Knowledge-Based Systems 286 (2024) 111380

Contents lists available at ScienceDirect

Knowledge-Based Systems

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

Information gain ratio-based subfeature grouping empowers particle swarm optimization for feature selection

Jinrui Gao [a], Ziqian Wang [a], Ting Jin [b], Jiujun Cheng [c,*], Zhenyu Lei [a], Shangce Gao [a,*]

[a] Faculty of Engineering, University of Toyama, Toyama-shi, 930-8555, Japan
[b] School of Science, Nanjing Forestry University, Nanjing, 210037, China
[c] Ministry of Education, Key Laboratory of Embedded System and Service Computing, Tongji University, Shanghai, China

ARTICLE INFO

Keywords:
Particle swarm optimization
Information gain ratio
Feature selection
Classification

ABSTRACT

Feature selection is a critical preprocessing step in machine learning with significant real-world applications. Despite the widespread use of particle swarm optimization (PSO) for feature selection, owing to its robust global search capabilities, developing an effective PSO method for this task is still a substantial challenge. This study introduces a novel PSO variant, ISPSO, which integrates the information gain ratio for assessing feature importance. ISPSO's feature selection process involves partitioning features into distinct groups to establish the initial population. Recognizing that feature selection tasks are inherently binary, ISPSO replaces the traditional PSO velocity concept with a probabilistic approach. In addition, introducing a penalty term enhances the algorithm's ability to achieve superior results. Experimental evaluations on 16 datasets consistently show that ISPSO surpasses compared algorithms, highlighting its efficiency in eliminating redundant and irrelevant features.

1. Introduction

Using high-dimensional datasets has become increasingly common in real-life applications as machine learning finds widespread use across various domains, including fields such as medicine, geology, video, and audio analysis [1]. However, it is essential to recognize that real-world data often encompass need ~~numerous~~ irrelevant and redundant features, negatively affecting the accuracy of systems [2]. In response to this challenge, feature selection has emerged as a preferred method to address this issue [3].

Feature selection [4,5] is a technique used to reduce dimensionality. It differs from another dimensionality reduction approach, feature extraction, employed in many areas [6]. Feature extraction [7] reduces dimensionality by creating smaller features derived from the original feature space. However, this method can potentially result in losing original relationships between features [2]. In contrast, feature selection involves directly selecting features from the initial feature space to create a new feature space. Through feature selection, the original semantic meanings of the features are preserved, thereby enhancing the model's interpretability and reliability [8].

In recent years, numerous effective and novel feature selection methods have been proposed. Xue et al. [9] introduced a method named the External Attention-Based Feature Ranker, which employs an attention module to score each feature, thereby ranking them. Ahadzadeh et al. [10] proposed a novel feature selection algorithm for high-dimensional datasets named SFE, which is divided into two stages: exploration and exploitation, utilizing non-selection and selection operations. Ref. [11] introduced a novel feature selection technique combining standard deviation with the difference between mean and median to determine statistical significance, thereby improving network performance. Abualigah et al. [12] proposed a feature selection method merging chaotic maps with the binary group search optimizer to tackle the feature selection problem, incorporating five types of chaotic maps.

Based on established criteria, feature selection can be categorized into three primary types: filter, wrapper, and embedded methods [13]. Filter methods assess features independently of any learning algorithms, relying solely on the inherent interconnections among the features. In contrast, wrapper methods use a predefined learning algorithm to evaluate feature subsets, selecting features based on their performance with the chosen learning algorithm. Embedded methods integrate feature selection directly into the model construction process. While filter methods are computationally less intensive than both embedded and wrapper methods due to their simplicity, they may not achieve optimal results because they lack an associated learning algorithm [14]. Embedded methods generally outperform filter methods but have limitations, including difficulties in parameter adjustment and a tendency to overfit [14]. Wrapper methods, by incorporating a

* Corresponding authors.
E-mail addresses: chengjj@tongji.edu.cn (J. Cheng), gaosc@eng.u-toyama.ac.jp (S. Gao).

PSO
feature selection
feature extraction.

https://doi.org/10.1016/j.knosys.2024.111380
Received 23 November 2023; Received in revised form 29 December 2023; Accepted 7 January 2024
Available online 8 January 2024

learning algorithm, have the potential to yield superior results [15]. These methods involve two key steps: (1) searching for an optimal feature subset and (2) evaluating this subset [2]. Thus, the efficiency of the search strategy is crucial for the wrapper methods' effectiveness.

In recent years, hybrid methods [16–18], combining filter and wrapper methods, have gained increased attention as the fourth method. Filter methods assist wrapper methods by reducing the search space and accelerating the convergence of hybrid methods. Hybrid techniques leverage the advantages of both filter and wrapper methods, leading to enhanced performance [19]. Evolutionary algorithms (EAs) [20,21], recognized for their global search capabilities, are frequently used as wrapper methods in feature selection. Researchers have developed hybrid feature selection algorithms combining EAs with filter methods [22]. A recent study [23] introduced a novel filter evaluation technique termed "cost-sensitive", seamlessly integrated into the differential evolution algorithm. Wang et al. [24] integrated symmetric uncertainty into ant colony optimization with a novel graphical representation based on this measure. Karimi et al. [25] proposed a feature selection method named SemiACO, based on ant colony optimization, selecting features by considering the minimum redundancy among features and the maximum relevance between the features and class labels. Ref. [26] proposed a method combining a filter ensemble of relief and fuzzy entropy, considering the top-n ranked features from each method. This approach enhances performance by integrating the equilibrium optimizer technique with an opposition-based learning strategy, the Cauchy Mutation operator, and a novel search strategy. Another study [27] presented an innovative PSO approach with explicit particle representation (ESAPSO), where features are ranked using the maximum information coefficient. These studies collectively demonstrate the effectiveness of integrating information methods with optimization algorithms for feature selection.

Particle swarm optimization (PSO), as a type of EAs, has found practical utility in feature selection due to its commendable global search capabilities [28–30]. Tran et al. [31] introduced a PSO approach featuring variable-length representation, where features are rearranged based on their interrelationships. In another study [32], the feature space is segmented according to feature importance, and the subswarm size is adaptively adjusted. Although these PSO algorithms have exhibited strong performance, they share the common trait of utilizing randomly generated initial populations. Notably, a study [33] has demonstrated that the choice of initialization strategy significantly influences PSO algorithm performance. Therefore, to enhance PSO's performance in feature selection, we introduce a subfeature grouping strategy based on the information gain ratio for initializing the PSO population. This approach includes multiple strategies to ensure a high-quality initial population, and the search factor has been redesigned for improved efficiency.

Compared with previous algorithms, this paper makes the following main contributions:

(1) Introducing a novel initialization strategy involving subfeature grouping. Before initialization, features are categorized into three groups based on information gain ratio, with each group assigned different initialization strategies.

(2) Departing from the traditional concept of velocity in PSO, this novel algorithm replaces it with a probabilistic approach. In addition, the algorithm modifies the original acceleration coefficients, replacing them with the information gain ratio associated with each feature. Notably, the offspring is no longer directly employed as the succeeding generation's population. Instead, optimal individuals are chosen from both the parental and offspring populations to construct a new population for the subsequent generation.

(3) Implementing a penalty term for each feature, wherein if the offspring's performance exceeds that of the parent, modifications are applied to the penalty term. If a feature's penalty term drops below zero, the respective feature is marked as unselected (assigned a value of zero). To mitigate the risk of a feature remaining excluded for an extended duration, the penalty terms are restored during each iteration.

The remainder of this paper is structured as follows: Section 2 introduces PSO and information gain ratio. Section 3 presents a detailed insight into the proposed algorithm. The experimental design and results are outlined in Section 4. Finally, Section 5 offers the conclusion of this paper.

2. Background

2.1. Particle swarm optimization

PSO [34] is a population-based algorithm inspired by the collective behavior of natural groups, such as bird flocks or fish schools. Within the solution space, each potential solution is represented as a particle, each possessing a position and velocity. These particles traverse the solution space, updating their velocities and positions based on the best individual solutions and the global best solution, as outlined below:

$$V_{i,j}^{t+1} = W \cdot V_{i,j}^{t} + \overbrace{C_1 \cdot r \cdot (P_{best} - l_{i,j}^{t})}^{\text{cognitive}} + \underbrace{C_2 \cdot r \cdot (G_{best} - l_{i,j}^{t})}_{\text{social part}}$$
$$l_{i,j}^{t+1} = l_{i,j}^{t} + V_{i,j}^{t+1} \tag{1}$$

where $V_{i,j}^{t+1}$ represents the velocity of the ith individual along the jth dimension at the subsequent iteration, and $V_{i,j}^{t}$ denotes the velocity at the current iteration. Likewise, $l_{i,j}^{t}$ and $l_{i,j}^{t+1}$ correspond to the location of the individual at the current iteration and the next iteration. P_{best} and G_{best} denote the personal best individual and the global best individual, respectively. The parameters C_1 and C_2 denote the two acceleration coefficients, while r denotes a random number sampled from the interval [0,1].

