Power System Technology
ISSN:1000-3673

Received: 06-04-2024 *Revised: 15-05-2024* *Accepted: 28-06-2024*

Particle Swarm Optimization for Feature Selection and Weighting in High-Dimensional Clustering

Sara Dehghani[1] , Razieh Malekhosseini[2]*, Karamollah Bagherifard[3] , S. Hadi Yaghoubyan[4].

[1] Ph.D. candidate, Department of Computer Engineering, Yasooj Branch, Islamic Azad University, Yasooj, Iran. Email: saradehghani18@gmail.com.

[2]* Ph.D. Assistant professor, Department of Computer Engineering, Yasooj Branch, Islamic Azad University, Yasooj, Iran. Email: malekhoseini.r@gmail.com, https://orcid.org/0000-0002-7496-9063.

[3] Ph.D. Assistant professor, Department of Computer Engineering, Yasooj Branch, Islamic Azad University, Yasooj, Iran. Email: ka.bagherifard@iau.ac.ir, https://orcid.org/0000-0001-9780-5814.

[4] Ph.D. Assistant professor, Department of Computer Engineering, Yasooj Branch, Islamic Azad University, Yasooj, Iran. Email: yaghoobian.h@gmail.com, https://orcid.org/0000-0001-6744-7884.

Abstract

Over the past decades, the rise of computers and database technologies has caused rapid growth in high-dimensional datasets. On the other hand, data is often described with numerous features, many of which may be unnecessary for a given data mining application, reducing the performance of machine learning algorithms. As such, using optimal feature selection methods is a must.

This article proposes a novel, improved version of the standard particle swarm optimization (PSO) algorithm enabled with crossover and mutation operators to enhance exploration and search capabilities. The proposed algorithm and feature clustering in the Hadoop framework are used to provide a new feature selection method. The final clusters are determined based on

Power System Technology
ISSN:1000-3673

Received: 06-04-2024 Revised: 15-05-2024 Accepted: 28-06-2024

the graph structure of the features and their relationships. This approach allows for a more comprehensive feature relevance analysis in large datasets.

The proposed method is compared to two feature selection methods, namely GCPSO_Random and GCPSO_Score, as well as some new methods that use evolutionary algorithms in their feature selection process. Given their comprehensive features, the UCI-based datasets are used to evaluate the proposed method and for comparison purposes. The results unequivocally show that the proposed method outperforms other methods on most test datasets, providing comparable or higher classification accuracy and shorter execution time.

Keywords: feature selection, big data, graph clustering, particle swarm optimization algorithm.

1. Introduction

Over the past decades, the advancement of computer and database technologies has resulted in an increasing growth of high-dimensional datasets. This growth is fueled by the increasing demand for applications with high-dimensional datasets requiring high speed and accuracy applications. Datamining is handling, processing, and analyzing this massive volume of data by linking artificial intelligence, machine learning, statistics, and databases (Alirezaei et al., 2019; Asdaghi & Soleimani, 2019). Datamining aims to extract knowledge from datasets and convert it into a comprehensible format for future applications. One challenge of datamining applications with high-dimensional datasets such as pattern recognition is the higher number of features than the number of patterns. High-dimensional datasets can reduce classifier performance in two ways: increased dimensions increases the volume of required computations, and the model built on high-dimensional data has a low generalization capability that increases the probability of overfitting (Cadenas et al., 2013; Liu & Zheng, 2006; Sun et al., 2012). Therefore, reducing problem dimension may lead to reduced computational complexity as well as improved performance of the classification algorithms. *Feature extraction* and *feature selection* are two primary introduced approaches for dimension reduction of datasets (Aghdam et al., 2009; Farahat et al., 2013; Liu & Zheng, 2006). In feature extraction, the primary feature space is mapped into a smaller space by combining existing features and creating a reduced set of features containing all or most of the information existing in the primary features. On the other hand, in feature selection a subset of the primary features are selected without creating new ones. Feature selection is considered a critical and popular technique in data preprocessing that affects the speed of machine learning algorithms and improves classifier performance. It has been recognized as an important and active research topic in pattern recognition, machine learning, and datamining since 1970s and has been extensively applied in many fields, including text classification (Aghdam et al., 2009; Jiang et al., 2010), face recognition (Vignolo et al., 2013; Zini et al., 2015), image retrieval (Da Silva et al., 2011; Rashedi et al., 2013), medical diagnosis (Inbarani et al., 2014; Jaganathan & Kuppuchamy, 2013), and finance (Huang & Tsai, 2009).

Power System Technology
ISSN:1000-3673

Received: 06-04-2024 *Revised: 15-05-2024* *Accepted: 28-06-2024*

Previous research suggest that the performance of feature selection can be improved by using optimization algorithms and selecting appropriate features in terms of both quantity and type. Some feature selection methods effectively eliminate irrelevant features, but, due to not considering the relationships between features, they fail to identify features with redundancy. On the other hand, another category of feature selection methods focuses solely on detecting and removing redundant features, neglecting the elimination of irrelevant features in the feature selection process. This issue represents a research gap. Additionally, in feature selection methods, determining the appropriate search algorithm plays a vital role. A feature-selection method must be evaluated from two perspectives: performance and effectiveness. The performance of a feature selection method depends on the time required to find the final subset of features, while effectiveness is contingent upon the quality of the selected feature subset. These two criteria are often in conflict with each other, and improving one usually leads to incompatibility with the other. Therefore, striking a balance between these two criteria has become a crucial and essential issue in feature selection, representing yet another research gap. In this paper, an effort is made to present a new model of particle swarm optimization algorithms, featuring novel operators for enhancing its search capabilities. Additionally, a feature clustering algorithm is introduced as part of a novel feature selection method. In this method, initial features are first classified into several clusters using an improved particle swarm optimization algorithm. The representation of the feature clustering problem involves determining the optimal number of clusters, and then the structure of each particle in the optimization algorithm is determined based on this optimal number of clusters. Subsequently, final clusters are formed in parallel for each particle. After clustering the features, the final features need to be selected from each cluster. For this purpose, a new criterion will be proposed that selects final features based on the level of relationships between features and their importance.

This study aims to contribute by addressing the following questions:

1. What is the degree of feature clustering impact on selection of relevant features and reduction of redundancy among selected features?
2. What is the degree of accuracy enhancement of feature selection for the optimization algorithms?
3. To what extent can Parallel processing of the map-reduce model reduce the computational complexity of optimal feature selection?

2. Literature review

In the feature selection problem, numerous search algorithms have been proposed based on various techniques to find a global optimal solution within a reasonable time. However, as the number of features increases, the execution time of these algorithms exponentially rises. This has led researchers to focus more on heuristic and metaheuristic algorithms. Methods operating based on heuristic search strike a balance between computational complexity and the quality of the obtained solution, significantly enhancing algorithm speed. While these methods yield a final solution in a reasonable time, they do not guarantee finding a global optimal solution. As

Power System Technology
ISSN:1000-3673
Received: 06-04-2024 Revised: 15-05-2024 Accepted: 28-06-2024

a result, various algorithms with diverse concepts have attempted to minimize this challenge in the pursuit of finding the best subset of primary features. These are named metaheuristic algorithms. Among the metaheuristic methods proposed for feature selection, population-based optimization algorithms, such as Genetic Algorithm (GA) (Holland, 1992), Ant Colony Optimization (ACO) (Dorigo et al., 1996), and Particle Swarm Optimization (PSO) (Kennedy & Eberhart, 1995), have gained more attention. Metaheuristic algorithms, through exploring the problem space and focusing on promising solutions, aim to find an optimal solution to the problem. Utilizing this approach, they have been successful in significantly reducing the risk of getting trapped in local optima. The Particle Swarm Optimization (PSO) algorithm was first introduced by Kennedy and Eberhart in 1995 as a global optimization method. This approach applies to problems where the solution is a point or a surface in an n-dimensional space. This algorithm operates based on probabilistic rules rather than deterministic ones. The quality of the proposed solution path is not dependent on the initial population. Starting from any point in the search space, the algorithm eventually converges the problem solution to the optimal answer.

Rashno and colleagues (Rashno et al., 2022) introduced a new multi-objective feature selection method based on particle swarm optimization. In this approach, feature vectors are decoded into particles and ranked in a two-dimensional optimization space. To tackle the complexity issue, an efficient optimization approach is required. In this regard, Thaher and colleagues (Thaher et al., 2022) proposed an efficient feature selection approach based on Boolean Particle Swarm Optimization (BPSO) enhanced with Evolutionary Population Dynamics (EPD). The proposed enhancement to BPSO aids in overcoming local optima by increasing exploratory capabilities. Song and colleagues (Song et al., 2022) introduced a combined feature selection algorithm using Sample Substitute Particle Swarm Optimization (SS-PSO). This algorithm is performed in two stages, aiming to reduce computational costs. Zhou and Hua (Zhou & Hua, 2022) proposed a novel feature selection method based on a correlation-guided genetic algorithm, aiming to enhance the efficiency of the evolutionary process. Additionally, Shreem and colleagues (Shreem et al., 2022) introduced an Advanced Binary Genetic Algorithm (EBGA) as a feature selection algorithm. Rostami and colleagues (Rostami et al., 2021) proposed a genetic algorithm based on community detection, functioning in three stages. Additionally, Manbari and colleagues (Manbari et al., 2019) introduced a hybrid method based on the ant colony algorithm, suitable for datasets with high dimensions due to its low computational complexity. However, it is worth noting that, due to its two-stage nature, there is a possibility of eliminating some features in the initial stage. Moradi and Rostami (Moradi & Rostami, 2015) introduced an algorithm for evaluating selected subsets using a new criterion. This has led to an increase in the accuracy of the method; however, due to its multi-stage nature, it comes with computational complexity and a relatively high execution time.

