Int. J. Nonlinear Anal. Appl. 15 (2024) 9, 271–287
ISSN: 2008-6822 (electronic)
http://dx.doi.org/10.22075/ijnaa.2023.30515.4421

Feature selection method based on clustering technique and optimization algorithm

Sara Dehghani, Razieh Malekhosseini*, Karamollah Bagherifard, S. Hadi Yaghoubyan

Department of Computer Engineering, Yasuj Branch, Islamic Azad University, Yasuj, Iran

(Communicated by Seyed Hossein Siadati)

Abstract

Data platforms with large dimensions, despite the opportunities they create, create many computational challenges. One of the problems of data with large dimensions is that most of the time, all the characteristics of the data are not important and vital to finding the knowledge that is hidden in them. These features can have a negative effect on the performance of the classification system. An important technique to overcome this problem is feature selection. During the feature selection process, a subset of primary features is selected by removing irrelevant and redundant features. In this article, a hierarchical algorithm based on the coverage solution will be presented, which selects effective features by using relationships between features and clustering techniques. This new method is named GCPSO, which is based on the optimization algorithm and selects the appropriate features by using the feature clustering technique. The feature clustering method presented in this article is different from previous algorithms. In this method, instead of using traditional clustering models, final clusters are formed by using the graphic structure of features and relationships between features. The UCI database has been used to evaluate the proposed method due to its extensive characteristics. The efficiency of the proposed model has also been compared with the feature selection methods based on the coverage solution that uses evolutionary algorithms in the feature selection process. The obtained results indicate that the proposed method has performed well in terms of choosing the optimal subset and classification accuracy on all data sets and in comparison with other methods.

Keywords: feature selection, optimization algorithms, hierarchical algorithm, graph clustering
2020 MSC: 05C85, 65Yxx

1 Introduction

New information and communication technologies as well as decision support technologies, by collecting, storing, evaluating, interpreting and analyzing, retrieving and disseminating information and knowledge to specific users, can provide timely, accurate and needed information to people. have a lot of influence. One of the tools used in these technologies is data mining. Data mining includes the use of advanced data analysis tools in order to discover reliable and previously unknown patterns in large data sets. These tools are statistical models, mathematical algorithms and machine learning methods. Machine learning algorithms are algorithms that automatically improve their performance through experience [40, 6, 21, 43, 4, 27, 28].

*Corresponding author
Email addresses: saradehghani18@gmail.com. (Sara Dehghani), malekhoseini.r@gmail.com (Razieh Malekhosseini), ka.bagherifard@iau.ac.ir (Karamollah Bagherifard), yaghoobian.h@gmail.com (S. Hadi Yaghoubyan)

Received: February 2023 *Accepted:* June 2023

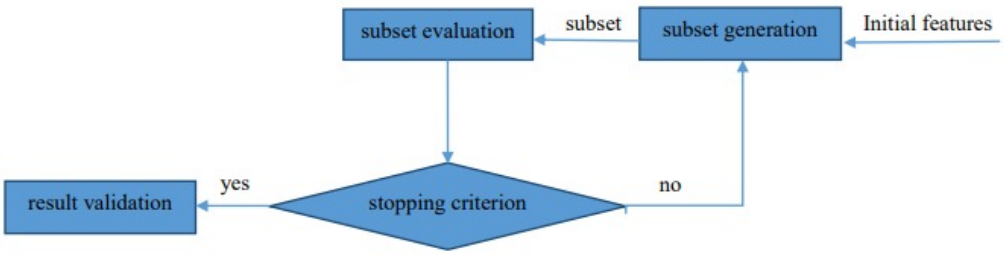

Figure 1: The main stages of the feature selection process [19].

Data mining is beyond data collection and management and includes analysis and prediction. In data mining, it is usually referred to the discovery of useful patterns among the data. A useful model is a model in data that describes the relationship between a subset of data that is valid, simple, understandable and new. Data mining is not a new technology, but its use has been growing meaningfully in various private and public sectors, and generally industries such as banking, insurance, medicine, and retail use data mining for the purpose of reducing costs, increasing research, and increasing sales [28, 22, 41, 15, 1, 32].

Data is usually described by a large number of attributes. Many of these features may be irrelevant and redundant for the intended data mining application. The existence of a large number of these irrelevant and redundant features in a data set has a negative effect on the performance of the machine learning algorithm and also increases the computational complexity [18, 16, 34]. Therefore, reducing the dimensions of a dataset is a fundamental task in data mining and machine learning applications. Decreasing the dimensions of the data set on one hand reduces the computational complexity and on the other hand, reduces the parameters of the classification algorithm. As a result, it is possible that the performance of the classification algorithm will increase. Therefore, it seems that the model based on the reduced features has a higher generalization ability than the original model. According to a general rule for a classification problem with n dimensions and C classes, at least $10\times n\times C$ training data are required [32, 18, 9]. When it is practically impossible to provide this amount of training data; Reducing the number of features reduces the amount of training data required. As a result, the performance of the classification algorithm increases.

In recent years, two general strategies have been proposed to reduce the dimension. feature selection and feature extraction [18, 2, 12]. Feature selection, also known as variable selection and subset selection, selects a subset of the initial features by searching through the available subsets. While in feature extraction, the primary features are transferred to a new space with less dimensions. During the feature selection process, a subset of primary features is selected by removing irrelevant and redundant features. The entire search space to find the most suitable features, including all possible subsets, is defined as:

$$\sum_{s=0}^{n} \binom{s}{n} = \binom{0}{n} + \binom{1}{n} + \cdots + \binom{n}{n} = 2^n \tag{1.1}$$

where n is the number of initial features and s is the size of the selected feature subset.

In general, the feature selection process includes four main stages of feature subset generation, subset evaluation, stopping criteria, and validation of results, as shown in Figure 1.

As seen in Figure 1, in each iteration of the search process, a candidate subset of the main features is generated and its suitability is measured by an evaluation criterion. The process of generating the subset and evaluating it is repeated until a predetermined stopping criterion is reached. At the end of this process, the best selected feature subset is validated on the test data set. Another way to reduce data dimensions is to use feature clustering. Feature selection based on the relationships between features is an old idea to reduce the dimensions of the problem, which has been considered for a long time [18, 2, 12]. Graphical representation of the problem has gained a lot of popularity in the last decade. This representation method provides a robust model for problem representation thanks to its ability to model relationships between problem elements. In this article, by modeling the problem using feature clustering, it is possible to better display the relationships between features and, as a result, select the final features. As research contributions, we seek to find a suitable answer to the following questions.

1. To what extent will the use of feature clustering affect the selection of related features?
2. To what extent can the use of feature clustering reduce redundancy between selected features?

3. To what extent can the use of optimization algorithm increase the accuracy of feature selection?

In fact, it seems that the use of feature clustering can be effective in selecting related features. In addition, the use of feature clustering can reduce the redundancy between selected features. Also, it can be claimed that the use of optimization algorithm can increase the accuracy of feature selection to some extent. The remainder of this article is organized as follows. First, in section 2, the research conducted in the field of feature selection is reviewed. Then Section 3 describes the proposed feature selection method in full detail. The performance of the proposed method for the feature selection problem is evaluated in the next sections, and also, the performance of the proposed method is compared with the latest feature selection methods based on the overlay solution. And finally, we will give a general summary of the method presented in this article.

2 Research literature

An important issue related to data mining applications is the problem of high dimensionality of the data set where the number of features is much more than the number of patterns. Data sets with high dimensions reduce the performance of the classifier in two ways. On the one hand, the volume of calculations increases, and on the other hand, a model based on high-dimensional data has a low generalization capability and the probability of overfitting increases [18]. As a result, reducing the dimensions of the problem can both reduce the computational complexity and improve the performance of classification algorithms. In recent years, two general solutions have been proposed to reduce the dimension. feature selection and feature extraction [18, 2, 12].