The algorithm described above is primarily designed for continuous problems. However, it is important to note that feature selection is a binary problem, where features can exist in only two states: selected or unselected. Therefore, binary PSO [35] is more suitable for feature selection tasks. In binary PSO, the location update equation differs from Eq. (1) and is expressed as follows:

Imp

domain $= \mathbb{R}$, range $= (0,1)$ $\quad S(x) = \frac{1}{1+e^{-x}}$ sigmoid fun^n

$$l_{i,j}^{t+1} = sigmoid(V_{i,j}^{t+1}) > rand \tag{2}$$

Here, the velocity is transformed into a range between 0 and 1 by applying the sigmoid function for the subsequent generation. Subsequently, this velocity compared with randomly generated values, and based on this comparison, the velocity is set to either 0 or 1.

2.2. Information metrics

In the context of feature selection, each feature typically encapsulates a significant amount of information. To effectively quantify this information, various methods from information theory are utilized.

??

2.2.1. Information entropy

Information entropy [36] is a concept in information theory used to measure the uncertainty or disorder of information. The fundamental idea of information entropy is that more uncertain information has higher entropy, while more certain information has lower entropy. The calculation formula is as follows:

$$H(D) = -\sum p_i \cdot log(p_i) \tag{3}$$

where $H(D)$ denotes the information entropy of set D, while p_i denotes the proportion of each category within the set. However, it is critical to understand that information entropy primarily measures data uncertainty and is not inherently suitable for assessing feature significance in feature selection contexts [37].

entropy - uncertainity.

2.2.2. Information gain

Information gain, a widely used criterion for feature selection [38], measures the decrease in overall information entropy of a dataset when a particular feature is used for data partitioning. A high information gain for a feature implies a significant reduction in uncertainty during data division, highlighting its importance. The formula to calculate information gain is as follows:

$$\begin{aligned} G_I(D,X) &= H(D) - H(D|X) \\ H(D|X) &= \sum p(X = x_i) \cdot H(D|X = x_i) \end{aligned} \tag{4}$$

where $G_I(D,X)$ denotes the information gain and $H(D|X)$ denotes the conditional entropy given the presence of the feature X. Moreover, $p(X = x_i)$ represents the probability of the feature X being equal to x_i, and $H(D|X = x_i)$ represents the entropy when X is equal to x_i. It should be noted that while information gain is effective in ranking features, it exhibits a bias towards features with a greater number of unique values [39].

2.2.3. Information gain ratio

The information gain ratio [39], proposed as an extension of information gain, incorporates the relationship between IG and intrinsic information. Its calculation is as follows:

$$\begin{aligned} R(D,X) &= \frac{G_I(D,X)}{I(X)} \\ I(X) &= -\sum p(x_i) \cdot log(p(x_i)) \end{aligned} \tag{5}$$

where $I(X)$ denotes the intrinsic information of feature X, and $p(x_i)$ represents the probability of each feature value. The information gain ratio, yielding values from 0 to 1, is akin to information gain. A value nearing 1 indicates a significant decrease in uncertainty within the classification task framework. This metric accounts for the intrinsic entropy of the feature, thus reducing bias towards features with a high number of distinct values. Consequently, the information gain ratio is a more balanced and comprehensive metric for feature selection, especially effective in situations with diverse feature values [40].

3. Methodology

By integrating the information gain ratio with PSO, we developed a hybrid algorithm named ISPSO. Fig. 1 illustrates the detailed process of ISPSO. In ISPSO's initial phase, features are categorized into three groups according to their information gain ratio values. During the initialization, each population member randomly selects a subfeature group, and the features in this group undergo individualized initialization. Given the binary nature of the feature selection search space, the traditional concept of velocity is inapplicable. In ISPSO, "velocity" is replaced with "probability", leading to a redefinition of the solution update formula. In addition, each feature is linked with a penalty term, influencing the decision to select or deselect a feature.

why

3.1. Subfeature grouping and initialization

In ISPSO, feature evaluation employs the information gain ratio. The information gain ratio values act as indicators of feature importance, with higher values denoting greater significance. This process involves calculating the information gain ratio for each feature, facilitating their ranking by importance, and leading to the creation of a feature list. Following this ranking, the top 30% of features are assigned to one subfeature group based on their information gain ratio values. The lowest 30% are allocated to another subfeature group, while the remaining features are grouped into a third subfeature group. Consequently, the features are categorized into three groups, denoted as F_1, F_2 and F_3. During the initialization phase, F_1, F_2 and F_3 are randomly selected. For each individual, the initialization equation is expressed as follows:

$$S_{i,j} = \begin{cases} rand < R(D,j) & j \in F_k \\ rand < 0.5 & \text{otherwise} \end{cases} \tag{6}$$

where $S_{i,j}$ represents the jth feature of the ith individual. $R(D,j)$ denotes the information gain ratio of the feature, mapped to a value between 0 and 1, and F_k represents the preselected subfeature. During initialization, F_k is reselected for each individual. Assuming a population size of n, F_k is randomly selected n times. This equation indicates that, for a feature belonging to F_k, its information gain ratio is compared with a random number between 0 and 1. For features not in subfeatures, a default threshold of 0.5 is applied to decide their selection status. As depicted in Fig. 2, 10 features were divided into three subfeatures based on their information gain ratios, with F_1 and F_3 each representing 30% of the total features, and the remaining features classified under F_2. In a specific individual's scenario, the F_3 subfeature is selected randomly. For features in the F_3 subfeature, their thresholds are substituted with their corresponding information gain ratio values, while thresholds for all other features are set at 0.5. This method aids in selecting or deselecting features by comparing them with random values. Notably, features within F_3 demonstrate comparatively lower information gain ratio values, implying their reduced significance. Therefore, substituting their thresholds with these information gain ratio values decreases their selection probability in the initialization phase.

3.2. Search operator

In binary tasks, the traditional method of updating positions using velocity is not applicable. Consequently, ISPSO diverges from BPSO by substituting velocity with probability while preserving the concepts of P_{best} and G_{best}. The corresponding formula is as follows:

$$p_{i,j}^{t+1} = w \cdot p_{i,j}^{t} + r_1 \cdot (1 - R_j) \cdot (P_{best} - s_{i,j}^{t}) + r_2 \cdot R_j \cdot (G_{best} - s_{i,j}^{t}) - r_3 \cdot (D_r - s_{i,j}^{t}) \tag{7}$$

$$s_{i,j}^{t+1} = \begin{cases} 1 - s_{i,j}^{t} & p_{i,j}^{t+1} \geq rand \\ s_{i,j}^{t} & \text{otherwise} \end{cases} \tag{8}$$

where w denotes the weight coefficient, representing the degree to which the offspring is influenced by the parent. $p_{i,j}^{t+1}$ and $p_{i,j}^{t}$ denote the probabilities of the ith individual on the jth dimension at the next iteration and the current iteration. $s_{i,j}^{t}$ denotes the jth feature of the current individual, R_j denotes the information gain ratio of the jth feature after mapping to the range [0,1], used as the coefficient for G_{best}. The difference between 1 and R_j is the P_{best} coefficient. This configuration causes G_{best} to guide features with higher information gain ratios, while P_{best} tends to favor those with lower information gain ratios. This approach effectively enhances the search for a new global optimal solution. r_1, r_2, r_3 are random numbers within the range [0,1]. D_r represents a set of random numbers matching the dimensions of an individual. The additional term in the equation bolsters the algorithm's search capability and mitigates the risk of becoming trapped in local optima. According to Eq. (8), the individual for the next iteration is determined by comparing its probability with a random number between 0 and 1. To enhance the algorithm's convergence, ISPSO assembles the next generation of the population by selecting the best individuals from both offspring and parents.

3.3. Penalty term

To augment the algorithm's search capabilities, ISPSO incorporates penalty terms for each feature. At the beginning of each iteration, penalty term values for each feature are initialized to 10. As iterations progress, if a new individual's fitness exceeds that of the corresponding original individual, the penalty term is updated using the following equation:

$$PT_j = PT_j - 5 \cdot R_j \cdot (s_j - N_j) \tag{9}$$

where PT_j represents the penalty term for feature j. s_j denotes the jth feature of the individual under consideration, and N_j denotes the