In feature clustering, which is one of the most effective solutions for reducing the dimensions of a dataset, the initial features are divided into several clusters, and then a certain number of features are selected from each cluster. Clustering is performed in such a way that the features within each cluster are similar to each other, while features from different clusters are distinct. Two key points must be considered when using the clustering approach in feature selection.

Power System Technology
ISSN:1000-3673
Received: 06-04-2024 Revised: 15-05-2024 Accepted: 28-06-2024

Firstly, a similarity metric needs to be introduced to measure the similarity between features. Additionally, a clustering algorithm must be specified.

Jiang and colleagues (Jiang et al., 2010) employed a method for feature clustering in which the similarity between a feature and a specific cluster is calculated using both the mean and variance metrics. Cheung and Jia (Cheung & Jia, 2012) proposed a dimensionality reduction method using feature clustering, where the number of clusters is automatically determined during the clustering process. In many past studies, graph-based methods have been utilized to address feature selection problems. Song and co-authors (Song et al., 2011) presented a graph clustering-based approach for feature selection in high-dimensional datasets.

2.1 Research Gap

When introducing a feature selection method, it needs to be evaluated from two perspectives. The first perspective is computational complexity, which refers to the time required to find the final feature subset. The second perspective is classification accuracy, which represents the quality of the selected feature subset. Previous studies have generally addressed these aspects separately, but none have investigated the simultaneous consideration of both criteria. This is because these two criteria are often conflicting, and typically, improving one may lead to a compromise on the other. This research aims to bridge this research gap and explore the trade-off between these two evaluation criteria. Therefore, in this study, a novel hierarchical algorithm based on feature relationships, incorporating clustering techniques, and presenting an improved version of the particle swarm optimization algorithm will be introduced for feature selection. In this paper, the representation of the feature clustering problem is achieved by initially forming a feature graph based on feature relationships. Then, the structure of each particle in the particle swarm optimization algorithm is determined based on the optimal number of clusters. Subsequently, considering the new operators with search capabilities, the final clusters are formed in parallel for each particle. After clustering the features, the final features need to be selected from each feature cluster. One of the main advantages of this method is the use of feature clustering based on a map-reduce model in the process of searching for optimal subsets that minimize redundancy between the selected features. Utilizing this model ensures that the similarity between the final features is minimized, resulting in high accuracy in the feature selection method.

3. research method

The proposed method consists of four stages:

1. Graph representation,
2. Feature clustering,
3. Executing the improved PSO algorithm on clusters in parallel,
4. Selecting the final features from each cluster.

Power System Technology
ISSN:1000-3673
Received: 06-04-2024 Revised: 15-05-2024 Accepted: 28-06-2024

Figure 1 illustrates the proposed method's process, each of these stages will be explained in detail below.

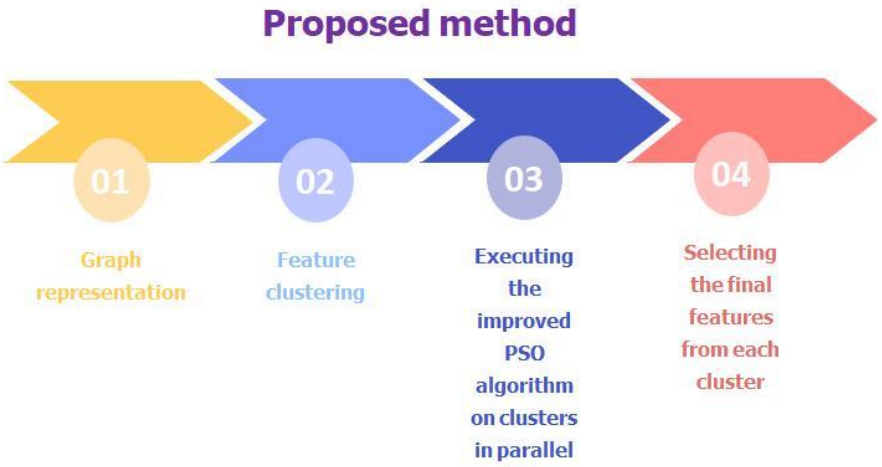


Fig.1 The proposed method's procedure

3.1 Creating features graph

To cluster features, the feature space should be represented in a graph representation. Thus, the problem is represented as a complete weighted undirected graph $G = (F, E, w_F)$, where $F = \{F_1, F_2, \ldots, F_n\}$ represents the set of n primary features that each feature is a node of the graph. $E = \{(F_i, F_j): F_i, F_j \in F\}$ represents the edges of the graph, and $w_F: (F_i, F_j) \rightarrow \mathbb{R}$ is a function indicating the similarity between two features F_i and F_j.

Determining a criterion for calculating the similarity between two features is critical to the feature clustering process. Selecting a suitable criterion significantly affects the performance of the feature selection algorithm. Various methods exist for calculating the similarity between features, each producing different results. The similarity degree can be calculated in terms of the distance between the two feature vectors. Since similarity and distance are inversely proportional, by providing a distance criterion for two feature vectors and inversing it we achieve the similarity between the features.

Various existing distance measurement criteria can be classified into two main categories: Euclidean and non-Euclidean. The cosine distance criterion is a non-Euclidean distance criterion, which is used in this article. This criterion calculates the distance between two points as the cosine of the angle between them. This can be achieved by taking the dot product between two vectors, provided that both vectors are normalized.

Power System Technology
ISSN:1000-3673
Received: 06-04-2024 Revised: 15-05-2024 Accepted: 28-06-2024

3.1.1 Cosine similarity

Cosine similarity is a measure of the closeness between two vector features, calculated as the cosine of the angle between them. The cosine of 0 is 1, and any other angle's cosine similarity is less than 1. In fact, angle 0 indicates the most similarity and angle 90 indicates the lowest similarity value, i.e., zero (perpendicular vectors).

Cosine similarity can be calculated as follows:

$$SimCos(S_i, S_j) = \frac{S_i \cdot S_j}{||S_i||\ ||S_j||} = \frac{\sum_{t=1}^{D}(S_{it} \times S_{jt})}{\sqrt{\sum_{t=1}^{D}(S_{it})^2} \times \sqrt{\sum_{t=1}^{D}(S_{jt})^2}} \quad (3\text{-}1)$$

3.2 Optimal number of clusters

For the feature clustering problem, the optimal number of clusters is determined using the improved version of the partition coefficient index. The partition coefficient index was first introduced by Bezdek in 1973 (Bezdek, 1973) and is defined as follows:

$$PC = \frac{1}{n}\sum_{i=1}^{C}\sum_{j=1}^{N}\mu_{ij}^2 \quad (3\text{-}2)$$

To calculate the optimal number of clusters using this criterion, the index is calculated for different numbers of cluster values, the highest value indicates the optimal number of clusters.

Since the partition coefficient index is uniformly dependent on the degree of fuzziness, an improved version of the index called MPC is used in this paper. The MPC does not have uniform dependence on the degree of fuzziness and is defined as follows:

$$MPC = 1 - \frac{c}{c-1}(1 - PC) \quad (3\text{-}3)$$

MPC is used to find the optimal number of clusters more robustly, taking into account the degree of fuzziness. The optimal number of clusters can be determined by calculating the MPC for different clusters.

3.3 Initial feature clustering

The data distribution within a cluster is a critical criterion in clustering, as it can significantly improve the performance considering features' dispersion. The Louvain algorithm (Blondel et

Power System Technology
ISSN:1000-3673
Received: 06-04-2024 Revised: 15-05-2024 Accepted: 28-06-2024

al., 2008) is a fast and efficient community detection algorithm, which performs graph clustering by maximizing the modularity function.

The simplicity, iterative nature of the Louvain algorithm results in a simple implementation and analysis. Moreover, it has an efficient time complexity of $O(n \log n)$, where n represents the number of nodes. Therefore, it is suitable for clustering graphs with large number of nodes, e.g., several million nodes.

Due to the mentioned advantages, Louvain is used for initial features clustering in the proposed method. Figure 2 illustrates the feature clustering using the Louvain algorithm.