From a general point of view, feature selection methods are divided into two categories with and without supervision [17, 30, 11]. In supervised methods, a set of training patterns is available, each pattern is described by a vector of feature values along with a class label, while unsupervised methods are faced with a dataset without class labels. In general, it can be said that feature selection methods in supervised mode have better efficiency and more reliable performance due to the use of class labels [30, 39]. Therefore, feature selection in unsupervised mode is more difficult and in many researches, this field has been considered. Each feature selection method consists of two main steps of creating candidate subsets and evaluating these subsets. Different subsets are generated according to the search strategy and their usefulness is calculated based on the evaluation criteria. These two steps are repeated until the stop criterion is reached.

Also, based on the evaluation criteria, feature selection methods are divided into four solutions: wrapper, filter, hybrid and embedded [9, 30].

2.1 Wrapper approach

The Wrapper approach uses a learning or classifier algorithm to evaluate the suitability of the selected feature subset. In this approach, a search method is used to find the optimal feature subset. Therefore, at each stage of the search process, a feature subset is generated and the quality of that subset is evaluated by training and testing a classifier. Finally, the best generated feature subset is selected as the final feature subset. For a specific data set, the wrapper approach is started by generating a subset of features and using a learning algorithm, its usefulness is evaluated. The search for the optimal subset is repeated until a predetermined stopping criterion is reached. After reaching this criterion, the search is stopped and the current subset is selected as the final subset and its quality is evaluated on the test data. The wrapper approach is divided into two general categories, sequential and random, based on the search strategy.

The method (PSOFS) [31] based on the particle swarm optimization algorithm and by providing three new initialization strategies, three new update mechanisms, searches for the optimal feature subset. The disadvantages of this method are the selection of the largest number of features and low classification accuracy.

2.2 Filter approach

The filter approach uses the statistical and probabilistic features of the data to select the feature and performs the feature selection process independently of the machine learning algorithms. In other words, this approach uses the inherent characteristics of the data to evaluate the features. Due to the fact that the filter approach does not include machine learning algorithms, it is computationally much faster than the methods based on the Wrapper approach, and it will be suitable for high-dimensional datasets. Figure 2 shows the general outline of this solution. The overall scheme of the filter approach is similar to the Wrapper approach, with the difference that here the learning algorithm is not used to evaluate the subset of the generated features and different subsets are evaluated based on an independent criterion

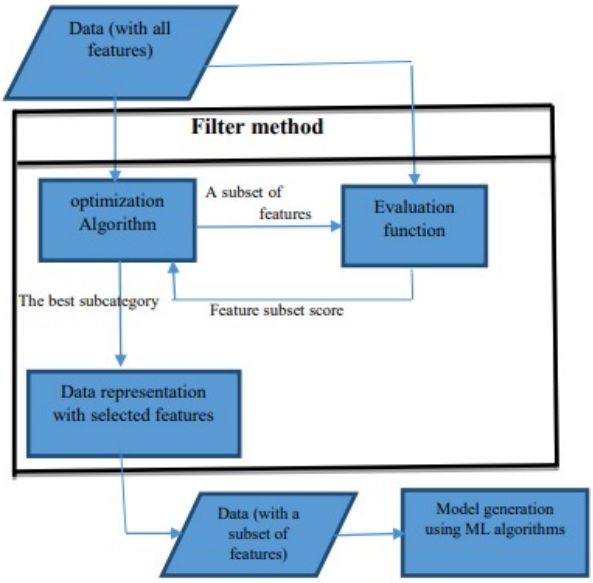

Figure 2: Overview of the filter approach

2.2.1 Univariate methods

In univariate methods, the appropriateness of each feature is evaluated alone and according to a certain criterion. In these methods, it is assumed that the features are independent of each other and the possible dependence between the features is not considered. This simplifying assumption may be a wrong assumption in many cases and reduce the efficiency of the feature selection method. The advantage of these methods is their simplicity and low computational complexity. In the following, a number of univariate methods based on the filter solution are briefly reviewed.

Information gain (IG) [26] is a method based on information theory that is often used in machine learning. Usually, information gain is defined as the amount of information provided by a feature for the classification system. This method is independent of the classifier and efficiently removes irrelevant features, but it is able to select features only in the case with the observer and has relatively high computational complexity.

Fisher's score (FS) [25] is a supervised feature selection method, which aims to select a subset of features, where the subset is the distance between patterns in the same class, as small as possible, and the distance between patterns in different classes, as much as possible. In other words, this criterion determines the ratio of the dispersion of patterns between different classes to the dispersion of patterns within each class. Therefore, this criterion gives a higher score to the features that have such a separating property. But this method has low usefulness due to ignoring the relationships between features.

Laplacian score (LS) [38] is a method based on graph theory that works in both supervised and unsupervised modes. The Laplacian score models the data space into a graph and is based on the belief that if two data points are close to each other, they are likely to belong to the same class. In fact, this method uses the local structure of the data space to select the feature. But in this method, it is possible to choose similar and redundant features, and also the performance of this method is relatively low.

2.2.2 Multivariate methods

Multivariate methods evaluate the suitability of features according to the dependence between them. Therefore, multivariate methods have more computational complexity than univariate methods, but they also have higher performance. Multivariate methods tend to overfit when the number of patterns is much less than the number of features [27]. In the following, the methods of multivariate feature selection based on the filter solution are briefly reviewed.

The random subspace method (RSM) [10] is a multivariate feature selection method based on the filter solution, whose main goal is to reduce the computational complexity in feature determination. This method is highly efficient due to considering the similarity between the features, but because it pays more attention to reducing the similarity between the selected features, this in turn causes the selection of unrelated features, and as a result, there is a possibility

of lowering the accuracy.

In [37], a feature selection method based on GAFS genetic algorithm is presented. This is a new method of multiple populations in the genetic algorithm, which has increased the efficiency of this method compared to previous methods. But in this method, in high-dimensional datasets, the convergence speed of the algorithm will decrease to a great extent. The convergence speed of the algorithm will be greatly reduced.

2.3 Hybrid approach

In the hybrid approach, an attempt is made to use the advantage of both filtering and wrapper approach and to provide an algorithm that balances the calculation efficiency in the filtering approach and the accuracy in the wrapper approach. In fact, the goal of this approach is to provide a method that performs well both in terms of efficiency and usefulness. In many feature selection methods based on the hybrid approach, the feature selection process is performed in two stages. In the first step, the initial feature set is reduced by the filter approach , then in the next step, the final feature set is selected from the reduced feature set using the wrapper approach.

In [9], a feature selection algorithm based on fuzzy theory is presented, which combines two solutions, covering and filtering. The method presented by them included three main steps:

1. Pre-processing of features using feature discretization process.
2. Pre-selection and ranking of features based on fuzzy random forest classifier.
3. Selecting the final features using a method based on the coverage solution.

In [29], a feature selection method based on a hybrid solution for high-dimensional data sets is presented. To select the final features in the proposed method, first, the set of candidate features is selected from among the initial features. Then, in the next step, the final feature sets are selected using an overlay method. Due to its two-stage nature, there is a possibility of removing suitable features in the initial stage.

2.4 Embedded approach

In the embedded approach , the feature selection process is considered as a part of the learning algorithm. In other words, the search for the appropriate subset of features is performed by a learning algorithm. The computational cost of the embedded approach is between the filter approach and the wrapper approach. As mentioned earlier, in the wrapper approach, classification accuracy is used in a predetermined learning algorithm to evaluate each candidate subset. One of the major problems of the methods based on the wrapper approach is their computational complexity. To solve this problem, in the embedded approach, it has been tried to reduce the computational time by combining feature selection with the training process. On the other hand, the embedded approach, like the wrapper approach, depends on the type of learning algorithm used during the feature selection process.