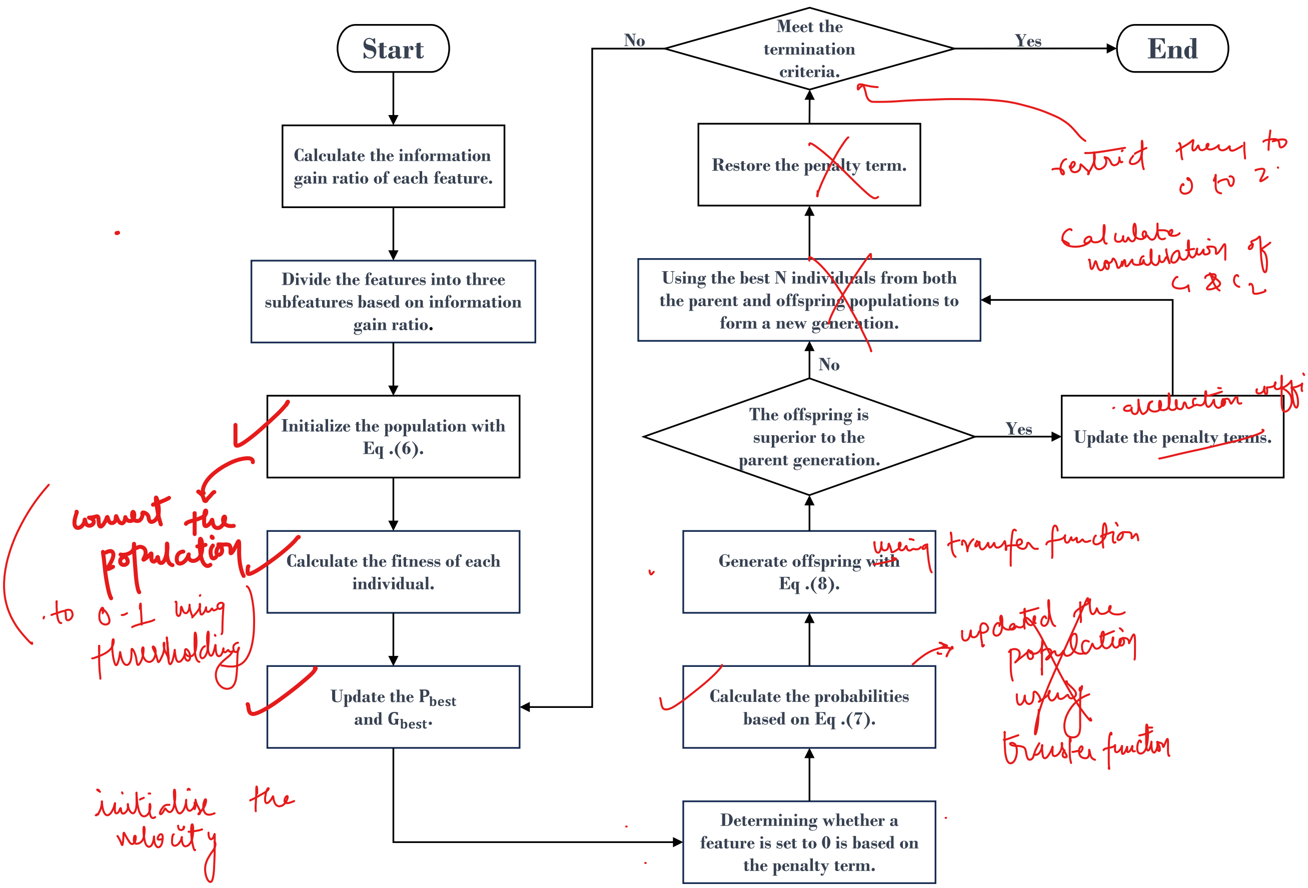

Fig. 1. Flowchart of ISPSO.

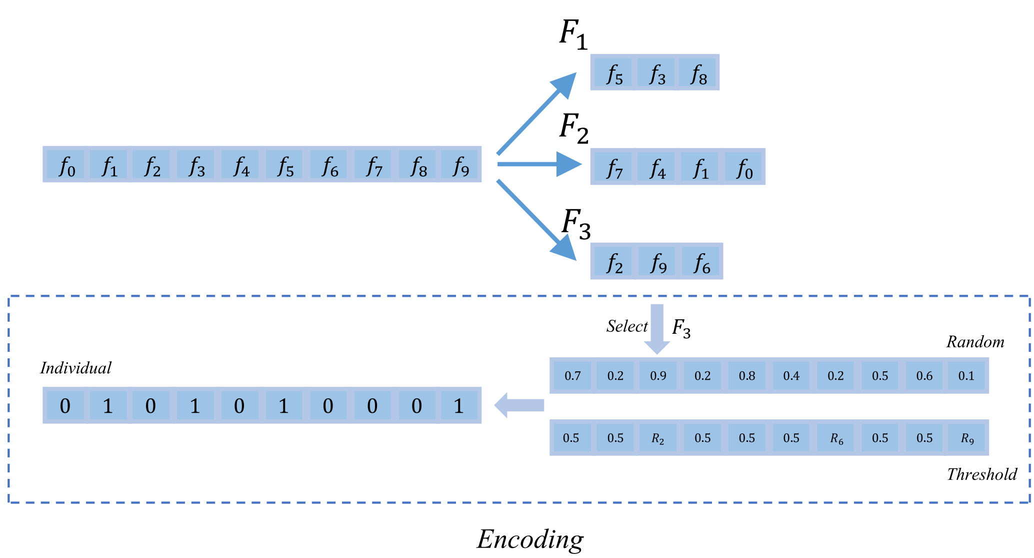

Fig. 2. Initialization based on subfeature.

status of the same feature in the new individual. The reduction in penalty terms is triggered only when a specific feature, selected in the original individual, is omitted in the improved new individual. Repeated reductions in the penalty term signify improved algorithm performance across various instances, particularly when a feature remains unselected. When the penalty term decreases below zero, ISPSO automatically sets the corresponding feature's value to zero. This procedure constricts the algorithm's search space, leading to enhanced search efficiency. To avert the risk of the algorithm becoming trapped in local optima due to prolonged unselection of certain features, the penalty term is progressively increased in each iteration. The formula for this incremental increase is

$$PT_j = PT_j + 0.05 \cdot R_j \tag{10}$$

where the coefficient is set to 0.05 to prevent the penalty term from increasing too rapidly. In addition, using the information gain ratio ensures that important features recover faster than redundant ones, preventing them from being left unselected for extended periods.

Algorithm 1: The pseudo-code of ISPSO.

```
Calculate the features' information gain ratio;
Divide the features into three subfeature groups;
for each individual do
    Select the subfeature randomly;
    Initialize the individual based on Eq. (6);
end
Set the PT of each feature to 10;
Update the P_best and G_best;
while t < T and FES < MaxFES do
    for i = 1 : N do
        for j = 1 : D do
            if PT_j ≤ 0 then
                S_{i,j} = 0
            end
            else
                Calculate the S_{i,j} based on Eqs. (7)–(8);
            end
        end
        if S_i is superior to X_i then
            Update the PT with Eq. (9);
        end
        Restore the penalty terms by Eq. (10);
        Combine the next population by selecting top N best
          individuals from the parent and the offspring;
        Update the P_best and G_best;
    end
    t = t + 1;
    FES = FES + N;
end
```

Algorithm 1 outlines the primary procedure of ISPSO. In this algorithm, t denotes the current iteration count, and T denotes the maximum number of iterations. FES represents the current evaluation count, while $MaxFES$ denotes the maximum number of evaluations. In addition, S_i denotes the offspring individuals, and X_i represents the parent individuals. In ISPSO, when generating a particular dimension for each individual, it is first determined whether to set this dimension to zero based on the corresponding penalty term of the dimension. Notably, the penalty term is updated solely when the offspring generation outperforms their respective parent generation and the penalty term is also restored after the generation of each individual.

3.4. Computational complexity analysis

The computational complexity of the proposed algorithm can be estimated based on Algorithm 1. Initially, ISPSO computes the information gain ratio for each of the D features, necessitating a total of $O(N_v \times D)$ basic operations, where N_v denotes the number of possible values for each feature. The values of N_v range from a minimum of 1 to a maximum potentially equal to the population size N. Consequently, computing the information gain ratio may require up to $O(N \times D)$ basic operations. In the main loop, the probability search operator accounts for the majority of the computational complexity, requiring $O(N \times D)$ basic operations to update each feature for every individual. However, each individual updates the penalty term once, resulting in a total of N updates. Therefore, the computational complexity of ISPSO is $O(N \times D)$.

3.5. General remarks on the proposed method

Incorporating the information gain ratio, this paper introduces a hybrid method known as ISPSO. The subfeature grouping and initialization method, outlined in Section 3.1 significantly differ from conventional random initialization techniques. The proposed approach, following the assessment of feature importance, increases the likelihood of selecting crucial features. Furthermore, the random selection method augments the diversity within the initial population.

As detailed in Section 3.2, the search operator replaces the traditional PSO's concept of velocity with probabilities to represent the likelihood of a feature being selected, making it more suitable for feature selection problems. The penalty term, discussed in Section 3.3, directly assigns certain features to zero during multiple learning iterations, effectively reducing the search space and enhancing the algorithm's search capabilities. The efficacy of these methods was verified in Section 4.2.6.

Table 1
The details of datasets.

No.	Name	Instances	Features	Classes
1	Australian	690	14	2
2	Zoo	101	16	7
3	German	1000	24	4
4	Ionosphere	351	34	2
5	Dermatology	366	34	6
6	KrVsKpEW	3196	36	2
7	Spambase	4601	57	2
8	Sonar	208	60	2
9	Hillvalley	1212	100	2
10	MUSK1	476	166	2
11	Semeion handwritten digit	1593	256	10
12	Arrhythmia	452	279	16
13	Madelon	2600	500	2
14	Isolet5	1559	617	26
15	Yale32	165	1024	15
16	ORL32	400	1024	40

4. Experiment setup

4.1. Experimental design

All experiments were conducted using an Intel i9-12900K 3.20 GHz CPU with 32 GB of memory. To comprehensively evaluate the algorithm's performance from various perspectives, we employed multiple evaluation metrics, including minimum classification error (MCE), precision, recall, F1-score, and area under the ROC curve (AUC). In addition, we utilized the Wilcoxon signed-rank test and the Wilcoxon rank-sum test to compare differences in algorithm performance.

4.1.1. Description of metrics

MCE is a widely accepted criterion in machine learning. Its primary objective is to minimize the classification error rate, which involves reducing the number of incorrectly classified samples in classification tasks. A lower MCE value indicates enhanced algorithm performance.