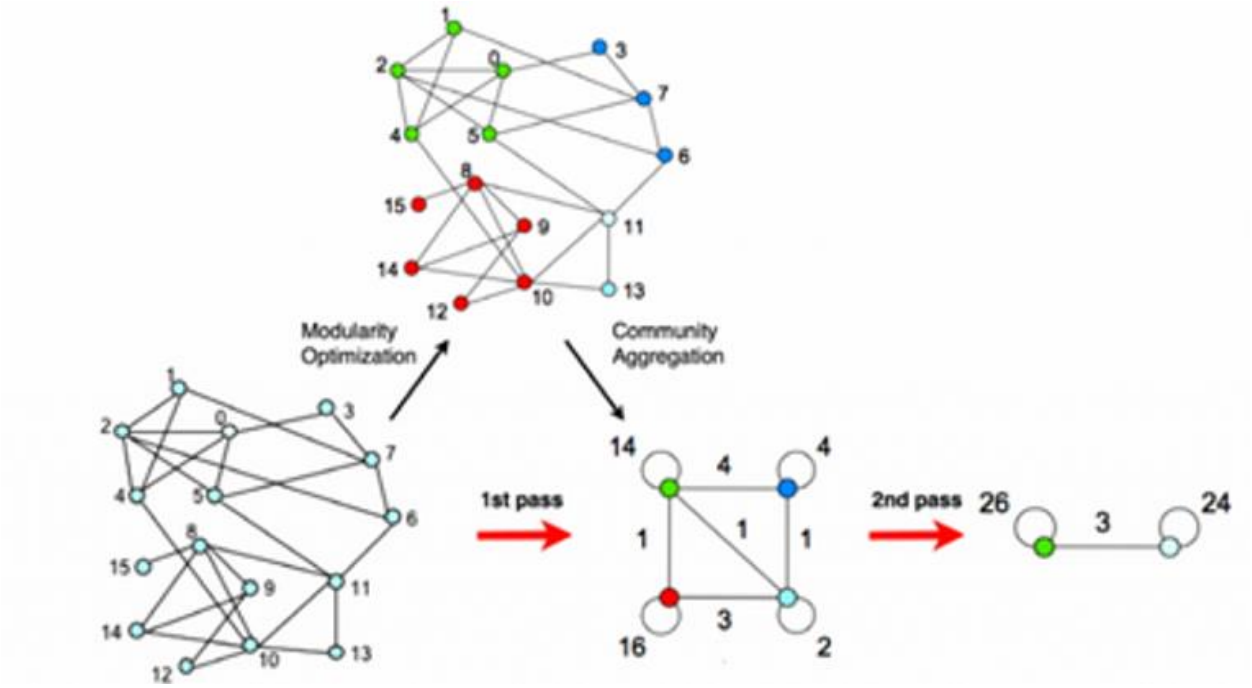


Fig.2 Feature clustering using the Louvain algorithm.

Then, the structure of each particle in the PSO algorithm is determined based on the new clusters. In this stage, an optimization function is defined to optimize the cluster centers using a combination of the parallel mapping model and the PSO. Each particle in the PSO algorithm acts as a mapping particle, indicating the resulted clusters.

3.4 Improving the PSO Algorithm and Selecting the Final Features

In 1995 (Kennedy & Eberhart, 1995), Eberhart and Kennedy introduced the PSO algorithm as a non-deterministic search method for optimizing a function.

Power System Technology
ISSN:1000-3673
Received: 06-04-2024 Revised: 15-05-2024 Accepted: 28-06-2024

The PSO algorithm initiates by randomly generating a set of particles and seeks an optimal solution by updating generations. In each step, the position and velocity of each particle are updated using two factors; the particle's best achieved position so far, known as p_{best}, and the best position discovered by the entire particles population so far, known as g_{best}. In the PSO algorithm, each particle is denoted by a vector $x_i = (x_{i1}, x_{i2}, \dots, x_{iD})$ in the search space, where D represents the dimensions of the problem. Moreover, each particle possesses the velocity of v, represented by the vector $v_i = (v_{i1}, v_{i2}, \dots, v_{iD})$. After determining the best values (pbest and gbest) in each iteration, each particle's velocity and position vectors are updated using the following equations.

$$x = x + v \quad (3\text{-}4)$$

$$v = v \times q\ C_1 \times r_1 \times (pbest - x) + C_2 \times r_2 \times (gbest - x) \quad (3\text{-}5)$$

where, q is an input weight that governs the effect of the previous velocity on the current velocity. Additionally, C_1 and C_2 random values take values between zero and one.

The proposed method utilizes the PSO algorithm to search for the optimal feature subset in seven steps, which is shown in Figure 3 and each of the steps are detailed below.

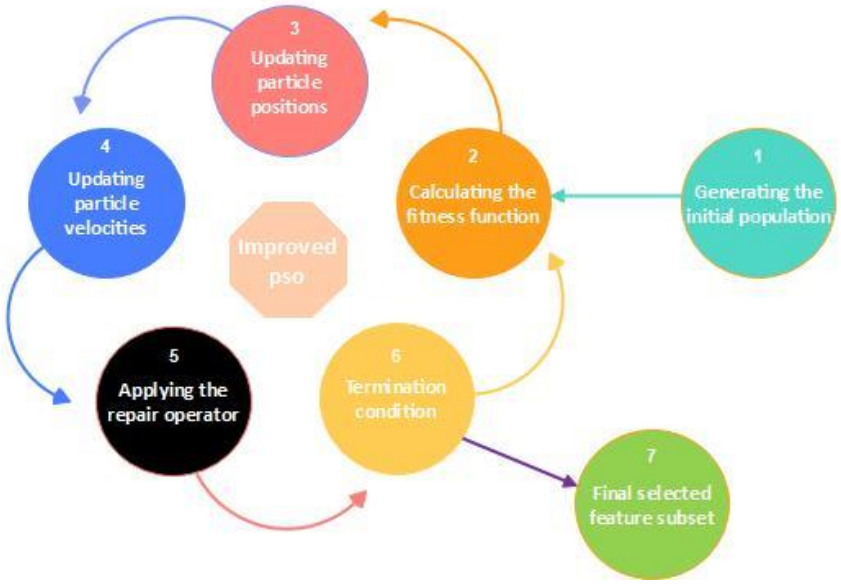


Fig.3 Improved PSO algorithm

Power System Technology
ISSN:1000-3673
Received: 06-04-2024 Revised: 15-05-2024 Accepted: 28-06-2024

Step 1: Generating the initial population

The first step in any PSO algorithm is to generate an initial population of particles. Each particle represents a solution; therefore, each particle must represent a subset of features. The length of each particle is equal to the size of the primary features, n. Each feature in the particle represents whether it has been selected or not, in other words, each feature is a binary value taking zero (not selected) or one (selected). The number of selected features for each particle, when generating the initial population, must be identical for the entire population. This value is calculated as $\omega \times k$, where k represents the number of clusters, and ω is a parameter that controls the number of selected features. In other words, in every particle of the initial population, only $\omega \times k$ features have value of one. The larger the value of ω, more features are selected. It is important to note that the initial population is created randomly, ensuring that each particle has only $\omega \times k$ features with value of one.

Step 2: Calculating the fitness function

After generating the initial population, the fitness function value must be calculated for all particles. To do this, we use a combination of classification accuracy in the KNN classification algorithm and the total similarity between the selected features. The fitness of the feature subset FS^k in iteration t, denoted by $J(FS^k(t))$, is calculated using Equation (3-6).

$$J\left(FS^{k}(t)\right)=\frac{CA\left(FS^{k}(t)\right)}{\frac{2}{\left|FS^{k}(t)\right|*\left(\left|FS^{k}(t)\right|-1\right)}\sum_{F_i,F_j\in FS^{k}} Sim(F_i,F_j)} \tag{3-6}$$

where, $CA\left(FS^k(t)\right)$ denotes the classification accuracy for the selected feature subset $FS^k(t)$ using the KNN classifier, $|FS^k(t)|$ represents the size of the selected feature subset $Fs^k(t)$ and $Sim(F_i, F_j)$ represents the similarity between feature F_i and F_j. As in Equation (3-6), the fitness calculation for each subset considers both the classification accuracy and the total similarity between selected features concurrently. Consequently, a higher fitness is assigned to a subset of features that has minimal redundancy as well as highest correlation with the target class.

Step 3: Updating particle positions

In this case, the particle positions representing our solutions are updated according to Equation (3-4). Specifically, the feature subsets generated by the PSO algorithm are updated.

Step 4: Updating particle velocities

In this step, like the previous one, the particle velocities are updated based on Equation (3-5). Every solution in the PSO algorithm comprises a velocity and a position.

Power System Technology
ISSN:1000-3673
Received: 06-04-2024 Revised: 15-05-2024 Accepted: 28-06-2024

Step 5: Applying the repair operator

The proposed method's most crucial and fundamental part lies in the repair operator, which is applied to all particles. This operator is executed to ensure that ω features are selected from each cluster. In other words, if the number of selected features from each cluster exceeds ω, ω features are retained, and the remaining features are removed. Conversely, if the number of selected features from a cluster is less than ω, features are selected from that cluster until the number of selected features equals ω. The details of the repair operator in this proposed method are fully explained in the following section.

Step 6: Termination condition

In this step, the iterations of optimum PSO algorithm is checked. If the number of iterations has reached the predetermined limit, the algorithm moves on to step seven. Otherwise, the algorithm cycle is repeated, and step two is executed again.

Step 7: Final selected feature subset

In this final step of the algorithm, the best particle, representing the best-selected feature subset among all particles generated in all iterations is chosen as the final feature subset.

