appl. Artif. Intell. 32 (2014) 112–123.
[24] S. Tabakhi, P. Moradi, Relevance–redundancy feature selection based on ant colony optimization, Pattern Recogn. 48 (2015) 2798–2811.
[25] P. Moradi, M. Rostami, A graph theoretic approach for unsupervised feature selection, Eng. Appl. Artif. Intell. 44 (2015) 33–45.
[26] P. Moradi, M. Rostami, Integration of graph clustering with ant colony optimization for feature selection, Know. -Based Syst. 84 (2015) 144–161.
[27] S. Abe, Modified backward feature selection by cross validation, in: Proceedings of the European Symposium on Artificial Neural Networks, 2005, pp. 163–168.
[28] E. Gasca, J.S. Sánchez, R. Alonso, Eliminating redundancy and irrelevance using a new MLP-based feature selection method, Pattern Recogn. 39 (2006) 313–315.
[29] S. Guan, J. Liu, Y. Qi, An incremental approach to contribution-based feature selection, J. Intell. Syst. (2004) 15–44.
[30] H. Chun-Nan, H. Hung-Ju, S. Dietrich, The ANNIGMA-wrapper approach to fast feature selection for neural nets, systems, man, and cybernetics, part B: cybernetics, IEEE Trans. 32 (2002) 207–212.
[31] D.P. Muni, N.R. Pal, J. Das, Genetic programming for simultaneous feature selection and classifier design, systems, man, and cybernetics, part B: cybernetics, IEEE Trans. 36 (2006) 106–117.
[32] E. Romero, J.M. Sopena, Performing feature selection with multilayer perceptrons, neural networks, IEEE Trans. 19 (2008) 431–441.
[33] R.K. Sivagaminathan, S. Ramakrishnan, A hybrid approach for feature subset selection using neural networks and ant colony optimization, Expert Syst. Appl. 33 (2007) 49–60.
[34] A. Verikas, M. Bacauskiene, Feature selection with neural networks, Pattern Recogn. Lett. 23 (2002) 1323–1335.
[35] W. Lipo, Z. Nina, C. Feng, A general wrapper approach to selection of class-dependent features, neural networks, IEEE Trans. 19 (2008) 1267–1278.
[36] V.H.J.H. Yang, Feature subset selection using a genetic algorithm, IEEE, Intell. Syst. Appl. 13 (1998) 44–49.
[37] A. Ghosh, A. Datta, S. Ghosh, Self-adaptive differential evolution for feature selection in hyperspectral image data, Appl. Soft Comput. 13 (2013) 1969–1977.
[38] D. Chakraborty, N.R. Pal, A neuro-fuzzy scheme for simultaneous feature selection and fuzzy rule-based classification, neural networks, IEEE Trans. 15 (2004) 110–123.
[39] J. Huang, Y. Cai, X. Xu, A hybrid genetic algorithm for feature selection wrapper based on mutual information, Pattern Recogn. Lett. 28 (2007) 1825–1844.
[40] O. Il-Seok, L. Jin-Seon, M. Byung-Ro, Hybrid genetic algorithms for feature selection, pattern analysis and machine intelligence, IEEE Trans. 26 (2004) 1424–1437.
[41] M.M. Kabir, M. Shahjahan, K. Murase, A new hybrid ant colony optimization algorithm for feature selection, Expert Syst. Appl. 39 (2012) 3747–3763.
[42] R.F.a.A.M.a.A. Keikhab, Article a novel approach for feature selection based on the Bee colony optimization, Int. J. Comput. Appl. 43 (2012) 13–16.
[43] C. Lai, M.J.T. Reinders, L. Wessels, Random subspace method for multivariate feature selection, Pattern Recogn. Lett. 27 (2006) 1067–1076.
[44] M. Dash, H. Liu, Feature selection for classification, Intell. Data Anal. 1 (1997) 131–156.
[45] L. Huan, Y. Lei, Toward integrating feature selection algorithms for classification and clustering, Knowledge and Data Engineering, IEEE Trans. 17 (2005) 491–502.
[46] Y. Saeys, I. Inza, P. Larrañaga, A review of feature selection techniques in bioinformatics, Bioinformatics 23 (2007) 2507–2517.