Precision measures the proportion of true positive samples among all samples predicted as positive. Higher precision denotes greater accuracy in the model's prediction of positive samples. Conversely, recall quantifies the proportion of true positive samples among all actual positive samples. A model with higher recall is more effective in identifying positive samples. Since precision and recall involve a trade-off, the F1-score is computed as their weighted harmonic mean, offering a balanced evaluation of the model's precision and recall. The AUC, representing the area under the ROC curve, evaluates the model's capability to differentiate between positive and negative samples.

4.1.2. Datasets

In this study, we selected 16 datasets to validate the effectiveness of our proposed method. The details are outlined in Table 1. The datasets range from 14 to 1024 features, with a maximum of 40 classes in any given dataset. These datasets have been widely used in previous studies [41–43] to evaluate algorithm performance.

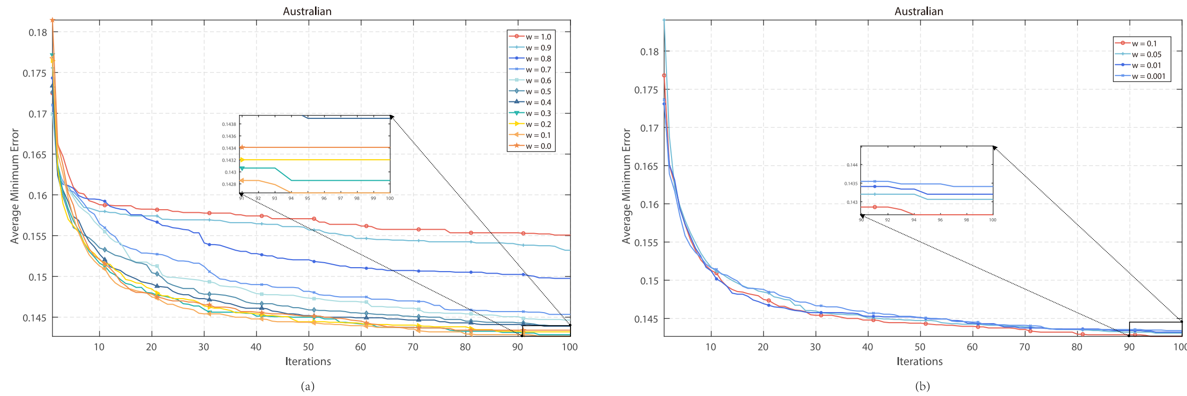

Fig. 3. The average classification error with different value of w on the Australian dataset.

4.1.3. Baseline techniques

To assess the performance of our proposed method, we compared it with seven existing EAs for feature selection. The selected algorithms are ESAPSO [27], SPACO [24], AAPSO [44], VS-CCPSO [32], CPBGSA [45], VLPSO [31], and WOA [46]. The rationale for choosing these algorithms includes:

(1) ESAPSO distinguishes itself by representing each particle with a selected feature subset and employing a feature grouping strategy based on feature importance. It also features an adaptive expansion strategy, allowing the swarm to autonomously decide on the next feature group to include.
(2) SPACO, a novel algorithm designed for feature selection, uniquely combines symmetric uncertainty with ant colony algorithms through an innovative graphical representation.
(3) AAPSO enhances the traditional approach using a memory-based adaptation parameter and introducing altruistic behavior among agents. Higher-ranking individuals occasionally assist lower-ranking ones, fostering a more comprehensive exploration of the search space.
(4) In VS-CCPSO, the feature space is divided based on feature importance, and the subpopulation size is adaptively adjusted. This algorithm has shown remarkable efficacy in feature selection contexts.
(5) CPBGSA integrates a clustering method into the traditional BGSA to distribute the initial population as broadly as possible across the search space.
(6) VLPSO rearranges features according to their correlations, allowing for different or shorter particle lengths. This approach reduces the search space and thereby enhances algorithm performance.
(7) WOA, a recent evolutionary algorithm, mimics the hunting behavior of whales and has been applied in feature selection.

Table 2
The average classification error with different value with different configurations.

K	K = 10	K = 7	K = 5	K = 3
	1.446E−01	1.442E−01	1.427E−01	1.433E−01
Rank	4	3	1	2
Proportion	1:8:1	2:6:2	3:4:3	4:2:4
	1.454E−01	1.465E−01	1.427E−01	1.430E−01
Rank	3	4	1	2

Table 3
Parameter settings.

Comparison algorithms	Parameters	Values
ESAPSO	Number of feature grouping (M) Gbest no-change thresholds (λ_1, λ_2) MIC threshold for feature filtering (θ)	$M = 5$ $\lambda_1 = 5$, $\lambda_2 = 15$ $\theta = 0.45 * MIC_{max}$
SPACO	Pheromone value significance (α) Heuristic information weight (β) Evaporation rate (ρ)	$\alpha = 0.7$ $\beta = 0.1$ $\rho = 0.5$
AAPSO	Elite agents proportion (k)	$k = 0.4$
VS-CCPSO	Feature importance (cd) threshold for elimination (δ) Number of subspaces (M)	$\delta = 0.1 * cd_{max}$ $M = 4$
CPBGSA	Term weightage (α) Term weightage (β) Term weightage (η)	$\alpha = 0.4$ $\beta = 0.7$ $\eta = 0.3$
VLPSO	Number of divisions Maximum iterations	12 9
WOA	Spiral shape parameter (b)	$b = 0.5$

4.1.4. Parameter settings

The value of w in Eq. (7) is a crucial parameter in ISPSO. In this algorithm, w largely determines the influence of the parent generation on the offspring. A higher value of w signifies a stronger influence from the parent generation, potentially increasing the probability of feature selection. However, this increase might lead to the selection of more features, adversely affecting the algorithm's performance when excessive. To investigate w's impact, we assigned it 11 distinct values ranging from 0 to 1, specifically {0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1}. We then conducted 30 trials on the Australian dataset, chosen for its low computational cost. The average minimum error across training sets was calculated for each value of w, with results depicted in Fig. 3(a). Notably, as w increased from 0.1 to 1, the algorithm's performance gradually declined. This suggests that diminishing the parent generation's influence on the offspring enhances performance. To further examine if smaller w values yield superior outcomes, we conducted experiments with four additional settings: {0.1, 0.05, 0.01, 0.001}, as illustrated in Fig. 3(b). The optimal performance was observed at $w = 0.1$. In addition, we explored the optimal grouping proportion and settings for the penalty term coefficient (K). Experiments were conducted on the Australian dataset with varying configurations: K values of {10, 7, 5, 3} and grouping ratios of {1:8:1, 2:6:2, 3:4:3, 4:2:4}, keeping all other conditions constant. The findings, presented in Table 2, indicate that the algorithm achieved its best solution with $K = 5$ and a grouping ratio of 3:4:3, thereby validating the effectiveness of these parameter settings.

The population size is consistently maintained at 30, the maximum number of iterations is established at 100, and the evaluation limit is capped at 3000. The algorithm terminates upon meeting any of the specified termination conditions. Given that feature selection is inherently a binary task, the lower and upper boundaries are defined as 0 and 1, respectively. To assess and validate the algorithms' performance, the dataset is divided into two segments: 70% for training