3.4.1 Repair Operator:

As explained in step five above, the repair operator is intended to modify the particles generated in each iteration so that a predetermined number of features are selected from each cluster. This ensures the selected features to be chosen from the entire feature space while minimizing redundancy. Figure 4 illustrates the repair operator steps for a particle representing a problem with eight features. It is assumed the parameter ω is set to one, meaning that one feature is selected from each cluster. Figure 4-(a) displays a particle representing a subset with three features, F_1, F_4, and F_6. After modification, the repair operator must determine how many features to be selected from each cluster. As seen in Figure 4-(b), features F_1 and F_4 are selected from the first cluster and feature F_6 from the second cluster. Therefore, in the next step, one of the selected features from the first cluster (here F_1 and F_4) should be removed, and a feature from the third cluster should be selected. This stage is shown in Figure 4-(c). In the final stage of the repair operator, the changes are applied to the initial particle and a new particle is created. Figure 4-(d) shows the modified particle. As you can see, only one feature is selected from each cluster.

No explanation was given before regarding which features from each cluster should be added/removed. For example, suppose one feature must be selected from each cluster. If no features are selected from one of the clusters in a specific particle, all features in that cluster are candidates to be added to that particle. The question raised here is which feature is better to select. There are two different strategies for selecting/removing a feature from a cluster.

Power System Technology
ISSN:1000-3673
Received: 06-04-2024 Revised: 15-05-2024 Accepted: 28-06-2024

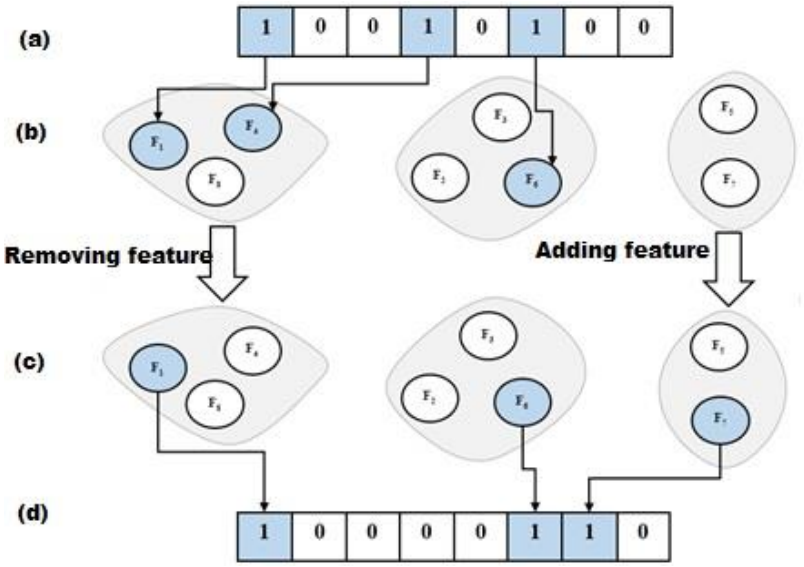


Fig.4 Example repair operator with eight features

First Strategy: Random Repair

The first strategy involves randomly selecting unselected features from a cluster until the required limit is reached when the number of features in a cluster is less than the required threshold. Similarly, if the number of selected features in a cluster exceeds the desired amount, randomly selected features are removed until the number of selected features reaches ω. The use of this strategy in the first proposed method is denoted by the addition of the "Random" suffix to its name. Therefore, the first proposed method that utilizes a random repair strategy is called GCPSO_Random.

Second Strategy: Score Repair

While the first strategy has a high-speed repair operator, features are randomly selected without considering their appropriateness, leading to a slower PSO convergence and reduced performance. To address this issue in the second strategy, the repair operator selects/removes features based on a probability calculated from a criterion that considers each feature's fitness. At the start of the PSO algorithm, the fitness of all features is calculated using a criterion, and the resulting values are normalized to the range zero to one. These values are then used to determine the probability of adding/removing each feature during the repair process. The Fisher score (Gu et al., 2012) is the criterion for evaluating the fitness of each feature in this strategy, calculated using Equation (3-7).

$$FS(S,A)=\frac{\sum_{v\in Values(S)} n_v\ (\bar{A}_v-\bar{A})^2}{\sum_{v\in Values(S)} n_v\ (\sigma_v(A))^2} \tag{3-7}$$

Power System Technology
ISSN:1000-3673
Received: 06-04-2024 Revised: 15-05-2024 Accepted: 28-06-2024

Where $\overline{A}$ is the overall mean of the pattern set corresponding to feature A, n_v is the number of patterns with a class label v, and $\sigma_v(A)$ and $\overline{A}_v$ indicates the standard deviation and mean of the within-class patterns v concerning feature A, respectively.

To normalize the fitness values of each feature to the range [0, 1], a two-step nonlinear scaling technique is utilized. For a dataset $X = \{x_1, x_2, \ldots, x_k\}$ comprising k data-points, the first step of the nonlinear scaling technique is computed using Equation (3-8), and the second step is calculated using Equation (3-9).

$$x_i' = \frac{x_i - \bar{X}}{\sigma(X)} \tag{3-8}$$

In Equation (3-8), x_i represents the value of data-point i from the set X, and $\bar{X}$ and $\sigma(X)$ denotes the mean and dispersion of the set X, respectively.

$$\tilde{x}_i = \frac{1}{1+\exp(-x_i')} \tag{3-9}$$

Where $\acute{x}_i$ is calculated from equation (3-8), and $\tilde{x}_i$ is the normalized value of data-point x_i.

When using the nonlinear scaling technique, x_i represents the fitness value of feature i, which is computed using the Fisher Score calculation function for feature F_i with respect to the pattern set S, denoted by $x_i = FS(S, F_i)$.

After computing the normalized Fisher Score values for all features, these values are used during the addition/removal process of features.

This strategy's primary objective is to incorporate each feature's fitness during the repair process, which leads to faster convergence of the PSO algorithm. Moreover, this repair strategy limits the search space to more suitable regions. This guides the PSO algorithm, increasing its capability to find the optimal solution. The second proposed method using the score-based repair strategy is named GCPSO_Score.

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

Algorithm 1: Graph Clustering-based PSO feature selection method (GCPSO)	
INPUT	*Dt*: training dataset Clustered graph with n nodes and k clusters, n number of original features *A*: Number of iteration *P*: Population size *N*: Size of reduced features ω: Number of selected feature from each cluster

Power System Technology
ISSN:1000-3673

Received: 06-04-2024 *Revised: 15-05-2024* *Accepted: 28-06-2024*

θ: Threshold for remove edges
OUTPUT *feature = { f1,..., fm}*

1: **BEGIN ALGORITHM**
2: ***Read*** : dataset
3: ***Normalize*** : dataset using softmax scaling method
4: ***Represent*** : feature by a graph
5: ***Calculate*** : edge weights using eqs. (3-1)
6: ***Normalize*** : edge weights in graph
7: ***Remove*** : edges which their associated weights are less than θ
8: ***Clustering*** : Clustered graph with n nodes and k clusters, n number of original features
9: **Step 1**. *Initialization*
10: For each particle , do
11: Initialize the particle's position with a uniformly distribution
12: Initialize Pbest to its initial position
13: Initialize Gbest to the minimal value of the swarm.
14: Initialize velocity V
15: **Step 2.** *Calculate the fitness function (Perform Mutation operator)*
16: **Step 3**. *For each particle* , do
17: Pick random numbers r1, r2
18: Update particle's position
19: Update particle's velocity
20: If Pi < Pbest , do
21: Update the best known position of particle i.
22: If Pi < Gbest, do
23: Update the swarm's best known position
24: **Step 4.** *Perform repair operator*
25: **Step 5**. *Repeat until a termination criterion is met*
26: **Step 6**. *Output that holds the best found solution*
27: **END ALGORITHM**.

Alg.1 Pseudocode of the proposed method

The MATLAB programming language was used to implement feature selection methods in this paper.

3.5 Datasets

In this article, multiple datasets with different specifications are used to evaluate the proposed method and compare its performance to the other feature selection methods. All datasets, except the Colon dataset, are selected from the University of California database (Asuncion & Newman, 2007). The details and specifications of the Colon dataset are provided by Mr. Alon

Power System Technology
ISSN:1000-3673
Received: 06-04-2024 Revised: 15-05-2024 Accepted: 28-06-2024

and his colleagues (Alon et al., 1999). These datasets are chosen for evaluation due to their extensive properties. Table 1 presents the specifications of the datasets used. In some datasets, the features have different value ranges. The features with larger value ranges may dominate the smaller ones, increasing their likelihood of being selected. All datasets are normalized before the feature selection process using the max-min normalization method to address this issue. This method adjusts the value range of all datasets to the interval [0, 1]. Additionally, some datasets contain missing values. To overcome this problem, the missing values in these features are replaced with the mean of the available data corresponding to that feature (Theodoridis & Koutroumbas, 2006).

Table 1. Datasets features

3**Dataset**	**Features**	**Classes**	**Patterns**
Hepatitis	19	2	155
WDBC	30	2	569
Spambase	57	2	4601
Sonar	60	2	208
Colon	2000	2	62

3.6 Classifiers

To demonstrate the proposed method's generalizability across different classifiers, four classifiers are used in the experiments: Support Vector Machine (SVM), Decision Tree (DT), Naive Bayes (NB), and K-Nearest Neighbors (KNN).

3.7 Parameter Settings

The proposed method has several parameters that need to be set before the feature selection process. Some of these parameters are not specific to the proposed method and need to be set in many PSO-based feature selection methods. The appropriate values for some of these parameters have been chosen through trial and error after several initial runs. Therefore, these values may not necessarily be the best values for these parameters.