Support vector machine, decision tree and simple Bayes are the most famous learning algorithms for feature selection in embedded solutions. Support vector machine is one of the supervised learning methods used for classification and regression. This method is among relatively new methods that have shown good performance in recent years compared to older methods, including perceptron neural networks.

In [14], SVM has been used for gene selection in the classification of cancer types. They applied a technique called recursive feature elimination, which is a backward recursive feature selection technique. This method has less computational complexity than the coverage methods, but it has a higher computational complexity than the filter methods, and also in this method, there is dependence on the SVM classifier.

Graphical representation of the problem has gained a lot of popularity in the last decade. This representation method provides a robust model for problem representation thanks to its ability to model relationships between problem elements. In many past researches, graph-based methods have been used to solve the feature selection problem [33, 7, 42]. For example, in [42] a method based on hypergraph clustering is presented for the problem of feature selection. The method presented in this research has two major advantages. The first advantage of this method is to use the MII criterion to calculate the size of supergraph edges. Using this criterion makes it possible to identify higher-order relationships between features and to minimize the redundancy between the selected features. The second advantage of this method is the use of hypergraph clustering to select suitable features, which has made the number of final features automatically determined.

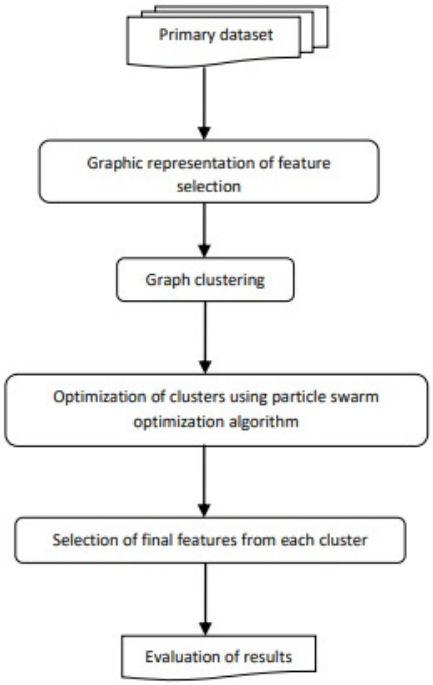

Figure 3: Working process of the proposed method

3 Research method

The main challenge in feature selection is the issue of redundant feature selection. The main goal of feature selection methods is to identify and remove these features. Some feature selection methods are able to effectively remove irrelevant features, but due to not considering the relationship between features, they are unable to identify redundant features. On the other hand, another category of feature selection methods only focuses on detecting and removing redundant features and irrelevant features are not removed in the feature selection process. Providing a feature selection method that can effectively identify both types of these features is one of the major challenges in feature selection.

In feature selection methods, determining the appropriate search algorithm plays a vital role. A feature selection method should be evaluated from two viewpoints of efficiency and usefulness. The efficiency of a feature selection method depends on the time required to find the final feature subset. While the usefulness is dependent on the quality of the selected feature subset. These two criteria have been in conflict with each other, and usually the improvement of one of them leads to a departure from the other. Therefore, creating a compromise between these two criteria has become an important and necessary issue in feature selection.

One of the main goals in feature selection is to select a suitable subset of features that is neither too large nor too small. If the number of features if selected is small, those features are not able to represent the entire primary features well, and as a result, the performance of the classification algorithm decreases. Also, if the number of selected features is too large, the probability of selecting unrelated and redundant features increases. As a result, the performance of the classification algorithm decreases. In most feature selection methods, the number of selected features is considered fixed and must be determined by the user before the feature selection process. Therefore, providing a feature selection method in which the number of suitable features is automatically determined is one of the major challenges in the feature selection problem.

3.1 Suggested method

In this paper, a new algorithm based on relationships between features will be presented for feature selection. Also, a new feature selection method has been presented by using the particle swarm optimization algorithm and feature clustering.

Figure 3 shows the process of the proposed method, each of these steps is described below. In this proposed method

to represent the feature clustering problem, it is done in such a way that at first, graph clustering is done using the relationships between features, and then, based on the new clusters, the structure of each particle is determined in the particle optimization algorithm. and finally the final clusters are selected.

3.1.1 Graphic representation of feature selection

To cluster the features using the graph clustering algorithm, the feature space must be represented graphically. Therefore, the problem is represented as a complete undirected weighted graph $G = (F, E, w_F)$, where $F = \{F_1, F_2, ..., F_n\}$ represents the initial feature set with n features, each feature is a node of the graph and $E = \{(F_i, F_j) : F_i, F_j \in F\}$represents the edges of the graph. Also, $w_F : (F_i, F_j) \longrightarrow \mathbb{R}$ is a function that shows the degree of similarity between two features F_i and F_j.

One of the important issues in feature clustering process is to determine a criterion to calculate the similarity between two features. Choosing an appropriate criterion to calculate the similarity between features has a great impact on the performance of the feature selection algorithm. There are different methods for calculating the similarity between features, each of which gives different results. The degree of similarity between two features can be calculated based on the distance between the two feature vectors. Considering that there are generally two measures of similarity and photo distance, if we can provide a distance measure for two feature vectors, by reversing this value, we will be able to calculate the similarity between the two feature vectors.

In this article, an attempt is made to provide a new criterion that has both high accuracy and less computational complexity. In this similarity criterion [24], an attempt has been made to improve the problem of high computational complexity in the cosine similarity coefficient, which is caused by the multiplication of features, or the problem of calculating the difference between the average and the features of each data in the Pearson similarity coefficient. For this purpose, instead of using the cosine criterion or the Pearson criterion, a new criterion based on the covariance vector is provided. This relationship is introduced as follows:

$$Sim(S_i, S_j) = \frac{Cov(S_i)Cov(S_j)}{\sqrt{Var(S_i)Var(S_j)}}. \tag{3.1}$$

In the above relation, $Cov(S_i)$ represents the covariance of the feature vector x_i and also $Var(S_i)$, represents the variance calculation for the feature vector x_i. As can be seen in this relationship, if two features are completely similar, in this case, the degree of similarity will be equal to 1 or -1, and two features that are completely independent of each other, in this case, the degree of similarity will be equal to zero.

Since most programming languages have library functions for calculating variance and covariance, as a result, the computational complexity of implementing this part has been almost eliminated, and due to the use of internal functions in programming languages, the time to implement this method Compared to other similarity calculation criteria, it is much lower. Also, due to the simultaneous use of the two concepts of covariance and variance, which indicates the amount of dispersion in that feature, the accuracy of the method will increase compared to similar methods. On the other hand, considering that the value of this feature will always be equal to 1 and -1 at the highest degree of similarity and equal to zero at the lowest degree of similarity, as a result, unlike Pearson's criterion, there is no need for normalization and only by taking the absolute value of the The obtained similarity, the weight of the corresponding edge in the graph is determined. Due to the elimination of the normalization step, the similarity calculation method presented in this article will be much more efficient than the previous examples.

3.1.2 Clustering of features

The main purpose of feature clustering is to divide the primary features into a number of different clusters based on their similarity to each other. In most feature clustering methods, the number of clusters must be specified by the user before performing the clustering algorithm. In general, determining the number of clusters for primary features is a difficult task, and the optimal number of clusters can only be determined by trial and error.

Data distribution in a cluster is considered one of the important criteria in clustering, which is not considered in most of the previously presented methods for feature clustering. Considering the dispersion of features in a cluster can greatly increase the performance of the clustering algorithm.