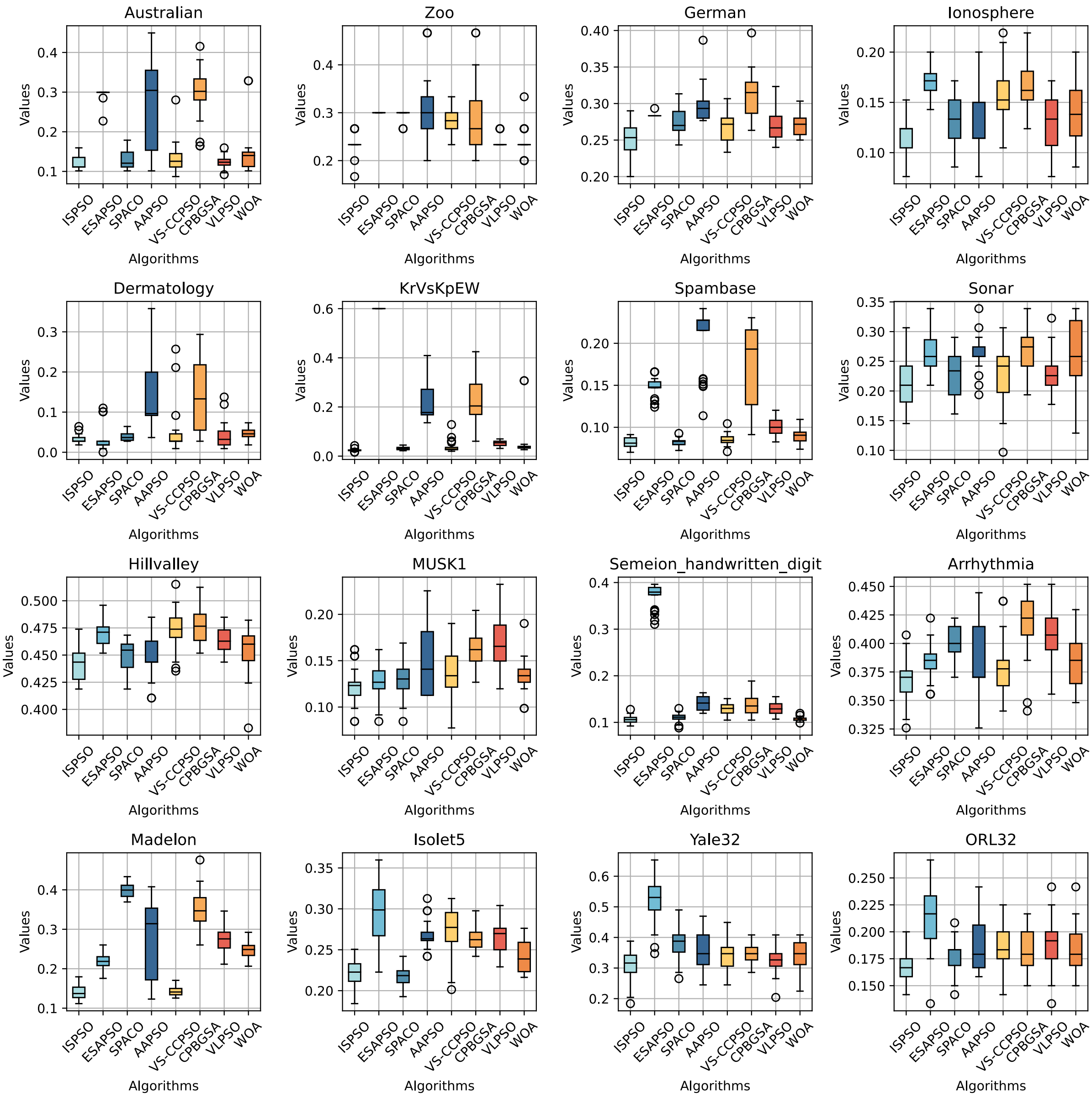

Fig. 4. Box plots of all testing datases.

and 30% for testing. A k-nearest neighbor classifier, with k set to 5 and frequently used in numerous experiments [47,48], serves as the classifier. During the training phase, optimized feature sets are derived through the selection process. These optimized sets are then applied to the testing set, enabling the evaluation of the algorithm's effectiveness on identical datasets. To guarantee the reliability and generalizability of the outcomes, each algorithm undergoes 30 independent executions on the same dataset. The optimum results from these 30 iterations are averaged to formulate the final result for subsequent analysis and comparison. The parameters for each algorithm under comparison are detailed in Table 3.

4.2. Experimental results

To thoroughly analyze the proposed algorithm's performance, this section utilizes five metrics as evaluation criteria for assessing ISPSO. Moreover, visualizing the results facilitates a more intuitive analysis of the outcomes.

4.2.1. Performance analysis

This section adopts the average MCE to intuitively measure the algorithm's effectiveness, where a lower value signifies enhanced performance. Experiments were carried out using ISPSO and its comparative algorithms on 16 datasets. The outcomes of these experiments are concisely summarized in Table 4, with the best results emphasized in bold.

From Table 4, ISPSO outperforms its competitors on 14 datasets and ranks second in performance on the remaining datasets. For further analysis, we employed the Wilcoxon signed-rank test [49] and the Wilcoxon rank-sum test [50] to assess whether ISPSO's performance shows statistically significant superiority over other comparative algorithms. These non-parametric hypothesis tests use a significance level of 0.05 to evaluate if the results have statistically significant differences. The p-value from the Wilcoxon signed-rank test is indicative of these differences. Table 4 shows that ISPSO significantly surpasses other algorithms in performance. In addition, the $W/T/L$ values in Table 4 represent the outcomes of the Wilcoxon rank-sum test. W, T, and L correspond to the symbols "+", "=", and "−", respectively, as

Table 4
The average and standard deviation of the MCE on the testing sets.

No.	ISPSO			ESAPSO			SPACO			AAPSO		
	Avg.	Std.		Avg.	Std.		Avg.	Std.		Avg.	Std.	
1	**1.180E−01**	1.643E−02		2.966E−01	1.340E−02	(+)	1.275E−01	2.060E−02	(+)	2.667E−01	1.099E−01	(+)
~~2~~	**2.344E−01**	1.854E−02		3.000E−01	5.646E−17	(+)	2.978E−01	8.457E−03	(+)	3.133E−01	6.644E−02	(+)
3	**2.503E−01**	2.177E−02		2.837E−01	1.826E−03	(+)	2.741E−01	1.933E−02	(+)	2.990E−01	2.334E−02	(+)
4	**1.159E−01**	1.971E−02		1.705E−01	1.666E−02	(+)	1.327E−01	2.250E−02	(+)	1.270E−01	3.026E−02	(=)
5	3.272E−02	1.122E−02		**2.997E−02**	2.999E−02	(−)	3.976E−02	1.032E−02	(+)	1.456E−01	7.964E−02	(+)
6	**2.387E−02**	4.871E−03		6.002E−01	4.517E−16	(+)	3.132E−02	5.638E−03	(+)	2.239E−01	7.192E−02	(+)
7	**8.164E−02**	5.835E−03		1.491E−01	1.137E−02	(+)	8.225E−02	4.873E−03	(=)	2.074E−01	3.469E−02	(+)
8	**2.156E−01**	4.095E−02		2.656E−01	3.612E−02	(+)	2.269E−01	3.465E−02	(+)	2.629E−01	2.684E−02	(+)
9	**4.421E−01**	1.603E−02		4.697E−01	1.111E−02	(+)	4.493E−01	1.457E−02	(=)	4.492E−01	1.888E−02	(=)
10	**1.237E−01**	1.910E−02		1.286E−01	1.734E−02	(=)	1.305E−01	1.810E−02	(+)	1.465E−01	3.673E−02	(+)
11	**1.061E−01**	7.776E−03		3.718E−01	2.588E−02	(+)	1.110E−01	9.162E−03	(+)	1.415E−01	1.594E−02	(+)
~~12~~	**3.669E−01**	1.791E−02		3.827E−01	1.460E−02	(+)	4.015E−01	1.435E−02	(+)	3.859E−01	2.997E−02	(+)
13	**1.408E−01**	1.834E−02		2.188E−01	1.834E−02	(+)	3.985E−01	1.746E−02	(+)	2.774E−01	9.493E−02	(+)
14	2.218E−01	1.481E−02		2.955E−01	3.502E−02	(+)	**2.177E−01**	1.152E−02	(=)	2.670E−01	1.536E−02	(+)
~~15~~	**3.061E−01**	5.306E−02		5.224E−01	7.457E−02	(+)	3.776E−01	4.756E−02	(+)	3.517E−01	6.836E−02	(+)
~~16~~	**1.681E−01**	1.486E−02		2.164E−01	3.220E−02	(+)	1.772E−01	1.417E−02	(+)	1.875E−01	2.365E−02	(+)
p-value	–			2.923E−04			3.534E−04			2.412E−04		
W/T/L	−/−/−			14/1/1			13/3/0			14/2/0		
No.	VS-CCPSO			CPBGSA			VLPSO			WOA		
	Avg.	Std.		Avg.	Std.		Avg.	Std.		Avg.	Std.	
1	1.325E−01	3.483E−02	(+)	2.976E−01	5.928E−02	(+)	1.245E−01	1.557E−02	(+)	1.420E−01	5.358E−02	(+)
2	2.844E−01	3.581E−02	(+)	2.911E−01	7.162E−02	(+)	2.367E−01	1.017E−02	(=)	2.378E−01	2.434E−02	(=)
3	2.693E−01	1.940E−02	(+)	3.113E−01	2.982E−02	(+)	2.713E−01	2.503E−02	(+)	2.720E−01	1.461E−02	(+)
4	1.537E−01	2.748E−02	(+)	1.660E−01	2.382E−02	(+)	1.260E−01	2.736E−02	(+)	1.400E−01	3.204E−02	(+)
5	4.740E−02	5.361E−02	(=)	1.437E−01	9.006E−02	(+)	4.037E−02	2.917E−02	(=)	4.618E−02	1.352E−02	(+)
6	3.619E−02	2.144E−02	(+)	2.354E−01	9.176E−02	(+)	5.285E−02	1.063E−02	(+)	5.442E−02	6.884E−02	(+)
7	8.524E−02	6.316E−03	(+)	1.729E−01	4.745E−02	(+)	1.012E−01	9.514E−03	(+)	9.012E−02	8.034E−03	(+)
8	2.339E−01	5.337E−02	(+)	2.651E−01	3.733E−02	(+)	2.317E−01	3.401E−02	(+)	2.597E−01	5.782E−02	(+)
9	4.746E−01	1.843E−02	(+)	4.766E−01	1.640E−02	(+)	4.640E−01	1.129E−02	(+)	4.558E−01	2.038E−02	(+)
10	1.352E−01	2.996E−02	(+)	1.627E−01	1.969E−02	(+)	1.716E−01	2.868E−02	(+)	1.345E−01	1.658E−02	(+)
11	1.290E−01	1.226E−02	(+)	1.358E−01	1.893E−02	(+)	1.294E−01	1.225E−02	(+)	1.067E−01	4.683E−03	(=)
12	3.785E−01	2.576E−02	(+)	4.180E−01	2.622E−02	(+)	4.089E−01	2.062E−02	(+)	3.852E−01	2.559E−02	(+)
13	1.424E−01	1.089E−02	(=)	3.478E−01	5.083E−02	(+)	2.781E−01	3.529E−02	(+)	2.472E−01	2.145E−02	(+)
14	2.728E−01	3.011E−02	(+)	2.642E−01	1.504E−02	(+)	2.659E−01	1.920E−02	(+)	2.419E−01	1.942E−02	(+)
15	3.422E−01	4.632E−02	(+)	3.503E−01	2.890E−02	(+)	3.286E−01	4.621E−02	(+)	3.449E−01	4.652E−02	(+)
16	1.869E−01	2.061E−02	(+)	1.842E−01	1.949E−02	(+)	1.883E−01	2.472E−02	(+)	1.822E−01	2.107E−02	(+)
p-value	2.412E−04			2.412E−04			2.412E−04			2.412E−04		
W/T/L	14/2/0			16/0/0			14/2/0			14/2/0		