The number of iterations (I) parameter indicates the number of iterations for the PSO algorithm. The proposed method can usually find the optimal subset with less than 50 iterations. The population size (p) parameter is an influential parameter on the PSO algorithm's performance and needs to be set appropriately. For better performance, the population size should be equal to the particle length. In the proposed method, the population size is equal to the particle length, which is equal to the number of features. For example, in a dataset with 60 features, the length of each particle is also 60, and the population size is set to 60. After setting the parameters related to the PSO algorithm, two other parameters remain. The θ parameter is used to remove

Power System Technology
ISSN:1000-3673
Received: 06-04-2024 Revised: 15-05-2024 Accepted: 28-06-2024

edges before clustering, and the ω parameter determines the number of selected features from each cluster. In the experiments, the θ parameter is set to 0.5, and the ω parameter is set to 0.3. The sensitivity of the proposed method to these two parameters is subsequently investigated. Table 2 presents the different parameter values for the proposed methods.

Table 2. Adjusting the features of the proposed method

Parameter	Notation	Value
Number of iteration	I	50
Population size	P	Number of features
Size of reduced features	N	100
The threshold for removing edges	θ	0.5
Number of selected features from each cluster	ω	0.3

4. Practical Results

Experiments are conducted to evaluate the proposed method's performance on different datasets using the classifiers introduced in the previous section. The dataset is randomly divided into training, validation, and test data. Since there is a large amount of data to evaluate, it is better to select more data for training. However, to avoid overfitting and underfitting, 50% of the dataset is assigned to training, 20% to validation, and the remaining 30% to test data. While other partitioning ratios could have been used, this ratio produced higher accuracy for the proposed method. Furthermore, each feature selection method is executed 10 times after determining the training, test, and validation sets for all experiments. The average of the 10 different runs is used to compare different methods.

The latest feature selection methods in big-data (Rostami et al., 2021),(Moradi & Rostami, 2015),(Gao et al., 2020) that use evolutionary algorithms in the feature selection process are compared to the proposed method.

The comparison methods in this section are evaluated based on the number of selected features, classification accuracy, and execution time. It is important to note that in this section, the proposed method with a random repair strategy and a score-based repair strategy are respectively named GCPSO_Random and GCPSO_Score.

4.1 Size of the selected feature subset

In this section, the methods are compared in terms of the number of features they select as the final subset. Table 3 presents the average size of the feature subset selected by each method on different datasets.

Power System Technology
ISSN:1000-3673

Received: 06-04-2024 *Revised: 15-05-2024* *Accepted: 28-06-2024*

As shown in Table 3, all the compared methods have significantly reduced the dataset dimensions. For instance, on the Colon dataset that has 2000 features, the GCPSO, GAFS, ACOFS, and PSOFS methods have selected on average, 10.6, 12.8, 11.9, and 96.9 features, respectively. Additionally, Table 3 provides the average number of selected features for all datasets. Among the methods compared in this section, the GCPSO method has achieved the best rank with an average of 6.78 features. The PSOFS method has selected the highest number of features with an average of 31.74, obtaining the lowest rank. It is important to note that the GCPSO_Random and GCPSO_Score methods only differ in their repair strategy, and the number of selected features in both methods is equal. Therefore, only one column for these two strategies is included in Table 3.

Table 3. Average size of the selected subset by the proposed method compared to other methods

Dataset	**Feature Selection Method**				
	GCPSO	**GAFS**	**ACOFS**	**PSOFS**	**All features**
Hepatitis	6	5.2	5.1	9.4	19
WDBC	7.9	6.2	6.4	13.7	30
Spambase	3.2	6.9	6.8	17.1	57
Sonar	6.2	7.3	7.1	21.6	60
Colon	10.6	12.8	11.9	96.9	2000
Average	**6.78**	7.68	7.46	31.74	433.2

4.2 Classification accuracy

This section compares the classification accuracy of different feature selection methods on various datasets. Table 4 presents the mean and variance of classification accuracy using the SVM classifier for each method. It can be observed from the table that the proposed methods have shown the best performance on most datasets. For example, on the Sonar dataset, the classification accuracy for the GCPSO_Random, GCPSO_Score, GAFS, ACOFS, and PSOFS methods are 86.96%, 87.36%, 76.33%, 79.29%, and 78.72%, respectively. Comparatively, when all features are selected, the classification accuracy is 76.05% in the Sonar dataset. The GCPSO_Score method demonstrates the best performance in this dataset, while the GAFS method exhibits the worst performance.

The mean classification accuracy (and variance) across all datasets for the GCPSO_Random, GCPSO_Score, GAFS, ACOFS, and PSOFS methods are 89.59% (1.17), 91.67% (1.54), 83.96% (1.80), 85.15% (1.68), and 86.28% (1.65), respectively. These values indicate that all feature selection methods have improved the classification accuracy compared to the state when using all features. The GCPSO_Score method shows the best performance with a 11.27% improvement, while the GAFS method exhibits the lowest improvement at 3.56%. Upon

Power System Technology
ISSN:1000-3673
Received: 06-04-2024 Revised: 15-05-2024 Accepted: 28-06-2024

examining the results in Table 4, it is evident that in most datasets, the GCPSO_Score method outperforms the GCPSO_Random method. This can be attributed to the selective and appropriate feature addition/removal employed by the GCPSO_Score method, as determined by its repair strategy.

Table 4. The proposed method's mean and variance of classification accuracy compared to other feature selection methods on the SVM classifier. Here, Acc represents classification accuracy, and Std denotes standard deviation in ten independent runs.

Dataset		**Feature Selection Method**					
		GCPSO_Random	**GCPSO_Score**	**GAFS**	**ACOFS**	**PSOFS**	**All features**
Hepatitis	Acc (%)	89.14	95.54	81.50	82.63	83.95	74.33
	Std	1.48	1.68	1.57	1.46	1.42	2.60
WDBC	Acc (%)	96.76	97.38	93.15	92.76	94.81	96.11
	Std	1.13	1.13	1.44	1.47	1.25	0.52
Spambase	Acc (%)	88.22	89.76	85.04	87.30	88.69	88.39
	Std	1.22	1.11	1.38	1.10	1.18	1.22
Sonar	Acc (%)	86.96	87.36	76.33	79.29	78.72	76.05
	Std	1.20	1.53	2.04	2.02	1.89	1.72
Colon	Acc (%)	86.85	88.33	83.81	83.80	85.23	67.14
	Std	1.82	2.26	2.60	2.36	2.55	2.47
Average	Acc (%)	89.59	91.67	83.96	85.15	86.28	80.40
	Std	1.17	1.54	1.80	1.68	1.65	1.70

The classification results obtained using the DT, NB, and KNN classifiers are comparable to those obtained using the SVM classifier which is shown as an average in Figure 5. As seen in Figure 5, The GCPSO_Score method has achieved the best performance in the DT classifier, with an average classification accuracy of 86.82%. The GCPSO_Random method has ranked second in all classifier, following the GCPSO_Score methods. In the NB and KNN classifiers, the proposed GCPSO_Score method has exhibited the best performance among all feature selection methods, with average classification accuracies of 90.67% and 92.46%, respectively.

Power System Technology
ISSN:1000-3673
Received: 06-04-2024 Revised: 15-05-2024 Accepted: 28-06-2024

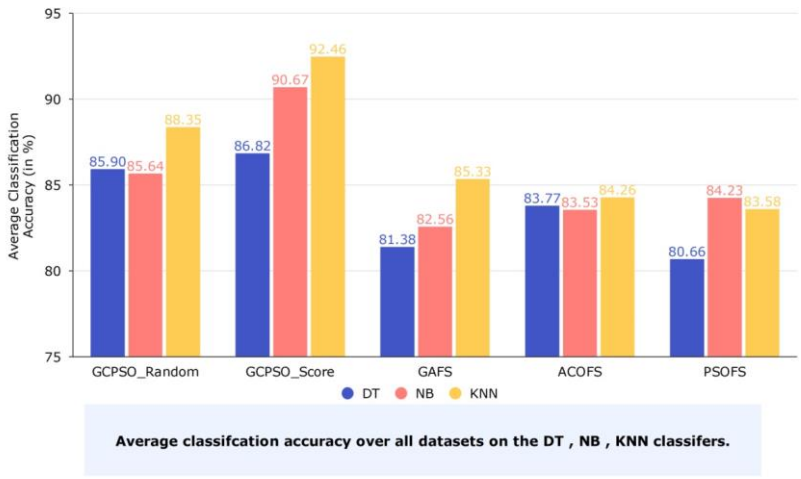


Fig.5 Average of accuracy in DT, NB, KNN classifiers on different feature selection methods

Table 5 presents the ranks of different feature selection methods for various datasets using classifiers. Here, the rank concept is the inverse of the obtained score, implying that a lower rank indicates a more optimal method. For instance, using the NB classifier, the average rank for the GCPSO_Random, GCPSO_Score, GAFS, ACOFS, and PSOFS methods are 2.4, 1.0, 4.6, 3.8, and 3.2, respectively, indicating that the GCPSO_Score method has achieved the best average rank. Additionally, the results in this table show that the GCPSO_Random and GCPSO_Score methods have exhibited better performance in the KNN classifier than in other classifiers. This can be attributed to the use of the KNN classifier in the feature selection process. Therefore, the superior performance of the proposed method in the KNN classifier is justifiable.