In this method, to deal with these problems, a community detection algorithm called Louvain [8] has been used for the initial clustering of features. The purpose of using the Louvain algorithm is that the features are divided into a number of clusters that each cluster has similarities with each other, and then appropriate features are extracted from each cluster. One of the fast and efficient algorithms for detecting communities is the Louvain algorithm, which

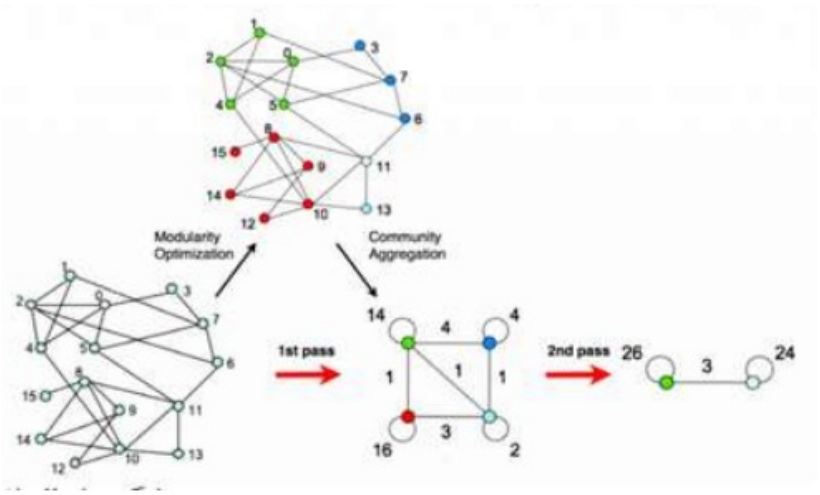

Figure 4: Steps of feature clustering using Louvain algorithm

performs graph clustering using Podmani function maximization. At the beginning of the algorithm, each node is considered as a cluster, and then clustering is done in two repeated steps as follows:

First step: for each node i, the benefit of assigning that node to cluster C is calculated using equation (3.2)

$$\Delta Q = \left[\frac{\sum_{\text{in}} + k_{i,\text{in}}}{2m} - \left(\frac{\sum_{\text{tot}} + k_i}{2m}\right)^2\right] - \left[\frac{\sum_{\text{in}}}{2m} - \left(\frac{\sum_{\text{tot}}}{2m}\right)^2 - \left(\frac{k_i}{2m}\right)^2\right] \tag{3.2}$$

where $\sum_{in}$ in is the sum of weights in cluster C, $\sum_{tot}$ is the sum of weights of edges that connect to the nodes of cluster C, k_i is the sum of edges of node i and $k_{i,in}$ represents the sum of weights of edges that have one end is node i and its other end is cluster C. Also, m is equal to the sum of the weights of all the edges of the graph.

Second step: each node is assigned to a cluster that maximizes the Podmani function. Clusters are then rebuilt based on this new structure. These two steps are repeated until there is no more change in the cluster structure. One of the advantages of the Louvain community detection algorithm is its simplicity and repeatability, which makes its analysis and implementation very easy. Also, the number of clusters is determined automatically and there is no need to have information about the data structure before clustering. In addition, in terms of computational complexity, the Louvain algorithm is very efficient and has a time complexity of O(n log n), where n represents the number of nodes. For this reason, this algorithm can be used for graphs with a very large number of nodes and up to several million nodes. Louvain community detection algorithm is used in many network analysis software. Figure 4 shows the clustering of features using the Louvain algorithm.

3.1.3 Optimization of clusters using particle swarm optimization algorithm

In the feature selection problem, many search algorithms based on different techniques have been proposed to find the global optimal solution in a reasonable time. But their computational complexity is exponential. Therefore, with the increase in the number of features, the execution time of these algorithms increases exponentially. So, solving this problem is impractical for high-dimensional datasets. This issue has made heuristic algorithms and meta-heuristic algorithms more interesting to researchers. The methods based on heuristic search increase the speed of the algorithm by compromising between the computational complexity and the quality of the found solution. These methods obtain the final solution in a reasonable time, but do not guarantee to obtain the global optimal solution. More precisely, it can be said that it is possible for these algorithms to get stuck in the local optimum. As a result, different algorithms with different ideas have tried to minimize this problem in order to find the general optimum which is the best subset of the main features. These algorithms try to find the optimal solution to the problem by searching the problem space and focusing on good solutions. These algorithms, which are named meta-heuristic algorithms, have been able to significantly reduce the probability of getting stuck in the local optimum by using this approach. Among the meta-heuristic methods that have been proposed for feature selection, population-based optimization algorithms, such as genetic algorithm (GA), ant colony optimization algorithm (ACO), and particle optimization algorithm (PSO), have received more attention. PSO has many advantages over other heuristic optimization methods. for example:

- This algorithm works based on probabilistic rules, not deterministic rules. Therefore, PSO is a stochastic optimization algorithm that can search uncertain and complex regions. This feature makes PSO more flexible and resistant than conventional methods.
- The quality of the solution of the proposed path does not depend on the initial population. Starting from any point in the search space, the algorithm eventually converges the solution to the optimal solution.
- PSO has great flexibility to control the balance between local and global search of the search space. This unique property of PSO overcomes the problem of untimely convergence and increases the search capacity. All these features make PSO different from genetic algorithm (GA) and other heuristic algorithms.

In the proposed method, by using feature clustering and combining it with the particle swarm optimization algorithm, a feature selection method based on the overlay solution is provided. The use of this method for feature selection results in the selection of features with maximum relevance and minimum redundancy.

The PSO method, which was first proposed by Eberhart and Kennedy in 1995 [19], is a global minimization method that can be used to deal with problems whose solution is a point or surface in the n-dimensional space. In such a space, an initial velocity is assigned to each particle, and communication channels between particles are also considered. These particles are then moved through the response space and the results are calculated based on a "merit criterion" after each time interval. With the passage of time, the particles accelerate towards the particles that have a higher merit criterion and are in the same communication group. Despite the fact that each method works well in a range of problems, the particle swarm optimization algorithm method has shown much success in solving continuous optimization problems.

In this part of the proposed method, using the particle swarm optimization algorithm with a new merit criterion and with the help of parallel mapping, the clusters formed in the previous step are optimized and the final clusters are formed. In this step, it is tried to optimize the centers of the clusters selected in the previous step. In fact, each particle in the particle optimization algorithm is considered as a mapping particle that defines the formed clusters.

The beginning of the particle swarm optimization algorithm is that a group of particles is created randomly and by updating the generations, they try to find the optimal solution. In each step, the position and speed of each particle is updated. Two items are used for this. The first case is the best position that the particle has achieved so far, which is known and stored as the pbest position and is used by the algorithm, and the second case is the best position that has been achieved by the population of particles. Is. This position is displayed by bagbest. In the particle swarm optimization algorithm, each particle in the search space is represented by the vector $\mathrm{x_i} = (\mathrm{x_{i1}}, \mathrm{x_{i2}}, \ldots, \mathrm{x_{iD}})$ where D represents the dimensions of the problem. Also, this particle has a velocity v, which is represented by the vector $\mathrm{v_i} = (\mathrm{v_{i1}}, \mathrm{v_{i2}}, \ldots, \mathrm{v_{iD}})$. In each iteration, after finding the best values (pbest and gbest), the velocity and position vector of each particle is updated using the following equations.

$$\mathrm{x} = x + v \tag{3.3}$$

$$v = v \times \omega + C_1 \times r_1 \times (pbest - x) + C_2 \times r_2 \times (gbest - x) \tag{3.4}$$

where ω is an input weight that determines the influence of the speed in the previous iteration on the current speed, also, $\mathrm{C_1}$ and $\mathrm{C_2}$ are two random values between zero and one. The used particle swarm optimization algorithm searches for the optimal feature subset in five steps, each of these steps is described below.