depicted in the table. The symbol "+" represent that ISPSO significantly outperforms the corresponding algorithms on the current dataset, while "−" signifies that ISPSO significantly underperforms compared to the respective algorithm. The "=" symbol denotes no significant difference between ISPSO and its competitors. The results demonstrate that ISPSO significantly outperformed CPBGSA across all datasets. Compared with ESAPSO, ISPSO is weaker on the fourth dataset but significantly outperforms ESAPSO on the other 14 datasets. This suggests that, although both methods rely on feature importance for feature grouping, the grouping approach proposed in this paper is more advantageous. While SPACO achieved the lowest MCE value on the fourteenth dataset, ISPSO exhibited superior performance on 13 datasets. Compared to AAPSO, VS-CCPSO, VLPSO, and WOA, ISPSO exceeds all 14 datasets and matches or exceeds their performance on the remaining datasets.

4.2.2. Visual analysis

To offer a more intuitive analysis of the algorithm's performance, we have presented the experimental results using box plots in Fig. 4. In these box plots, the lower horizontal line at the bottom signifies the minimum value in the dataset, and the upper horizontal line represents the maximum value. The box's size reflects the dispersion or spread of the results, while the vertical line within the box indicates the median. According to Fig. 4, ISPSO achieved the best results in terms of minimum values on 10 datasets. Furthermore, ISPSO's median value is lower than all other comparative algorithms on 11 datasets, suggesting its higher accuracy on these datasets. Most notably, on the Madelon dataset, the maximum values obtained by ISPSO are lower than the minimum values achieved by the other four algorithms.

In addition, to further assess the algorithms' performance, receiver operating characteristic (ROC) curves were presented for each algorithm, corresponding to nine different problems, as shown in Fig. 5. ROC curves are constructed based on the true positive rate (TPR) and false positive rate (FPR) of a classifier. TPR reflects the classifier's ability to correctly identify samples from the positive class, while FPR indicates the classifier's propensity to incorrectly classify samples from the negative class as positive. Ideally, TPR should approach 1, and FPR should approach 0. Thus, the closer the ROC curve is to the point (0,1), the more effective the classifier's performance. In this study, due to the unknown nature of the testing datasets, the best feature set from each of the eight algorithms in the training sets was selected to plot the ROC curves for the testing sets. As illustrated in Fig. 5, ISPSO is notably closer to the (0,1) point than the other algorithms on four datasets, indicating superior performance on these datasets. Moreover, ISPSO's performance remains competitive with its peers on the remaining datasets.

4.2.3. Analysis of other metrics

In this section, we utilized four commonly used metrics for classification problems: precision, recall, F1-score, and AUC, to further validate the superior performance of the proposed algorithm. It is important to note that all these metrics are considered better when they are higher. The results for these metrics are presented in Tables 5 to 8.

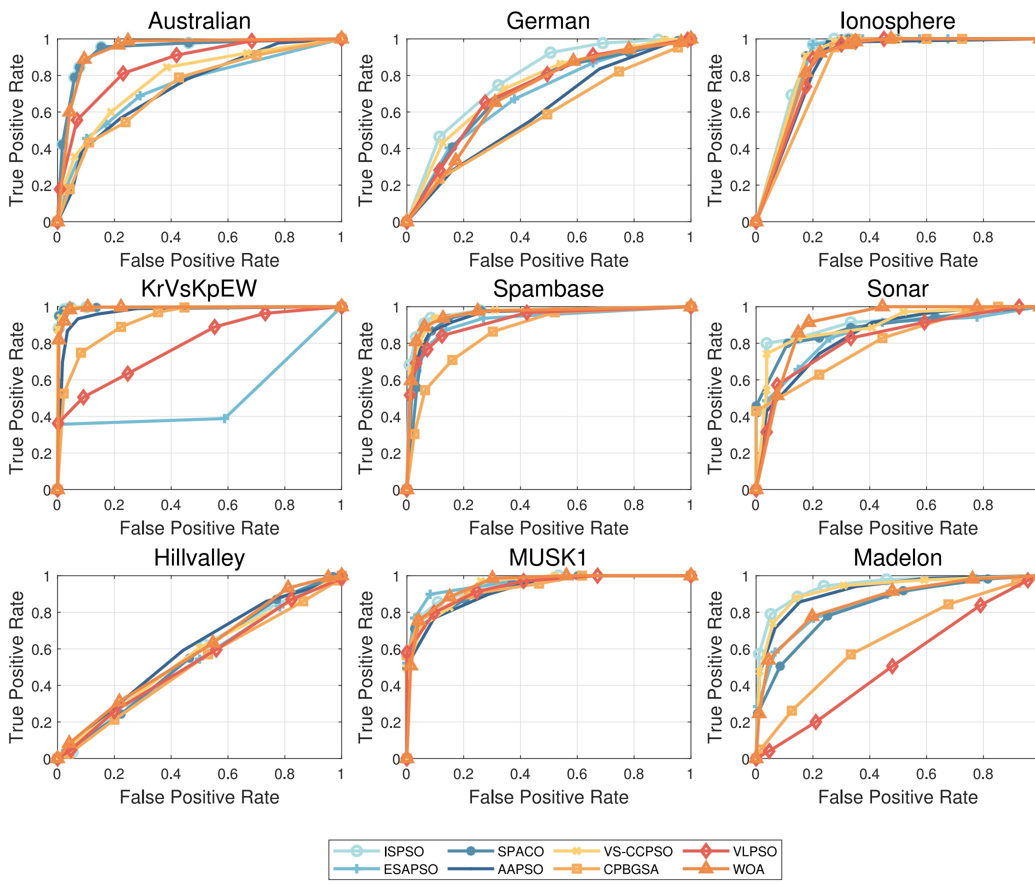

Fig. 5. Receiver operating characteristic curves for nine datasets during the testing process.

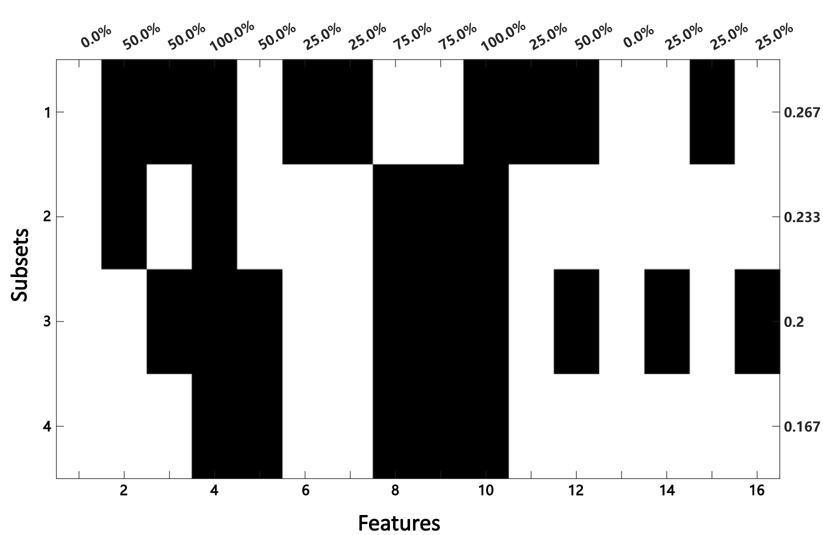

Fig. 6. Frequency matrix on the Zoo dataset of ISPSO.

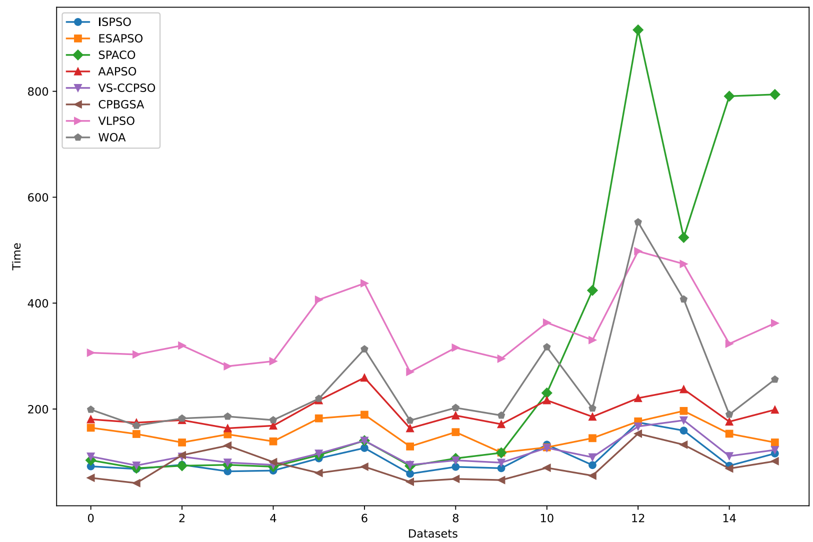

Fig. 7. The average running time of eight algorithms on all datasets.