Table 5. Ranks obtained by different methods on all classifiers

Dataset		**Feature Selection Method**				
		GCPSO_Random	**GCPSO_Score**	**GAFS**	**ACOFS**	**PSOFS**
Hepatitis	SVM	2	1	5	4	3
	DT	3	1	5	4	2
	NB	2	1	5	4	3
	KNN	2	1	5	4	3
WDBC	SVM	3	1	4	5	2
	DT	3	1	4	5	2
	NB	4	1	5	3	2

Power System Technology
ISSN:1000-3673
Received: 06-04-2024 Revised: 15-05-2024 Accepted: 28-06-2024

	KNN	2	1	5	4	3
Spambase	SVM	3	1	5	4	2
	DT	3	1	5	4	2
	NB	3	1	5	4	2
	KNN	2	1	5	4	3
Sonar	SVM	2	1	5	3	4
	DT	2	1	4	3	5
	NB	2	1	3	5	4
	KNN	2	1	5	3	4
Colon	SVM	2	1	4	3	5
	DT	4	2	5	3	1
	NB	4	3	5	2	1
	KNN	2	1	5	3	3
Average	SVM	2.4	1.0	4.6	3.8	3.2
	DT	3.0	1.2	4.6	3.8	2.4
	NB	3.0	1.4	4.6	3.6	2.4
	KNN	2.0	1.0	5.0	3.6	3.2

4.3 Execution time

This section presents a comparison of different feature selection methods in terms of their execution time. Table 6 displays the average execution time for 10 independent runs of various feature selection methods. The results in this table indicate that, in most datasets, the GCPSO_Random and GCPSO_Score methods exhibit lower or comparable execution times compared to other feature selection methods.

Furthermore, Table 6 reports the average execution time across all datasets. Considering the mean execution time, the GCPSO_Random method demonstrates the lowest execution time, with an average of 0.22 minutes, while the PSOFS method shows the highest execution time, with an average of 3.04 minutes. Moreover, the results in this table reveal that the GCPSO_Random method exhibits a lower execution time than the GCPSO_Score method in all datasets. Given the simpler repair strategy employed by the GCPSO_Random method, this lower execution time is justifiable.

Table 6. Average execution time (in minutes) in the proposed method compared to other methods

Dataset	**Feature Selection Method**				
	GCPSO_Random	**GCPSO_Score**	**GAFS**	**ACOFS**	**PSOFS**
Hepatitis	0.019	0.022	0.056	0.043	0.062
WDBC	0.016	0.024	0.155	0.105	0.148
Sonar	0.018	0.033	0.70	0.62	0.76

Power System Technology
ISSN:1000-3673

Received: 06-04-2024 Revised: 15-05-2024 Accepted: 28-06-2024

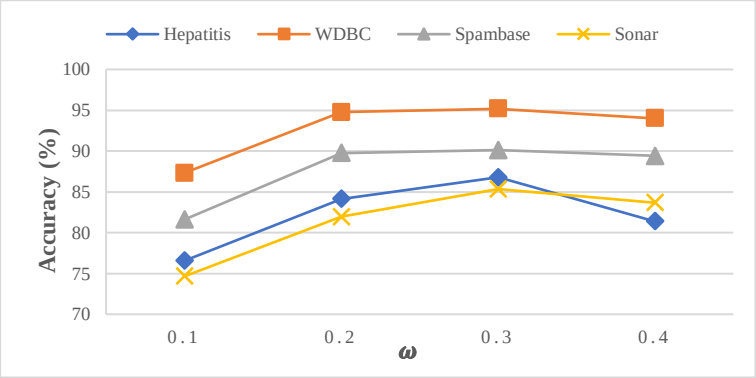
Colon	0.86	0.88	0.46	0.41	11.21
Average	0.22	0.24	0.34	0.29	3.04

4.4 Sensitivity Analysis of Parameters

This section investigates the sensitivity of the proposed GCPSO feature selection method in terms of its parameters. Accurate determination of these parameters can significantly affect the performance of the proposed method, and incorrect parameterization may lead to a decreased performance. This section examines the sensitivity of two parameters, θ and ω, used to remove edges before clustering and to select the number of features from each cluster, respectively, as described in section 3.7.

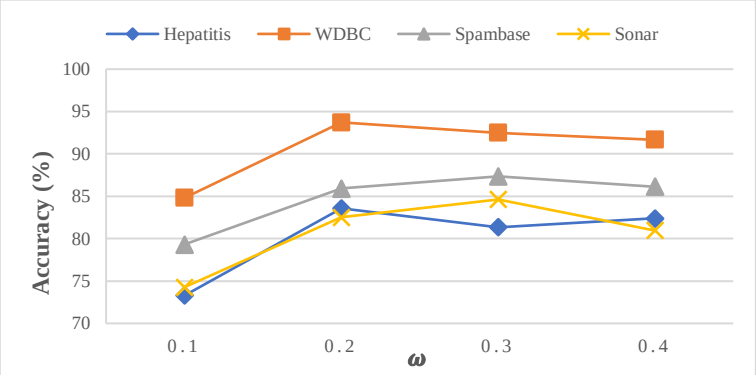
In the first set of experiments, the effect of the ω parameter is examined. Accurately selecting this parameter can have a significant impact on the performance of the proposed method. Setting ω to a large value may increase the size of the final feature subset, leading to the selection of inappropriate and redundant features. Conversely, setting ω to a small value may cause the final feature subset not to fully represent the target class information.

Figure 6 illustrates the effect of parameter ω on classification accuracy for different datasets using the SVM and DT classifiers for the GCPSO_Score method. Figure 6-(a) shows the effect of ω on classification accuracy using the SVM classifier for different datasets. As seen in the figure, the proposed method exhibits the best performance for most datasets when ω is set to 0.3. Figure 6-(b) shows the effect of this parameter on the performance of the proposed method using the DT classifier. Similar to Figure 6-(a), the proposed method achieves higher accuracy when the ω parameter is set to 0.3.



a

Power System Technology
ISSN:1000-3673
Received: 06-04-2024 Revised: 15-05-2024 Accepted: 28-06-2024



b

Fig.6 Sensitivity analysis of the GCPSO_Score method to parameter ω (a) on the SVM classifier (b) on the DT classifier

The subsequent parameter to be examined herein is parameter θ. This parameter influences the number of clusters and, consequently, the classification accuracy. This parameter takes any real values in range [0, 1]. The larger the value of this parameter, the greater the number of eliminated edges; therefore, the likelihood of obtaining additional clusters is increased. Table 7 depicts the impact of this parameter on the number of clusters derived and the classification accuracy yielded by the GCPSO_Score method as applied to the SVM classifier. As evinced by the data presented in this table, the proposed method attains greater classification accuracy when parameter θ is set to 0.4 or 0.6.

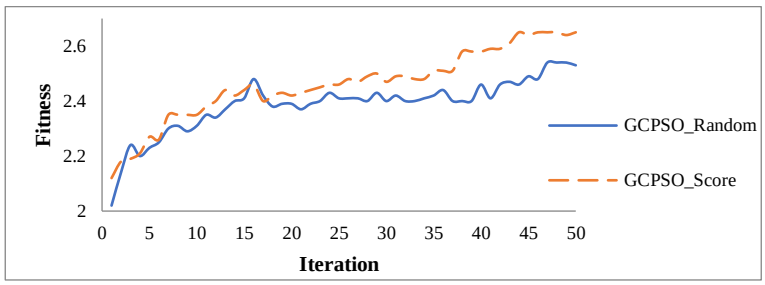
Table 7. Sensitivity analysis of the proposed method to the theta parameter. The best classification accuracy for each dataset is highlighted.

Dataset	θ	#Average Obtained Clusters	Accuracy (in %)
Hepatitis	0.2	2	83.01
	0.4	3.3	85.08
	0.6	3.5	**88.86**
	0.7	5.1	81.50
WDBC	0.2	2	92.58
	0.4	3.4	**95.85**
	0.6	5.5	94.14
	0.7	6.7	94.76
Sonar	0.2	2.4	73.09
	0.4	3.6	79.43
	0.6	4.2	**84.78**
	0.8	5.7	82.38

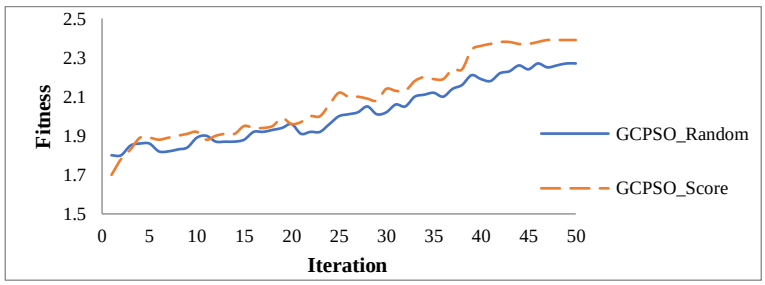
Power System Technology
ISSN:1000-3673
Received: 06-04-2024 Revised: 15-05-2024 Accepted: 28-06-2024

4.5 Algorithm convergence process

To examine the convergence of the proposed method in two separate experiments, the convergence of the GCPSO_Random and GCPSO_Score methods has been compared on the WDBC and Sonar datasets. Figure 7 illustrates the convergence trends of the GCPSO_Random and GCPSO_Score methods. Figure 7 (a) depicts the convergence trend for the WDBC dataset. As shown in this figure, in the score strategy (GCPSO_Score), the fitness function has a higher convergence value, and the proposed method converges faster. However, the convergence plot in the random strategy mode (GCPSO_Random) exhibits more fluctuations. Figure 7 (b) illustrates the convergence trend of the particle swarm optimization algorithm on the Sonar dataset. In this figure as well, in the score strategy (GCPSO_Score), the particle swarm optimization algorithm has a higher fitness function and converges more quickly. Considering the score strategy's influence on the appropriateness of feature selection, this faster convergence trend is justifiable.