[47] M.M. Kabir, M.M. Islam, K. Murase, A new wrapper feature selection approach using neural network, Neurocomputer 73 (2010) 3273–3283.
[48] I. Guyon, J. Weston, S. Barnhill, V. Vapnik, Gene selection for cancer classification using support vector machines, Mach. Learn. 46 (2002) 389–422.
[49] T.M. Hamdani, J.-M. Won, A.M. Alimi, F. Karray, Hierarchical genetic algorithm with new evaluation function and bi-coded representation for the selection of features considering their confidence rate, Appl. Soft Comput. 11 (2011) 2501–2509.
[50] M. Rostami, P. Moradi, A clustering based genetic algorithm for feature selection, in: In: Information and Knowledge Technology (IKT), 2014 6th Conference on, IEEE, 2014, pp. 112–116.
[51] L.-Y. Chuang, S.-W. Tsai, C.-H. Yang, Improved binary particle swarm optimization using catfish effect for feature selection, Expert Syst. Appl. 38 (2011) 12699–12707.
[52] L.-Y. Chuang, C.-H. Yang, J.-C. Li, Chaotic maps based on binary particle swarm optimization for feature selection, Appl. Soft Comput. 11 (2011) 239–248.
[53] M. Clerc, J. Kennedy, The particle swarm - explosion, stability, and convergence in a multidimensional complex space, evolutionary computation, IEEE Trans. 6 (2002) 58–73.
[54] Y. Liu, G. Wang, H. Chen, H. Dong, X. Zhu, S. Wang, An improved particle swarm optimization for feature selection, J. Bionic Eng. 8 (2011) 191–200.
[55] S.M. Vieira, L.F. Mendonça, G.J. Farinha, J.M.C. Sousa, Modified binary PSO for feature selection using SVM applied to mortality prediction of septic patients, Appl. Soft Comput. 13 (2013) 3494–3504.
[56] B. Xue, M. Zhang, W.N. Browne, Particle swarm optimisation for feature selection in classification: Novel initialisation and updating mechanisms, Appl. Soft Comput. (2014).
[57] C.-L. Huang, J.-F. Dun, A distributed PSO–SVM hybrid system with feature selection and parameter optimization, Appl. Soft Comput. 8 (2008) 1381–1391.
[58] E. García-Gonzalo, J.L. Fernández-Martínez, A brief historical review of particle swarm optimization (PSO), J. Bioinformatics Intell. Control 1 (2012) 3–16.
[59] L. Ali, S.L. Sabat, Particle swarm optimization based universal solver for global optimization, J. Bioinformatics Intell. Control 1 (2012) 95–105.
[60] M. Salehi Maleh, S. Soleymani, R. Rasouli Nezhad, N. Ghadimi, Using particle swarm optimization algorithm based on multi-objective function in reconfigured system for optimal placement of distributed generation, J. Bioinformatics Intell. Control 2 (2013) 119–124.
[61] X.J.C.Z.H. Cui, J.C. Zeng, Y.F. Yin, PID-controlled particle swarm optimization, J. Multiple-Valued Logic Soft Comput. 16 (2010) 585–610.

[62] C. Priya, P. Lakshmi, Particle swarm optimisation applied to real time control of spherical tank system, Int. J. Bio-Inspired Comput. 4 (2012) 206–216.
[63] H.M. Abdelsalam, A.M. Mohamed, Optimal sequencing of design projects' activities using discrete particle swarm optimisation, Int. J. Bio-Inspired Comput. 4 (2012) 100–110.
[64] M.H. Aghdam, N. Ghasem-Aghaee, M.E. Basiri, Text feature selection using ant colony optimization, Expert Syst. Appl. 36 (2009) 6843–6853.
[65] S.-W. Lin, T.-Y. Tseng, S.-Y. Chou, S.-C. Chen, A simulated-annealing-based approach for simultaneous parameter optimization and feature selection of back-propagation networks, Expert Syst. Appl. 34 (2008) 1491–1499.
[66] R. Meiri, J. Zahavi, Using simulated annealing to optimize the feature selection problem in marketing applications, Eur. J. Oper. Res. 171 (2006) 842–858.