First step: Creating the initial population: The first step in any particle swarm optimization algorithm is to create an initial population of particles. Each particle in the particle swarm optimization algorithm represents a solution of the problem. So, in the algorithm used in this proposed method, each particle must show a subset of features. The length of each particle is equal to the size of the original features, n. Each feature of the particle specifies whether the corresponding feature is selected or not. In other words, each attribute is a binary value that can have two states of zero and one. A value of zero for each feature indicates that that feature is not selected, and a value of one indicates that the feature is selected. The restriction applied in creating the initial population is that the number of selected features in each particle must be the same for the entire initial population. This value is considered equal to K. In other words, in each particle of the initial population, only and only K features have a value of one. In other words, the length of all primitive subsets are equal.

Second step: calculation of the fitness function: after creating the initial population, the value of the fitness function

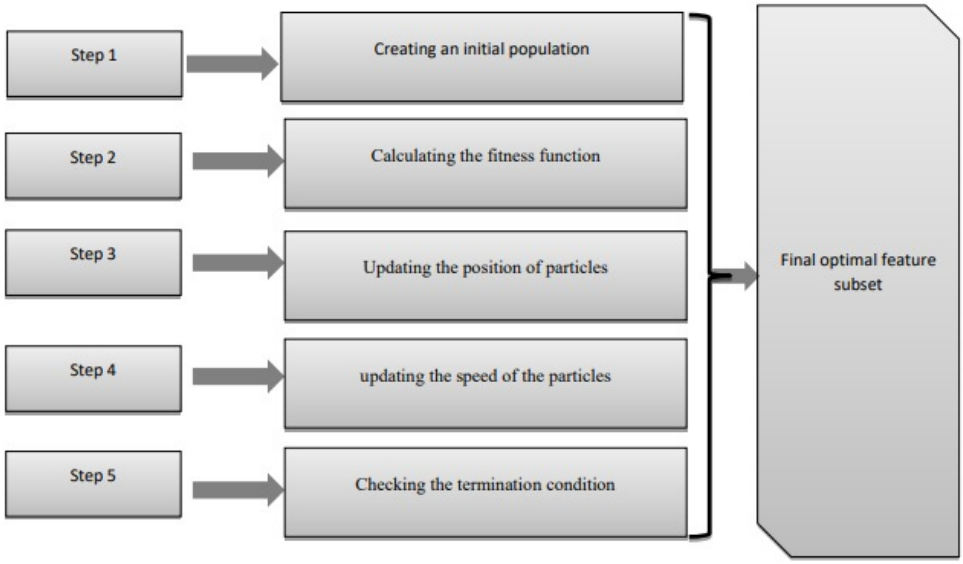

Figure 5: Flowchart of particle swarm optimization algorithm

for all particles should be calculated. For this purpose, in this proposed method, the following criterion is used.

$$Q = \sum_{s=1}^{m} \left[\frac{l_s}{m} - \left(\frac{d_s}{2m} \right)^2 \right] \tag{3.5}$$

where m is the number of primary particles, l_s is the number of input edges of particle s, and d_s is the sum of input and output degrees of particle s. For a fixed population c, the first term, i.e. $1_s/m$ is the fraction of edges of the graph that is input to the particle s, while the second term, i.e. $(d_s/2m)$ the fraction of edges expected in a graph Random with the same distribution of G degrees, belongs to the particle s.

Third step: updating the position of the particles: in this case, the position of the particles is updated based on the relation (3.3) In other words, in this case, the subset of features created by the particle swarm optimization algorithm is updated by this relationship.

Fourth step: updating the speed of the particles: in this step, the speed of the particles is updated, similar to the previous step. In fact, each particle (solution) in the particle swarm optimization algorithm has a speed and a position, which is updated using the relation (3.3) and then using the relation (3.4) particle velocity is updated.

Fifth step: check the termination condition: depending on the type of problem, the condition for stopping the algorithm can be a certain number of repetitions, time, lack of improvement of the answers, or lack of convergence of the obtained answers. In this stage of the proposed method, the termination condition, which is the number of iterations of the particle swarm optimization algorithm, is checked. If the predetermined number of repetitions is reached, the algorithm ends.

Figure 5 shows the flowchart of particle swarm optimization algorithm.

3.1.4 Selection of the final feature subset

The main goal of this step of the proposed method is to search for the optimal feature subset using the appropriateness criterion of each feature. The measure of suitability of each feature in the unsupervised mode is data dispersion (TV) [36] and in the supervised mode, Fisher's score (FS) [25].

Data dispersion is the simplest unsupervised measure for feature evaluation, the data dispersion in a feature indicates the representative power of that feature. Therefore, features with high dispersion have valuable information. Also, Fisher's score is an observational feature selection method whose goal is to select a subset of features in which the distance between patterns in the same class is as small as possible and the distance between patterns in different classes is as large as possible. In other words, this criterion determines the ratio of the dispersion of patterns between different classes to the dispersion of patterns within each class. Therefore, this criterion gives a higher score to the features

that have such a separating property. The Fisher score for feature A is determined using the following equation.

$$\mathrm{FS(S,A)} = \frac{\sum_{\mathrm{v}\in \mathrm{Values(S)}} \mathrm{n_v}(\bar{\mathrm{A}}_\mathrm{v} - \bar{\mathrm{A}})^2}{\sum_{\mathrm{v}\in \mathrm{Values(S)}} \mathrm{n_v}(\sigma_\mathrm{v}(\mathrm{A}))^2} \tag{3.6}$$

where $\bar{A}$ is the average of the entire set of patterns corresponding to feature A, $\mathrm{n_v}$ represents the number of patterns whose class label is v, and $\sigma_\mathrm{v}(\mathrm{A})$ and $\bar{\mathrm{A}}_\mathrm{v}$ represent the standard deviation and mean of patterns within class v, respectively, according to Attribute A specifies.

In this proposed method, after calculating the Fisher score for all the features, the features with the highest Fisher score are selected as the subset of the final features.

3.2 Evaluation of the proposed method

In this section, the performance of the proposed method for selecting features in big data is evaluated using a new algorithm based on clustering techniques and optimization algorithms. For this purpose, the proposed method is compared with the latest methods of feature selection in big data that use evolutionary algorithms in the feature selection process. The feature selection methods for comparison with the proposed methods are:

Hybrid Genetic Algorithm Based on Local Search for Feature Selection(HGAFS)[27]: This method selects the final feature subset by using a hybrid genetic algorithm and applying a local search operator. The HGAFS method evaluates a subset of candidate features using a multilayer perceptron neural network. Therefore, it is a feature selection method based on the coverage strategy.

Combined Ant Colony Optimization Algorithm for Feature Selection (ACOFS)[23]: This method searches for the optimal subset by using the search property of the ant colony algorithm and combining it with neural network.

Particle Swarm Optimization Algorithm for Feature Selection (PSOFS) [31]: This method searches for the optimal feature subset based on particle swarm optimization algorithm and by providing three new initialization strategies, three new update mechanisms. The main goal of PSOFS method is to select the least number of features and maximize the classification accuracy. This method is also based on the cover solution.

The feature selection method presented in this article is named GCPSO and MATLAB programming language was used to implement this method.

In the rest of this section, the characteristics of the data sets used in the experiments, the classifiers used, and the practical results are described.