A



B

Fig.7 Comparison of the convergence process of GCPSO_Random and GCPSO_Score methods (a): on WDBC dataset (b): on Sonar dataset

4.6 Statistical analysis of the results

This section discusses the statistical analysis of the results obtained for different feature selection methods using the Friedman test. The Friedman test is a non-parametric statistical test that can evaluate the results of N different methods on K datasets. This article used SPSS software (Morgan et al., 1988) to perform Friedman's test.

Power System Technology
ISSN:1000-3673

Received: 06-04-2024 *Revised: 15-05-2024* *Accepted: 28-06-2024*

Table 8- 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

Table 8 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, the results of different methods are distinguishable, and the proposed method is superior to other methods.

5. Discussion

5.1 Interpretation of results

This study aims to propose a method for feature selection based on improving the PSO optimization algorithm and feature clustering. An improved and novel version of the particle swarm optimization algorithm is presented, equipped with crossover and mutation operators. Key findings indicate that the new operators enhance the search capability of the standard optimization algorithm. Subsequently, by employing this new optimization algorithm and incorporating feature clustering, a novel feature selection method named GCPSO was introduced. It utilized two repair strategies, namely GCPSO_Random and GCPSO_Score. Our findings indicate that the use of the repair operator led to a quicker convergence of the PSO algorithm. This underscores the fact that the new optimization algorithm, in combination with the feature clustering approach, efficiently found a feature subset in the shortest possible time that was introduced as a representative subset of the core features.

The performance of the proposed method was evaluated on various datasets using four classifiers: support vector machines (SVM), naive bayes (NB), decision trees (DT), and k-nearest neighbors (KNN). Additionally, the proposed method was compared with the latest feature selection methods based on three criteria: the number of selected features, classification accuracy, and execution time.

In the examination and analysis of the results obtained from the experiments, it was evident that all the compared methods significantly reduced the dimensions of the dataset; for instance, in the Colon dataset containing 2000 features, the GCPSO, GAFS, ACOFS, and PSOFS methods selected, on average, 10.6, 12.8, 11.9, and 96.9 features, respectively. This indicates that the proposed method performs better as the size of the datasets increases.

Power System Technology
ISSN:1000-3673

Received: 06-04-2024 *Revised: 15-05-2024* *Accepted: 28-06-2024*

Furthermore, the examination of various feature selection methods in terms of classification accuracy across all datasets revealed that both methods, GCPSO_Random and GCPSO_Score, exhibited strong performance, and in most cases, they achieved a better rank compared to other methods. For example, in the Sonar dataset, the classification accuracy using the SVM classifier for the GCPSO_Random, GCPSO_Score, GAFS, ACOFS, and PSOFS methods has been 86.96%, 87.36%, 76.33%, 79.29%, and 78.72%, respectively. Furthermore, the classification accuracy for the Sonar dataset when all features are selected has been 76.05%. These values indicate that all feature selection methods have been able to improve classification accuracy compared to the scenario where all features are used. Additionally, experimental results show that, in most cases, the GCPSO_Score method has demonstrated better performance in comparison to the GCPSO_Random method. This superior performance can be justified considering the specific strategy followed by the GCPSO_Score method, which systematically adds and removes features in a targeted manner.

Additionally, comparing the proposed method with other feature selection methods in terms of execution time revealed that, in most datasets, the proposed method has a shorter or comparable execution time to other methods. Considering the average execution time, the GCPSO_Random method, with an average of 0.22 minutes, had the shortest execution time, while the PSOFS method, with an average of 3.04 minutes, had the longest execution time. Furthermore, the results indicated that the GCPSO_Random method has a shorter execution time across all datasets compared to the GCPSO_Score method. Given the simpler restoration strategy in the GCPSO_Random method, the reduced execution time for this approach is justifiable.

Furthermore, the sensitivity of two parameters, ω, which is used for "the number of selected features from each cluster," and θ, which is employed for "removing edges before clustering," was investigated. The results indicated that for most datasets when ω is set to 0.3, the proposed method exhibits the best performance. Additionally, when the parameter θ is assigned values of 0.4 or 0.6, the proposed method achieves higher classification accuracy.

5.2 Comparison with the previous studies

Our findings align with several previous studies, indicating that the combination of optimization algorithms with feature clustering methods has successfully reduced the dimensions of the dataset and identified a final set of features that effectively describe relevant data.

- In a study by Moradi and Rostami (Moradi & Rostami, 2015), a new criterion for evaluating selected subsets was introduced, employing a combination of optimization algorithms and clustering methods. This approach has led to improved accuracy, albeit with computational complexity and a relatively high execution time. The method is based on the graph structure between features and the ACO-based algorithm, integrated with Louvain clustering. They have used the Pearson correlation coefficient to calculate

Power System Technology
ISSN:1000-3673
Received: 06-04-2024 Revised: 15-05-2024 Accepted: 28-06-2024

the similarity between features. Although both cosine similarity and the Pearson coefficient have high accuracy and are vectorization methods, in our proposed method, we utilized cosine similarity for computing feature similarities. This choice was made based on the examination of results, which revealed that our proposed method is more successful when employing cosine similarity. Furthermore, due to advantages such as a simple concept, easy implementation, and rapid convergence of the Particle Swarm Optimization (PSO) algorithm compared to other optimization algorithms, we utilized the PSO algorithm in our proposed method. In contrast, Moradi and Rostami (Moradi & Rostami, 2015) employed the Ant Colony Optimization (ACO) algorithm. The ACO algorithm has a higher computational complexity compared to the PSO algorithm, leading to increased execution time. In our proposed method, we introduced a repair operator to modify the particles generated in the optimization process of the Particle Swarm Optimization (PSO) algorithm. This operator adjusts the particles in such a way that a specific number of features is chosen from each cluster. The use of this criterion enhances the performance of the PSO optimization algorithm in two aspects. On one hand, it guides the search towards optimal subsets, and on the other hand, it reduces redundancy among the selected features. In the presented repair operator of this proposed method, two different strategies were considered. In the first strategy, known as random repair, only the selected features from each cluster are taken into account. Meanwhile, in the second strategy, named score-based repair, particle adjustment is performed based on the suitability of the features. It is evident that the use of a score-based repair strategy, which systematically conducts the recovery process, can be more effective. It is worth mentioning that the use of parallel processing in the mapping and reduction model in this study, where clusters are formed in parallel for each particle, reduces computational complexity, resulting in a reduction in algorithm execution time.

- In another analysis conducted for feature selection, Rostami and his colleagues (Rostami et al., 2021) endeavored to propose a genetic algorithm in combination with a clustering method based on community detection, operating in three stages. Although they utilized a community-based repair operator, they employed a clustering algorithm other than Louvain. While one of the fastest and most efficient algorithms for community detection is the Louvain algorithm, which performs graph clustering by maximizing the modularity function, this algorithm is so simple and repeatable that its analysis and implementation are straightforward. In this algorithm, the number of clusters is automatically determined, and there is no need for prior information about the data structure before performing clustering. Furthermore, in terms of computational complexity, the Louvain algorithm has proven to be highly efficient. Therefore, this algorithm is suitable for graphs with a large number of nodes. This clustering method is entirely different from previous algorithms. Instead of using traditional clustering models such as k-means and c-means, it shapes the final clusters based on the graph structure of features and the relationships between them. For the reasons mentioned, we utilized this algorithm in our proposed method. Additionally, employing feature clustering directs the search path of the particle swarm optimization algorithm, expediting the convergence of the particle swarm optimization algorithm.

Power System Technology
ISSN:1000-3673
Received: 06-04-2024 Revised: 15-05-2024 Accepted: 28-06-2024

- In other research, Gao and colleagues (Gao et al., 2020) proposed a hybrid algorithm based on k-means clustering and PSO, utilizing Gaussian estimation for population information updates. Although their method performs better compared to PSO-based algorithms, it has shortcomings when compared to our proposed approach, which incorporates a repair operator to enhance convergence in the PSO algorithm.
- In a study by Jiang and colleagues (Jiang et al., 2010), they employed the average and variance metrics to calculate the similarity of a feature to a specific cluster. which These metrics are used in computing the Pearson correlation coefficient. However, as mentioned earlier, in our proposed method, we utilized the cosine similarity metric for feature similarity calculation due to its higher accuracy in the proposed algorithm. Additionally, they automatically determine the number of clusters in their research, a concept shared with the Louvain algorithm, which we incorporated into our proposed method.

The consequences derived from this study indicate that we have proposed a method through which we can reduce the high dimensions of large-scale data. This is one of the preprocessing stages applied to the large-scale data, contributing to the improved performance of machine learning algorithms.