[67] S.-W. Lin, Z.-J. Lee, S.-C. Chen, T.-Y. Tseng, Parameter determination of support vector machine and feature selection using simulated annealing approach, Appl. Soft Comput. 8 (2008) 1505–1512.
[68] R. Panda, M.K. Naik, B.K. Panigrahi, Face recognition using bacterial foraging strategy, Swarm Evol. Comput. 1 (2011) 138–146.
[69] J.-F. Chang, A performance comparison between genetic algorithms and particle swarm optimization applied in constructing equity portfolios, Int. J. Innovative Comput. Inform. Control 5 (December) (2009) 5069–5079.
[70] E. Talbi, L. Jourdan, J. Garcia-Nieto, E. Alba, Comparison of population based metaheuristics for feature selection: application to microarray data classification, in: In: Computer Systems and Applications, 2008. AICCSA 2008 IEEE/ACS International Conference on, 2008, pp. 45–52.
[71] S.-W. Lin, K.-C. Ying, S.-C. Chen, Z.-J. Lee, Particle swarm optimization for parameter determination and feature selection of support vector machines, Expert Syst. Appl. 35 (2008) 1817–1824.
[72] H.J. Escalante, M. Montes, L.E. Sucar, Particle swarm model selection, J. Mach. Learn. Res. 10 (2009) 405–440.
[73] L. Yu, H. Liu, Feature selection for high-dimensional data: a fast correlation-based filter solution, in: In: Proceedings of the 20th International Conference on Machine Learning, 2003, pp. 856–863.
[74] T.M. Mitchell, Machine Learning, McGraw-Hill, NewYork, 1997.
[75] S. Theodoridis, K. Koutroumbas, Pattern Recognition, Academic Press, Oxford, 2008.
[76] L.E. Raileanu, K. Stoffel, Theoretical comparison between the Gini index and information gain criteria, Ann. Math. Artif. Intell. 41 (2004) 77–93.
[77] H. Xiaofei, P. Deng Cai, Niyogi1, Laplacian score for feature selection, Adv. Neural Inf. Process. Syst. 18 (2005) 507–514.
[78] Q. Gu, Z. Li, J. Han, Generalized fisher score for feature selection, in: In: Proceedings of the International Conference on Uncertainty in Artificial Intelligence, 2011.
[79] H. Peng, F. Long, C. Ding, Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy, IEEE Trans. Pattern Anal. Mach. Intell. 27 (2005) 1226–1238.
[80] A.J. Ferreira, M.A.T. Figueiredo, An unsupervised approach to feature discretization and selection, Pattern Recogn. 45 (2012) 3048–3060.
[81] R. Sikora, S. Piramuthu, Framework for efficient feature selection in genetic algorithm based data mining, Eur. J. Oper. Res. 180 (2007) 723–737.
[82] J. Yang, V. Honavar, Feature subset selection using a genetic algorithm, IEEE Intell. Syst. Appl. 13 (1998) 44–49.
[83] M.E. Farmer, S. Bapna, A.K. Jain, Large scale feature selection using modified random mutation hill climbing, in: In: 17th International Conference on Pattern Recognition, 2004, pp. 287–290.
[84] D.B. Skalak, Prototype and feature selection by sampling and random mutation hill climbing algorithms, in: In: 11th International Conference on Machine Learning, 1994, pp. 293–301.
[85] R. Forsati, A. Moayedikia, A. Keikha, A novel approach for feature selection based on the bee colony optimization, Int. J. Comput. Appl. 43 (2012).
[86] H. Mauricio Schiezaro, Pedrini, Data feature selection based on artificial bee colony algorithm, EURASIP J. Image Video Process. 47 (2013) 2013.
[87] D.K. Sanjay Singla, H.M. Rai, S. Priti, A hybrid PSO approach to automate test data generation for data flow coverage with dominance concepts, Int. J. Adv. Sci. Technol. 37 (December) (2011) 15–26.
[88] S.M. Bhimsen Tudu, K. Kamal, Mandal, C. Niladri, Comparative performance study of genetic algorithm and particle swarm optimization applied on off-grid renewable hybrid energy system, in: SEMCCO'11 Proceedings of the Second international conference on Swarm, Evolutionary, and Memetic Computing, Part I (Berlin, Heidelberg), 2011, pp. 151–158.