3.2.1 Datasets

In this article, several datasets with different characteristics have been used to evaluate the proposed methods and compare its performance with other feature selection methods. These datasets include Wine, Hepatitis, Wisconsin Diagnostic Breast Cancer (WDBC), Ionosphere, Spambase, Sonar, Arcene, and Colon. All these data sets except the last data set (Colon) were selected from the University of California database [5]. Also, the specifications and details of the Colon dataset have been presented by Mr. Alon and his colleagues [3]. These datasets have been selected for evaluating the proposed methods due to their extensive characteristics. For example, the Spambase dataset is a small-dimensional dataset with a large number of patterns, but the Colon dataset has only a small number of patterns available despite its very high dimension. The general characteristics of these datasets are shown in Table 1.

In some of these datasets, different features have different value ranges. In this case, features with a larger value range may dominate over features with a smaller value range, and the probability of their selection will increase. To solve this problem, all the different datasets are normalized before starting the feature selection process. In this article, maximum-minimum normalization method is used to normalize data sets. Using this normalization method, the range of values of all used data sets is changed to the range of zero to one.

In some of these data sets, there are several missing values. To overcome this problem, the missing values in these features are replaced by the average of the available data corresponding to that feature that is available [35].

3.2.2 The size of the selected feature subset

In this section, different methods are compared in terms of the number of features they select as the final subset. Table 2 shows the average feature subset size selected by each of the methods on different datasets.

Table 1: Specifications of datasets

Dataset	Features	Classes	Patterns
Wine	13	3	178
Hepatitis	19	2	155
WDBC	30	2	569
Ionosphere	34	2	351
Spambase	57	2	4601
Sonar	60	2	208
Colon	2000	2	62
Arcene	10000	2	900

As it is clear from the results of Table 2, all the methods compared in this section have greatly reduced the dimensions of the dataset. For example, in the Colon dataset which has 2000 features, GCPSO, HGAFS, ACOFS and PSOFS methods have selected 10.6, 12.8, 11.9 and 96.9 features respectively. This shows that the larger the size of the data sets, the more optimal the proposed method is than the other methods. Also, in this table, the average number of selected features for all data sets is also calculated. Among the methods compared in this section, the ACOFS method, which is a method based on the ant colony algorithm, has the best rank with an average of 7.19 features. Also, the PSOFS method, which is based on the particle swarm optimization algorithm, has selected the highest number of features with an average of 29.85 and won the lowest rank. Also, the HGAFS method is a genetic algorithm based method. Therefore, since the proposed method is based on the particle swarm optimization algorithm, and on the other hand, the PSOFS method is also based on the particle swarm optimization, so the comparison is more on the PSOFS algorithm, and on the other hand, the smaller the set of features selected by each of the algorithms , that algorithm is more optimal.

Table 2: The average size of the subset selected by the proposed method compared to other methods

Dataset	Feature Selection Method				
	GCPSO	HGAFS	ACOFS	PSOFS	All features
Wine	6.2	4.7	4.9	7.3	13
Hepatitis	6.8	5.2	5.1	9.4	19
WDBC	8.2	6.2	6.4	13.7	30
Ionosphere	6.8	6.5	6.7	15.2	34
Spambase	7.8	6.9	6.8	17.1	57
Sonar	8.2	7.3	7.1	21.6	60
Colon	10.6	12.8	11.9	96.9	2000
Arcene	11.4	8.4	8.6	57.6	10000
Average	8.25	7.25	7.19	29.85	311.50

3.2.3 Classifiers

To show the generalizability of the proposed method in different classifiers, four classifiers Support Vector Machine (SVM), Decision Tree (DT), Naive Bayes (NB), and K-Nearest Neighbor (KNN) have been used in the experiments.

3.2.4 Classification accuracy

In this section, different feature selection methods are compared in terms of classification accuracy on different datasets. Table 3 shows the mean and variance of classification accuracy using SVM classifier for different feature selection methods. As can be seen in the table, in most of the data sets, the proposed method performed best. For example, in the Sonar dataset, the classification accuracy for GCPSO, HGAFS, ACOFS, and PSOFS methods is 84.36, 76.33, 79.29, and 78.72, respectively. Also, the classification accuracy for the Sonar dataset in the case where all the features are selected is equal to 76.05. In this data set, the GCPSO method performed best and the HGAFS method performed the worst. The average classification accuracy in all datasets for GCPSO, HGAFS, ACOFS and PSOFS methods is equal to 85.78, 82.58, 83.79 and 84.88%, respectively. These values show that all feature selection methods were able to improve the classification accuracy compared to the case where all features were used. The GCPSO method has the best performance with 5.56% improvement and the HGAFS method has the worst performance with only 2.36% improvement.

Table 3: The mean and variance of the classification accuracy of the proposed method compared to other feature selection methods on the SVM classifier. Acc represents the classification accuracy and Std represents the standard deviation in ten independent runs.

Dataset		Feature Selection Method				
		GCPSO	HGAFS	ACOFS	PSOFS	All features
Wine	Acc (%)	93.11	92.78	93.44	94.42	96.38
	Std	1.55	1.98	1.92	1.48	0.83
Hepatitis	Acc (%)	87.54	81.50	82.63	83.95	74.33
	Std	1.68	1.57	1.46	1.42	2.60
WDBC	Acc (%)	95.38	93.15	92.76	94.81	96.11
	Std	1.13	1.44	1.47	1.25	0.52
Ionosphere	Acc (%)	91.25	87.97	89.49	90.75	86.46
	Std	1.19	1.63	1.52	1.36	1.10
Spambase	Acc (%)	89.76	85.04	87.30	88.69	88.39
	Std	1.11	1.38	1.10	1.18	1.22
Sonar	Acc (%)	84.36	76.33	79.29	78.72	76.05
	Std	1.53	2.04	2.02	1.89	1.72
Arcene	Acc (%)	61.55	60.06	61.68	62.52	56.91
	Std	2.32	2.42	2.56	2.34 1.77	
Colon	Acc (%)	83.33	83.81	83.80	85.23	67.14
	Std	2.26	2.60	2.36	2.55	2.47
Average	Acc (%)	85.78	82.58	83.79	84.88	80.22
	Std	1.79	1.88	1.80	1.68	1.58

The results obtained using the three classifiers DT, NB and KNN are almost similar to the results obtained using the SVM classifier. For example, according to Table 4, which shows the classification results according to the DT classifier, the GCPSO method has the best performance with an average classification accuracy of 84.72% and has been able to achieve an average accuracy of 7.28%. Increase the classification compared to the case without feature selection. Also, according to Tables 5 and 6, in NB and KNN classifiers, respectively, the proposed method has the best performance among all feature selection methods with average classification accuracy of 85.06 and 86.89%, respectively.

Table 4: Mean and variance of classification accuracy in the proposed method compared to other feature selection methods on the DT classifier. Acc represents the classification accuracy and Std represents the standard deviation in ten independent runs.

Dataset		Feature Selection Method				
		GCPSO	HGAFS	ACOFS	PSOFS	All features
Wine	Acc (%)	92.45	91.30	91.96	92.78	92.45
	Std	1.48	2.00	1.84	1.44	1.06
Hepatitis	Acc (%)	84.52	80.74	81.50	84.14	72.82
	Std	1.22	1.57	1.39	1.42	2.36
WDBC	Acc (%)	94.40	91.96	91.37	94.08	94.86
	Std	1.56	1.14	1.46	1.23	1.07
Ionosphere	Acc (%)	90.83	86.71	88.56	88.14	83.61
	Std	1.15	1.16	1.56	1.42	2.07
Spambase	Acc (%)	88.48	83.01	84.90	86.36	84.94
	Std	1.82	1.67	1.90	1.89	1.72
Sonar	Acc (%)	83.37	79.99	80.83	76.75	74.08
	Std	1.62	2.32	1.60	2.15	2.21
Arcene	Acc (%)	60.38	57.39	59.93	61.97	53.30
	Std	2.25	2.39	2.45	2.45	1.14
Colon	Acc (%)	83.33	80.48	82.85	84.76	63.80
	Std	2.01	2.62	2.45	2.50	1.82
Average	Acc (%)	84.72	81.44	82.73	83.62	77.44
	Std	1.63	1.85	1.83	1.81	1.68

3.2.5 Statistical analysis of the results

In this section, using the Friedman test, the statistical analysis of the results obtained for different feature selection methods is discussed. The Friedman test is a non-parametric statistical test that can be used to evaluate the results of N different methods on K datasets. In this article, SPSS software [20] was used to perform Friedman's test.