Table 5
The average and standard deviation of precision on the testing sets.

No.	ISPSO		ESAPSO		SPACO		AAPSO	
	Avg.	Std.	Avg.	Std.	Avg.	Std.	Avg.	Std.
1	**8.809E−01**	1.700E−02	7.046E−01	1.327E−02	8.715E−01	2.084E−02	7.442E−01	9.959E−02
~~2~~	6.443E−01	5.313E−02	6.319E−01	1.129E−16	6.428E−01	2.620E−02	5.973E−01	9.492E−02
2	**6.929E−01**	3.237E−02	6.404E−01	2.592E−03	6.586E−01	2.843E−02	6.033E−01	5.431E−02
3	**9.122E−01**	1.861E−02	8.808E−01	1.807E−02	9.067E−01	1.696E−02	8.980E−01	1.981E−02
4	9.624E−01	1.121E−02	**9.676E−01**	3.083E−02	9.328E−01	5.388E−02	8.294E−01	1.008E−01
5	**9.764E−01**	4.880E−03	4.003E−01	0.000E+00	9.695E−01	5.218E−03	7.641E−01	7.957E−02
6	8.344E−01	7.801E−03	8.058E−01	1.866E−02	**8.396E−01**	7.334E−03	7.387E−01	5.454E−02
7	**9.178E−01**	6.228E−03	8.516E−01	1.389E−02	7.988E−01	3.421E−02	8.220E−01	4.740E−02
8	**7.915E−01**	4.214E−02	7.378E−01	3.669E−02	7.769E−01	3.831E−02	7.359E−01	3.718E−02
9	**5.550E−01**	1.634E−02	5.277E−01	1.139E−02	5.480E−01	1.474E−02	5.501E−01	1.874E−02
10	**8.774E−01**	1.942E−02	8.729E−01	1.718E−02	8.722E−01	1.822E−02	8.442E−01	2.403E−02
11	**9.014E−01**	7.127E−03	6.348E−01	2.686E−02	8.980E−01	8.765E−03	8.777E−01	1.350E−02
~~12~~	3.268E−01	4.113E−02	2.869E−01	2.702E−02	3.274E−01	4.824E−02	3.307E−01	6.068E−02
13	**8.594E−01**	1.831E−02	7.815E−01	1.830E−02	7.603E−01	1.223E−02	7.217E−01	1.007E−01
~~14~~	8.110E−01	1.594E−02	7.174E−01	3.795E−02	**8.169E−01**	1.278E−02	7.642E−01	1.616E−02
~~15~~	**7.674E−01**	6.644E−02	5.226E−01	8.467E−02	6.185E−01	6.723E−02	7.065E−01	9.223E−02
~~16~~	**8.768E−01**	1.479E−02	8.399E−01	3.030E−02	8.699E−01	1.404E−02	8.576E−01	2.441E−02
p-value	–		3.534E−04		8.789E−04		2.923E−04	

No.	VS-CCPSO		CPBGSA		VLPSO		WOA	
	Avg.	Std.	Avg.	Std.	Avg.	Std.	Avg.	Std.
1	7.346E−01	5.834E−02	6.983E−01	6.150E−02	6.976E−01	5.795E−02	8.572E−01	4.953E−02
2	5.751E−01	7.659E−02	6.124E−01	8.861E−02	5.363E−01	9.590E−02	**6.538E−01**	4.662E−02
3	6.655E−01	2.897E−02	6.011E−01	4.450E−02	6.543E−01	3.073E−02	6.616E−01	2.126E−02
4	8.962E−01	1.782E−02	8.942E−01	1.121E−02	8.728E−01	4.179E−02	8.918E−01	2.862E−02
5	9.497E−01	5.601E−02	8.435E−01	1.019E−01	8.857E−01	5.582E−02	9.501E−01	1.203E−02
6	9.639E−01	2.109E−02	7.663E−01	9.155E−02	6.673E−01	6.415E−02	9.517E−01	5.551E−02
7	9.145E−01	6.639E−03	8.279E−01	4.831E−02	8.246E−01	2.994E−02	9.090E−01	8.587E−03
8	7.706E−01	5.253E−02	7.421E−01	4.065E−02	7.438E−01	5.633E−02	7.417E−01	5.901E−02
9	5.217E−01	1.884E−02	5.190E−01	1.652E−02	5.119E−01	1.637E−02	5.415E−01	2.076E−02
10	8.663E−01	3.006E−02	8.450E−01	1.711E−02	8.268E−01	2.188E−02	8.680E−01	1.716E−02
11	8.801E−01	1.291E−02	8.748E−01	1.735E−02	8.578E−01	1.927E−02	9.020E−01	4.464E−03
12	3.313E−01	4.957E−02	2.900E−01	6.321E−02	2.721E−01	4.808E−02	**3.475E−01**	6.740E−02
13	8.578E−01	1.074E−02	6.532E−01	5.096E−02	5.019E−01	1.714E−02	7.533E−01	2.148E−02
14	7.619E−01	3.253E−02	7.727E−01	2.000E−02	7.629E−01	1.644E−02	7.930E−01	2.223E−02
15	7.319E−01	7.114E−02	7.255E−01	4.580E−02	6.387E−01	8.626E−02	7.242E−01	6.122E−02
16	8.611E−01	2.281E−02	8.666E−01	1.587E−02	8.473E−01	2.631E−02	8.654E−01	1.619E−02
p-value	4.262E−04		2.412E−04		2.412E−04		3.317E−03	

As shown in the tables, ISPSO exhibits superior performance across all metrics on 12 datasets. Regarding AUC evaluation, ISPSO outperforms its competitors on 14 datasets. Furthermore, the results of the Wilcoxon signed-rank test indicate that ISPSO significantly outperforms other algorithms in these four metrics, further underscoring the superiority of the ISPSO algorithm.

4.2.4. Analysis of selected features

In this section, to verify whether the selected subset has achieved the goal of feature selection, namely, selecting important features while eliminating irrelevant and redundant ones, a frequency matrix [51] was generated. This matrix pertains to four individuals with differing MCE values obtained by ISPSO on the Zoo dataset, as illustrated in

Table 6
The average and standard deviation of recall on the testing sets.

No.	ISPSO		ESAPSO		SPACO		AAPSO	
	Avg.	Std.	Avg.	Std.	Avg.	Std.	Avg.	Std.
1	**8.798E−01**	1.669E−02	6.846E−01	1.492E−02	8.691E−01	2.148E−02	7.387E−01	1.030E−01
2	7.371E−01	5.843E−02	7.224E−01	3.388E−16	7.241E−01	6.305E−03	6.556E−01	8.577E−02
3	**6.630E−01**	2.880E−02	6.065E−01	1.286E−03	6.361E−01	2.577E−02	5.820E−01	4.651E−02
4	**8.518E−01**	2.501E−02	7.798E−01	2.219E−02	8.276E−01	2.879E−02	8.413E−01	4.033E−02
5	**9.732E−01**	1.164E−02	9.661E−01	4.576E−02	9.422E−01	5.712E−02	8.311E−01	1.003E−01
6	**9.760E−01**	4.853E−03	4.002E−01	5.646E−17	9.694E−01	5.224E−03	7.613E−01	7.992E−02
7	**9.150E−01**	5.967E−03	8.424E−01	1.023E−02	7.892E−01	3.601E−02	8.113E−01	5.071E−02
8	**7.715E−01**	4.460E−02	7.195E−01	4.049E−02	7.608E−01	3.611E−02	7.112E−01	3.118E−02
9	**5.544E−01**	1.626E−02	5.275E−01	1.133E−02	5.476E−01	1.468E−02	5.502E−01	1.877E−02
10	**8.769E−01**	1.917E−02	8.721E−01	1.716E−02	8.705E−01	1.802E−02	8.373E−01	2.655E−02
11	**8.956E−01**	7.859E−03	6.333E−01	2.523E−02	8.907E−01	9.235E−03	8.683E−01	1.321E−02
12	2.703E−01	3.319E−02	2.667E−01	4.094E−02	2.227E−01	3.251E−02	2.413E−01	5.520E−02
13	**8.592E−01**	1.834E−02	7.812E−01	1.834E−02	7.598E−01	1.219E−02	7.211E−01	1.008E−01
14	7.784E−01	1.482E−02	7.042E−01	3.523E−02	**7.831E−01**	1.263E−02	7.312E−01	1.286E−02
15	**7.133E−01**	5.207E−02	5.011E−01	7.563E−02	5.854E−01	6.027E−02	6.615E−01	6.849E−02
16	**8.319E−01**	1.486E−02	7.836E−01	3.220E−02	8.228E−01	1.417E−02	8.125E−01	2.365E−02
p-value	–		2.412E−04		2.923E−04		2.412E−04	