6. Conclusion

In this article, efforts were made to present a new feature selection method by utilizing particle swarm optimization algorithms combined with feature clustering. In the proposed method, feature clustering and the relationships among them are utilized in the search process of the particle swarm optimization algorithm. In the presented method, the Louvain community detection algorithm was employed for feature clustering. In this algorithm, the optimal number of clusters is automatically determined. Therefore, many of the challenges posed by earlier methods for feature clustering have been addressed in this proposed approach. Issues such as the need to determine the number of clusters and dependence on initial solutions in this feature clustering method have been mitigated. Additionally, in this method, clustered features were utilized in the repair operator for the correction of generated particles. The use of feature clustering in the particle swarm optimization algorithm offers two significant advantages. On one hand, the search space of the particle swarm optimization algorithm is reduced, guiding the search path. On the other hand, the use of the repair operator ensures a specific number of features are selected from each cluster. Consequently, the final subset can serve as a representative set for all initial features. Employing this method for feature selection leads to the selection of features with maximum correlation and minimal redundancy.

The repair operator in our proposed method imposes the constraint that a specific number of features must be selected from each cluster. In some datasets, the suitable number of features in different clusters may vary. Therefore, this constraint could potentially reduce the effectiveness of our proposed method. To address this issue, the repair operator can be modified to allow the selection of different features from various clusters, facilitating future

Power System Technology
ISSN:1000-3673

Received: 06-04-2024 Revised: 15-05-2024 Accepted: 28-06-2024

investigations. Additionally, our proposed method introduces a framework for a graph representation of the problem. Therefore, insights from research in other scientific domains, such as social networks and graph theory, can be leveraged to accurately identify relationships between features.

The evaluation of the proposed method and its performance comparison with other feature selection methods revealed that the proposed approach exhibits favorable performance, outperforming other methods in most datasets. Additionally, comparing two different repair strategies, random repair, and score-based repair, demonstrated that the proposed method performs better in the score-based repair mode across the majority of datasets.

References

1. Aghdam, M. H., Ghasem-Aghaee, N., & Basiri, M. E. (2009). Text feature selection using ant colony optimization. *Expert systems with applications*, *36*(3), 6843-6853.
2. Alirezaei, M., Niaki, S. T. A., & Niaki, S. A. A. (2019). A bi-objective hybrid optimization algorithm to reduce noise and data dimension in diabetes diagnosis using support vector machines. *Expert Systems with Applications*, *127*, 47-57.
3. Alon, U., Barkai, N., Notterman, D. A., Gish, K., Ybarra, S., Mack, D., & Levine, A. J. (1999). Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays. *Proceedings of the National Academy of Sciences*, *96*(12), 6745-6750.
4. Asdaghi, F., & Soleimani, A. (2019). An effective feature selection method for web spam detection. *Knowledge-Based Systems*, *166*, 198-206.
5. Asuncion, A., & Newman, D. (2007). UCI repository of machine learning datasets. *Availablefrom:< http://archive. ics. uci. edu/ml/datasets. html*.
6. Bezdek, J. C. (1973). Cluster validity with fuzzy sets.
7. Blondel, V. D., Guillaume, J.-L., Lambiotte, R., & Lefebvre, E. (2008). Fast unfolding of communities in large networks. *Journal of statistical mechanics: theory and experiment*, *2008*(10), P10008.
8. Cadenas, J. M., Garrido, M. C., & MartíNez, R. (2013). Feature subset selection filter–wrapper based on low quality data. *Expert systems with applications*, *40*(16), 6241-6252.
9. Cheung, Y.-m., & Jia, H. (2012). Unsupervised feature selection with feature clustering. 2012 IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology,
10. Da Silva, S. F., Ribeiro, M. X., Neto, J. d. E. B., Traina-Jr, C., & Traina, A. J. (2011). Improving the ranking quality of medical image retrieval using a genetic feature selection method. *Decision support systems*, *51*(4), 810-820.
11. Dorigo, M., Maniezzo, V., & Colorni, A. (1996). Ant system: optimization by a colony of cooperating agents. *IEEE transactions on systems, man, and cybernetics, part b (cybernetics)*, *26*(1), 29-41.
12. Farahat, A. K., Ghodsi, A., & Kamel, M. S. (2013). Efficient greedy feature selection for unsupervised learning. *Knowledge and information systems*, *35*, 285-310.

Power System Technology
ISSN:1000-3673

Received: 06-04-2024 *Revised: 15-05-2024* *Accepted: 28-06-2024*

13. Gao, H., Li, Y., Kabalyants, P., Xu, H., & Martinez-Bejar, R. (2020). A novel hybrid PSO-K-means clustering algorithm using Gaussian estimation of distribution method and Lévy flight. *IEEE Access*, *8*, 122848-122863.
14. Gu, Q., Li, Z., & Han, J. (2012). Generalized fisher score for feature selection. *arXiv preprint arXiv:1202.3725*.
15. Holland, J. H. (1992). *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence*. MIT press.
16. Huang, C.-L., & Tsai, C.-Y. (2009). A hybrid SOFM-SVR with a filter-based feature selection for stock market forecasting. *Expert Systems with applications*, *36*(2), 1529-1539.
17. Inbarani, H. H., Azar, A. T., & Jothi, G. (2014). Supervised hybrid feature selection based on PSO and rough sets for medical diagnosis. *Computer methods and programs in biomedicine*, *113*(1), 175-185.
18. Jaganathan, P., & Kuppuchamy, R. (2013). A threshold fuzzy entropy based feature selection for medical database classification. *Computers in Biology and Medicine*, *43*(12), 2222-2229.
19. Jiang, J.-Y., Liou, R.-J., & Lee, S.-J. (2010). A fuzzy self-constructing feature clustering algorithm for text classification. *IEEE transactions on knowledge and data engineering*, *23*(3), 335-349.
20. Kennedy, J., & Eberhart, R. (1995). Particle swarm optimization. Proceedings of ICNN'95-international conference on neural networks,
21. Liu, Y., & Zheng, Y. F. (2006). FS_SFS: A novel feature selection method for support vector machines. *Pattern recognition*, *39*(7), 1333-1345.
22. Manbari, Z., AkhlaghianTab, F., & Salavati, C. (2019). Hybrid fast unsupervised feature selection for high-dimensional data. *Expert Systems with Applications*, *124*, 97-118.
23. Moradi, P., & Rostami, M. (2015). Integration of graph clustering with ant colony optimization for feature selection. *Knowledge-Based Systems*, *84*, 144-161.
24. Morgan, G. A., Leech, N. L., Barrett, K. C., Brace, N., Snelgar, R., Griego, O. V., Gloeckner, G. W., NorusÏÕis, M., Field, A. P., & Klecka, W. R. (1988). 1. SPSS: statistical package for the social sciences by Norman H Nie.
25. Rashedi, E., Nezamabadi-Pour, H., & Saryazdi, S. (2013). A simultaneous feature adaptation and feature selection method for content-based image retrieval systems. *Knowledge-Based Systems*, *39*, 85-94.
26. Rashno, A., Shafipour, M., & Fadaei, S. (2022). Particle ranking: an efficient method for multi-objective particle swarm optimization feature selection. *Knowledge-based systems*, *245*, 108640.
27. Rostami, M., Berahmand, K., & Forouzandeh, S. (2021). A novel community detection based genetic algorithm for feature selection. *Journal of Big Data*, *8*(1), 1-27.
28. Shreem, S. S., Turabieh, H., Al Azwari, S., & Baothman, F. (2022). Enhanced binary genetic algorithm as a feature selection to predict student performance. *Soft Computing*, *26*(4), 1811-1823.
29. Song, Q., Ni, J., & Wang, G. (2011). A fast clustering-based feature subset selection algorithm for high-dimensional data. *IEEE transactions on knowledge and data engineering*, *25*(1), 1-14.

Power System Technology
ISSN:1000-3673

Received: 06-04-2024 *Revised: 15-05-2024* *Accepted: 28-06-2024*

30. Song, X., Zhang, Y., Gong, D., Liu, H., & Zhang, W. (2022). Surrogate sample-assisted particle swarm optimization for feature selection on high-dimensional data. *IEEE transactions on evolutionary computation*.
31. Sun, X., Liu, Y., Li, J., Zhu, J., Liu, X., & Chen, H. (2012). Using cooperative game theory to optimize the feature selection problem. *Neurocomputing*, *97*, 86-93.
32. Thaher, T., Chantar, H., Too, J., Mafarja, M., Turabieh, H., & Houssein, E. H. (2022). Boolean Particle Swarm Optimization with various Evolutionary Population Dynamics approaches for feature selection problems. *Expert Systems with Applications*, *195*, 116550.
33. Theodoridis, S., & Koutroumbas, K. (2006). Pattern Recognition, vol. 855. In: Elsevier, Amsterdam.
34. Vignolo, L. D., Milone, D. H., & Scharcanski, J. (2013). Feature selection for face recognition based on multi-objective evolutionary wrappers. *Expert Systems with Applications*, *40*(13), 5077-5084.
35. Zhou, J., & Hua, Z. (2022). A correlation guided genetic algorithm and its application to feature selection. *Applied Soft Computing*, *123*, 108964.
36. Zini, L., Noceti, N., Fusco, G., & Odone, F. (2015). Structured multi-class feature selection with an application to face recognition. *Pattern Recognition Letters*, *55*, 35-41.