Table 5: Average and variance of classification accuracy in the proposed method compared to other feature selection methods on the NB classifier. Acc represents the classification accuracy and Std represents the standard deviation in ten independent runs

Dataset		Feature Selection Method				
		GCPSO	HGAFS	ACOFS	PSOFS	All features
Wine	Acc (%)	92.45	94.09	91.14	92.78	95.40
	Std	1.48	1.70	1.60	1.44	1.22
Hepatitis	Acc (%)	85.65	80.18	82.51	84.17	72.82
	Std	1.14	1.35	1.37	1.49	2.52
WDBC	Acc (%)	95.64	91.60	92.95	94.81	94.91
	Std	1.26	1.02	1.62	1.25	1.29
Ionosphere	Acc (%)	90.83	86.13	88.65	88.06	85.62
	Std	1.15	0.66	1.67	1.21	1.00
Spambase	Acc (%)	88.48	82.11	84.45	85.58	89.03
	Std	1.82	1.58	1.99	2.01	0.81
Sonar	Acc (%)	83.37	80.70	79.99	80.13	76.61
	Std	1.62	2.08	1.57	1.34	1.60
Arcene	Acc (%)	62.20	61.42	63.11	62.55	56.03
	Std	2.23	2.31	2.65	2.58	1.71
Colon	Acc (%)	81.90	79.52	82.37	83.33	67.62
	Std	2.21	2.59	2.34	2.74	2.58
Average	Acc (%)	85.06	81.96	83.14	83.92	79.75
	Std	1.61	1.66	1.85	1.75	1.59

Table 7 shows the values obtained from this test for all four classifiers. As can be seen in this table, the p-value is less than 0.05 for all data sets. Therefore, it can be concluded that the results of different methods are distinguishable from each other and the proposed method is superior to other methods.

4 Discussion

By using the particle swarm optimization algorithm and with the help of feature clustering, a feature selection method based on the overlay solution called GCPSO was presented. One of the advantages of this proposed method is to select a subset with minimum redundancy and maximum relevance. In the presented method, the Louvain community detection algorithm was used for feature clustering, which is considered the most important achievement of this article. Using this algorithm for feature clustering is important in several ways. On the one hand, the number of clusters is determined automatically and there is no need to determine the number of clusters by the user before the clustering process. On the other hand, in most of the feature clustering methods presented earlier, the dispersion of the features in each cluster was not considered. Therefore, these methods will not be able to detect optimal clusters. In the community detection algorithm used in this article, both the distribution of features within each cluster and the degree of connection of features in different clusters are considered. Therefore, this algorithm will be able to find optimal clusters.

In the previous sections, the size of the selected feature subset was shown by each of the methods on different data sets. All the compared methods have greatly reduced the dimensions of the dataset. In the review and analysis of the results obtained from the experiments, it was found that the larger the size of the data sets, the better the proposed method works than the other methods. For example, in the Colon dataset that has 2000 features, the GCPSO method performed better.

In total, the examination of different feature selection methods, in terms of choosing the optimal subset and classification accuracy on all data sets, as well as the results of the tests, showed that the GCPSO method has shown good performance.

5 Conclusion

In this article, by using particle swarm optimization algorithm and feature clustering, a new method of feature selection based on overlay solution was presented. In the proposed method, first, using the particle swarm optimization algorithm, the primary features were categorized into a number of clusters. To represent the feature clustering problem, it was done that first, the feature graph was formed using a new criterion based on the covariance vector, and then based on a community detection algorithm called Louvain, the initial feature clustering was formed. Finally, using

Table 6: Average and variance of classification accuracy in the proposed method compared to other feature selection methods on the KNN classifier. Acc represents the classification accuracy and Std represents the standard deviation in ten independent runs.

Dataset		Feature Selection Method				
		GCPSO	HGAFS	ACOFS	PSOFS	All features
Wine	Acc (%)	96.39	92.29	94.09	93.76	92.78
	Std	1.30	2.02	1.60	1.55	1.17
Hepatitis	Acc (%)	87.35	81.88	82.06	84.14	72.07
	Std	1.58	1.64	1.35	1.64	2.38
WDBC	Acc (%)	95.28	92.27	93.05	94.29	95.12
	Std	1.07	1.36	1.61	1.21	1.08
Ionosphere	Acc (%)	91.75	87.13	89.32	90.14	85.62
	Std	0.97	1.49	1.57	1.35	1.00
Spambase	Acc (%)	90.29	84.29	86.57	88.18	85.94
	Std	1.11	1.39	1.27	1.20	1.71
Sonar	Acc (%)	84.78	76.61	80.41	78.72	73.37
	Std	1.57	1.84	2.05	1.89	2.14
Arcene	Acc (%)	63.63	61.68	62.00	61.74	53.95
	Std	2.57	2.44	2.74	2.40	1.13
Colon	Acc (%)	85.71	81.85	83.33	83.33	64.76
	Std	1.97	2.53	2.53	2.53	2.00
Average	Acc (%)	86.89	82.25	83.85	84.28	77.95
	Std	1.51	1.83	1.84	1.72	1.57

Table 7: Friedman test results for the proposed method

Classifier	χ^2	Degree of freedom	p-value
SVM	14.40	4	0.006122
DT	21.30	4	0.000276
NB	10.50	4	0.032797
KNN	29.91	4	0.000005

the particle swarm optimization algorithm, the structure of each particle was determined, and finally, final clusters were formed for each particle. After clustering the features, the final features must be selected from each cluster, and for this, Fisher's criterion was used. The use of this method for feature selection led to the selection of features with maximum relevance and minimum redundancy. Evaluation of the proposed method and comparison of its performance with other feature selection methods showed that the proposed method has good performance and has the best performance among different methods in most datasets.

References

[1] M. Abdel-Basset, D. El-Shahat, I. El-Henawy, V.H.C. De Albuquerque, and S. Mirjalili, *A new fusion of grey wolf optimizer algorithm with a two-phase mutation for feature selection*, Expert Syst. Appl. 139 (2020), 112824.

[2] M.H. Aghdam, N. Ghasem-Aghaee, and M.E. Basiri, *Text feature selection using ant colony optimization*, Expert Syst. Appl. **36** (2009), no. 3, 6843–6853.

[3] U. Alon, N. Barkai, D.A. Notterman, K. Gish, S. Ybarra, D. Mack, and A.J. Levine, *Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays*, Proc. Nat. Acad. Sci. USA **96** (1999), 6745–6750.

[4] F. Amini and G. Hu, *A two-layer feature selection method using Genetic Algorithm and Elastic Net*, Expert Syst. Appl. **166** (2021), 114072.

[5] A. Asuncion and D. Newman, *UCI repository of machine learning datasets*, Available from: ¡http://archive.ics.uci.edu/ml/datasets.php¿, 2007.