[89] J. Kennedy, R. Eberhart, Particle swarm optimization, in: In: Proceedings IEEE International Conference on Neural Networks, 1995, pp. 1942–1948.
[90] R. Eberhart, J. Kennedy, A new optimizer using particle swarm theory, in: in: Micro Machine and Human Science, 1995 MHS '95., Proceedings of the Sixth International Symposium on, 1995, pp. 39–43.
[91] E. Elbeltagi, T. Hegazy, D. Grierson, Comparison among five evolutionary-based optimization algorithms, Adv. Eng. Inform. 19 (2005) 43–53.
[92] B. Xue, M. Zhang, W.N. Browne, Particle swarm optimization for feature selection in classification: a multi-objective approach, IEEE Trans. Cyb. 43 (2013) 1656–1671.
[93] B. Xue, M. Zhang, W.N. Browne, Particle swarm optimisation for feature selection in classification: novel initialisation and updating mechanisms, Appl. Soft Comput. 18 (2014) 261–276.
[94] H.H. Inbarani, A.T. Azar, G. Jothi, Supervised hybrid feature selection based on PSO and rough sets for medical diagnosis, Comput. Meth. Prog. Biomed. 113 (2014) 175–185.
[95] J. Jiang, Y. Bo, C. Song, L. Bao, Hybrid algorithm based on particle swarm optimization and artificial fish swarm algorithm, in: J. Wang, G. Yen, M. Polycarpou (Eds.), Advances in Neural Networks–ISNN 2012, Springer Berlin Heidelberg, 2012, pp. 607–614.
[96] P.-C. Chang, J.-J. Lin, C.-H. Liu, An attribute weight assignment and particle swarm optimization algorithm for medical database classifications, Comput. Meth. Prog. Biomed. 107 (2012) 382–392.
[97] M.C. Fu, F.W. Glover, J. April, Simulation optimization: a review, new developments, and applications, in: In: Simulation Conference, 2005 Proceedings of the Winter, 2005, p. 13.
[98] A. Boubezoul, S. Paris, Application of global optimization methods to model and feature selection, Pattern Recogn. 45 (2012) 3676–3686.
[99] J. Kennedy, R.C. Eberhart, A discrete binary version of the particle swarm algorithm, In: Systems, Man, and Cybernetics, 1997. Computational Cybernetics and Simulation., 1997 IEEE International Conference on 4105 (1997) 4104–4108.
[100] F.V.D. Bergh, An Analysis of Particle Swarm Optimizers, University of Pretoria, 2002, pp. 1.
[101] M.M. Kabir, M. Shahjahan, K. Murase, A new local search based hybrid genetic algorithm for feature selection, Neurocomputing 74 (2011) 2914–2928.
[102] P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, J. Riedl, GroupLens. An open architecture for collaborative filtering of netnews, in: In Proceedings of the ACM Conference on Computer Supported Cooperative Work (CSCW'94), 1994, pp. 175–186.
[103] M. Stone, Cross validation choice and assessment of statistical predictions, J. R. Stat. Soc. B 36 (1974) 111–147.
[104] S.H.D.J. Newman, C.L. Blake, C.J. Merz, UCI Repository of Machine Learning Databases, 1998, http://www.ics.uci.edu/%08mlearn.
[105] L.-Y. Chuang, C.-S. Yang, K.-C. Wu, C.-H. Yang, Gene selection and classification using Taguchi chaotic binary particle swarm optimization, Expert Syst. Appl. 38 (2011) 13367–13377.
[106] S. Yu, Z. Wu, H. Wang, Z. Chen, A hybrid particle swarm optimization algorithm based on space transformation search and a modified velocity model, in: W. Zhang, Z. Chen, C. Douglas, W. Tong (Eds.), High Performance Computing and Applications, Springer Berlin Heidelberg, 2010, pp. 522–527.
[107] A.K.A. (Ed.), Machine Recognition of Patterns, IEEE Press, New York, 1977.
[108] M. Friedman, A comparison of alternative tests of significance for the problem of m rankings, Ann. Math. Stat. 11 (1940) 86–92.