[6] S.R. Bandela and T.K. Kumar, *Unsupervised feature selection and NMF de-noising for robust Speech Emotion Recognition*, Appl. Acoustics **172** (2021), 107645.

[7] S. Bandyopadhyay, T. Bhadra, P. Mitra, and U. Maulik, *Integration of dense subgraph finding with feature*

clustering for unsupervised feature selection, Pattern Recog. Lett. **40** (2014), 104–112.

[8] V.D. Blondel, J.L. Guillaume, R. Lambiotte, and E. Lefebvre, *Fast unfolding of communities in large networks*, J. Statist. Mech.: Theory Experiment **10008** (2008), 1–12.

[9] J.M. Cadenas, M.C. Garrido, and R. Martínez, *Feature subset selection Filter-Wrapper based on low quality data*, Expert Syst. Appl. **40** (2013), no. 16, 6241–6252.

[10] L. Carmen, M. Reinders, and L. Wessels, *Random subspace method for multivariate feature selection*, Pattern Recog. Lett. **27** (2006), no. 10, 067–1076.

[11] G. Chandrashekar and F. Sahin, *A survey on feature selection methods*, Comput. Electric. Engin. **40** (2014), no. 1, 16–28.

[12] A.K. Farahat, A. Ghodsi, and M.S. Kamel, *Efficient greedy feature selection for unsupervised learning*, Knowledge Inf. Syst. **35** (2013), no. 2, 285–310.

[13] I. Guyon and A.E. Elisseeff, *An introduction to variable and feature selection*, J. Machine Learn. Res. **3** (2003), 1157–1182.

[14] I. Guyon, J. Weston, S. Barnhill, and V. Vapnik, *Gene selection for cancer classification using support vector machines*, Machine Learn. **46** (2002), no. 1, 389–422.

[15] E. Hancer, *A new multi-objective differential evolution approach for simultaneous clustering and feature selection*, Engin. Appl. Artific. Intell. **87** (2020), 103307.

[16] S.M. Hazrati Fard, A. Hamzeh, and S. Hashemi, *Using reinforcement learning to find an optimal set of features*, Comput. Math. Appl. **66** (2013), no. 10, 1892–1904.

[17] H. Liu and L. Yu, *Toward integrating feature selection algorithms for classification and clustering*, IEEE Trans. Knowledge Data Engin. **17** (2005), no. 4, 491–502.

[18] Y. Liu and Y.F. Zheng, *FS-SFS: A novel feature selection method for support vector machines*, Pattern Recog. **39** (2006), no. 7, 1333–1345.

[19] J. Kennedy and R. Eberhart, *Particle swarm optimization*, Proc. ICNN'95-Int. Conf. Neural Networks, IEEE, 1995, pp. 1942–1948.

[20] J. Kim, F.J. Kohout, N.H. Nie, C.H. Hull, J.G. Jenkins, K. Steinbrenner, and D.H. Bent, *Statistical Package for the Social Sciences*, McGraw Hill, New York NY, 1975.

[21] N. Maleki, Y. Zeinali, and S.T.A. Niaki, *A k-NN method for lung cancer prognosis with the use of a genetic algorithm for feature selection*, Expert Syst. Appl. **164** (2021), 113981.

[22] P. Nimbalkar and D. Kshirsagar, *Feature selection for intrusion detection system in Internet-of-Things (IoT)*, ICT Express **7** (2021), no. 2, 177–181.

[23] M. Paniri, M.B. Dowlatshahi, and H. Nezamabadi-Pour, *MLACO: A multi-label feature selection algorithm based on ant colony optimization*, Knowledge-Based Syst. **192** (2020), 105285.

[24] R. Pascual-Marqui, D. Lehmann, K. Kochi, T. Kinoshita, and N. Yamada, *A measure of association between vectors based on "similarity covariance"*, 2013-01-21, arXiv: 1301.4291 [stat.ME]. http://arxiv.org/abs/1301.4291.

[25] G. Quanquan, L. Zhenhui, and J. Han, *Generalized Fisher score for feature selection*, Proc. Int. Conf. Uncertainty Artificial Intell., 2011.

[26] L.E. Raileanu and K. Stoffel, *Theoretical comparison between the Gini index and information gain criteria*, Ann. Math. Artif. Intell. **41** (2004), 77–93.

[27] M. Rostami, K. Berahmand, and S. Forouzandeh, *A novel community detection based genetic algorithm for feature selection*, J. Big Data **8** (2021), no. 1, 1–27.

[28] M. Rostami, K. Berahmand, E. Nasiri, and S. Forouzandeh, *Review of swarm intelligence-based feature selection methods*, Engin. Appl. Artific. Intell. **100** (2021), 104210.

[29] R. Ruiz, J.C. Riquelme, J.S. Aguilar-Ruiz, and M. García-Torres, *Fast feature selection aimed at high-dimensional data via hybrid-sequential-ranked searches*, Expert Syst. Appl. **39** (2012), 11094–11102.

[30] Y. Saeys, I. Inza, and P. Larranaga, *A review of feature selection techniques in bioinformatics*, Bioinformatics **23** (2007), no. 19, 2507–2517.

[31] M. Sharif, J. Amin, M. Raza, M. Yasmin, and S.C. Satapathy, *An integrated design of particle swarm optimization (PSO) with fusion of features for detection of brain tumor*, Pattern Recog. Lett. **129** (2020), 150–157.

[32] C. Shi, Z. Gu, C. Duan, and Q. Tian, *Multi-view adaptive semi-supervised feature selection with the self-paced learning*, Signal Process. **168** (2020), 107332.

[33] Q. Song, J. Ni, and G. Wang, *A fast clustering-based feature subset selection algorithm for high-dimensional data*, IEEE Trans. Knowledge Data Engin. **25** (2013), no. 1, 1–14.

[34] X. Sun, Y. Liu, J. Li, J. Zhu, H. Chen, and X. Liu, *Feature evaluation and selection with cooperative game theory*, Pattern Recog. **45** (2012), no. 8, 2992–3002.

[35] S. Theodoridis and K. Koutroumbas, *Pattern Recognition*, Academic Press, Oxford, 2008.

[36] S. Theodoridis and C. Koutroumbas, *Pattern Recognition*, 4th Edn, Elsevier Inc, 2009.

[37] D. Wang, Z. Zhang, R. Bai, and Y. Mao, *A hybrid system with filter approach and multiple population genetic algorithm for feature selection in credit scoring*, J. Comput. Appl. Math. **329** (2018), 307–321.

[38] H. Xiaofei, C. Deng, and P. Niyogi, *Laplacian Score for Feature Selection*, Adv. Neural Inf. Process. Syst. **18** (2005), 507–514.

[39] Y. Yang, Z. Ma, A.G. Hauptmann, and N. Sebe, *Feature selection for multimedia analysis by sharing information among multiple tasks*, Multimedia IEEE Trans. **15** (2012), no. 3, 661–669.

[40] S. Yildirim, Y. Kaya, and F. Kılıç, *A modified feature selection method based on metaheuristic algorithms for speech emotion recognition*, Appl. Acoustics **173** (2021), 107721.

[41] Y. Zhang, D. Gong, X. Gao, T. Tian, and X. Sun, *Binary differential evolution with self-learning for multi-objective feature selection*, Inf. Sci. **507** (2020), 67–85.

[42] Z. Zhang, and E.R. Hancock, *Hypergraph based information-theoretic feature selection*, Pattern Recog. Lett. **33** (2012), no. 15, 1991–1999.

[43] Y. Zhou, W. Zhang, J. Kang, X. Zhang, and X. Wang, *A problem-specific non-dominated sorting genetic algorithm for supervised feature selection*, Inf. Sci. **547** (2021), 841–859.