No.	VS-CCPSO		CPBGSA		VLPSO		WOA	
	Avg.	Std.	Avg.	Std.	Avg.	Std.	Avg.	Std.
1	7.311E−01	6.108E−02	6.910E−01	6.166E−02	6.949E−01	5.842E−02	8.593E−01	5.250E−02
2	6.659E−01	7.982E−02	6.907E−01	9.231E−02	5.866E−01	9.246E−02	**7.470E−01**	5.467E−02
3	6.359E−01	2.552E−02	5.828E−01	3.895E−02	6.287E−01	2.864E−02	6.359E−01	1.879E−02
4	8.003E−01	3.695E−02	7.822E−01	3.156E−02	8.186E−01	3.016E−02	8.225E−01	4.113E−02
5	9.522E−01	5.834E−02	8.488E−01	1.015E−01	8.681E−01	6.397E−02	9.625E−01	1.214E−02
6	9.642E−01	2.157E−02	7.633E−01	9.260E−02	6.605E−01	6.351E−02	9.516E−01	5.546E−02
7	9.108E−01	6.514E−03	8.163E−01	5.020E−02	8.015E−01	3.530E−02	9.064E−01	7.898E−03
8	7.525E−01	5.852E−02	7.183E−01	4.172E−02	7.249E−01	5.083E−02	7.277E−01	6.144E−02
9	5.215E−01	1.865E−02	5.185E−01	1.607E−02	5.117E−01	1.621E−02	5.413E−01	2.077E−02
10	8.653E−01	2.999E−02	8.393E−01	1.926E−02	8.212E−01	2.300E−02	8.664E−01	1.667E−02
11	8.713E−01	1.235E−02	8.654E−01	1.895E−02	8.430E−01	2.252E−02	8.953E−01	4.540E−03
12	**2.759E−01**	4.826E−02	2.184E−01	6.397E−02	1.983E−01	3.168E−02	2.590E−01	4.951E−02
13	8.576E−01	1.089E−02	6.522E−01	5.083E−02	5.019E−01	1.713E−02	7.528E−01	2.145E−02
14	7.275E−01	3.002E−02	7.360E−01	1.504E−02	7.249E−01	1.346E−02	7.583E−01	1.939E−02
15	6.755E−01	4.540E−02	6.689E−01	2.846E−02	6.054E−01	6.984E−02	6.746E−01	4.652E−02
16	8.130E−01	2.061E−02	8.158E−01	1.949E−02	7.889E−01	2.831E−02	8.178E−01	2.107E−02
p-value	4.262E−04		2.412E−04		2.412E−04		4.262E−04	

Fig. 6. In the frequency matrix, each row represents a feature subset obtained by the algorithm, and each column corresponds to a feature of the dataset. If the mth feature is selected in the nth subset, the corresponding position (m, n) in the matrix is shaded. The right side of the frequency matrix displays the classification error achieved by each subset on the testing sets, while the top side shows the frequency of feature selection across all subsets. This frequency represents the ratio of the number of times a feature is chosen to the total number of subsets.

In Fig. 6, the MCE of ISPSO is 0.167, significantly lower than that of VLPSO. ISPSO consistently selects two common features, F_4 and F_{10}, across all subsets, suggesting their importance. According to the UCL machine learning repository's description of the Zoo dataset [52], F_4 and F_{10} correspond to the presence of $milk$ and $breathes$, respectively. Drawing from biological knowledge and the discussion in [51], these features are crucial for differentiating animal species. This evidence supports the assertion that ISPSO effectively identifies key features. Moreover, in ISPSO, the information gain ratio values for these features are the highest, ranking first and second among all features. This ranking indicates the efficiency of ISPSO's information gain ratio in evaluating feature significance, enabling ISPSO to select essential features based on this metric. Furthermore, features such as $F_6(aquatic)$, $F_7(predator)$, and $F_{11}(venomous)$, which are less distinctive in classifying animals and appear infrequently in the frequency matrix. This observation underscores ISPSO's proficiency in discarding non-essential features.

Furthermore, considering the potential impact of redundant features on classification performance, experiments were conducted to evaluate the redundancy of features selected by ISPSO. These experiments focused on the fourth subset, S_4: $\{F_4, F_5, F_8, F_9, F_{10}\}$, which has the minimum number of features. Each feature was systematically removed from S_4, and the resulting classification error on testing sets was computed; the findings are presented in Table 9. The first row of Table 9 lists the features removed from the subset. The results indicate that the removal of any feature leads to an increased classification error, thereby demonstrating the non-redundancy of the selected features. This outcome further substantiates the ISPSO algorithm's efficacy in eliminating redundant features.

4.2.5. Computational cost

To better compare algorithm performance, the average CPU running times of eight algorithms on each dataset were computed, with results shown in Fig. 7. The figure illustrates that while ISPSO has the shortest running time on three datasets, it consistently outperforms CPBGSA across all datasets. CPBGSA shows the shortest running times on the other datasets. When excluding CPBGSA, ISPSO has the shortest running time on 13 datasets. This highlights that ISPSO not only exceeds the performance of the comparison algorithms but also requires less running time than its counterparts.

4.2.6. Ablation study

(1) Performance analysis: To validate the effectiveness of various strategies within the algorithm, we conducted a series of ablation experiments, the results of which are presented in Table 10. BPSO, as described in Section 2.1, represents the original algorithm. BPSO-1, an enhancement of BPSO, was developed by incorporating the proposed initialization method. Differing from BPSO-1, BPSO-2 introduces a

Table 7
The average and standard deviation of F1-score on the testing sets.

No.	ISPSO		ESAPSO		SPACO		AAPSO	
	Avg.	Std.	Avg.	Std.	Avg.	Std.	Avg.	Std.
1	**8.803E−01**	1.670E−02	6.944E−01	1.408E−02	8.703E−01	2.104E−02	7.414E−01	1.012E−01
2	6.467E−01	5.156E−02	6.040E−01	2.258E−16	6.118E−01	2.329E−02	5.815E−01	7.976E−02
3	**6.775E−01**	2.979E−02	6.229E−01	1.908E−03	6.470E−01	2.667E−02	5.924E−01	5.018E−02
4	**8.808E−01**	2.050E−02	8.270E−01	1.681E−02	8.652E−01	2.291E−02	8.684E−01	2.904E−02
5	**9.659E−01**	1.180E−02	9.643E−01	4.204E−02	9.350E−01	5.548E−02	8.254E−01	1.003E−01
6	**9.761E−01**	4.857E−03	4.002E−01	0.000E+00	9.694E−01	5.219E−03	7.626E−01	7.971E−02
7	**9.163E−01**	5.988E−03	8.469E−01	1.195E−02	7.939E−01	3.510E−02	8.166E−01	4.902E−02
8	**7.812E−01**	4.253E−02	7.284E−01	3.800E−02	7.687E−01	3.665E−02	7.232E−01	3.316E−02
9	**5.547E−01**	1.630E−02	5.276E−01	1.136E−02	5.478E−01	1.471E−02	5.501E−01	1.875E−02
10	**8.771E−01**	1.929E−02	8.724E−01	1.713E−02	8.713E−01	1.808E−02	8.407E−01	2.514E−02
11	**8.925E−01**	8.006E−03	6.273E−01	2.538E−02	8.875E−01	9.566E−03	8.643E−01	1.426E−02
12	2.780E−01	2.624E−02	2.592E−01	2.813E−02	2.388E−01	2.850E−02	2.545E−01	5.543E−02
13	**8.593E−01**	1.833E−02	7.813E−01	1.832E−02	7.600E−01	1.220E−02	7.213E−01	1.008E−01
14	7.691E−01	1.632E−02	6.935E−01	3.559E−02	**7.744E−01**	1.425E−02	7.211E−01	1.491E−02
15	**6.876E−01**	5.957E−02	4.620E−01	7.515E−02	5.617E−01	6.282E−02	6.376E−01	7.741E−02
16	**8.321E−01**	1.468E−02	7.826E−01	3.253E−02	8.230E−01	1.403E−02	8.085E−01	2.334E−02
p-value	–		2.412E−04		3.534E−04		2.412E−04	

No.	VS-CCPSO		CPBGSA		VLPSO		WOA	
	Avg.	Std.	Avg.	Std.	Avg.	Std.	Avg.	Std.
1	7.328E−01	5.964E−02	6.946E−01	6.152E−02	6.962E−01	5.809E−02	8.582E−01	5.099E−02
2	5.677E−01	7.020E−02	5.949E−01	8.826E−02	5.226E−01	8.694E−02	**6.537E−01**	5.128E−02
3	6.503E−01	2.629E−02	5.917E−01	4.141E−02	6.410E−01	2.785E−02	6.484E−01	1.944E−02
4	8.452E−01	2.647E−02	8.343E−01	2.283E−02	8.444E−01	3.134E−02	8.555E−01	3.321E−02
5	9.486E−01	5.877E−02	8.409E−01	1.009E−01	8.643E−01	6.162E−02	9.521E−01	1.384E−02
6	9.640E−01	2.133E−02	7.647E−01	9.206E−02	6.638E−01	6.367E−02	9.516E−01	5.549E−02
7	9.126E−01	6.488E−03	8.220E−01	4.922E−02	8.128E−01	3.246E−02	9.076E−01	8.183E−03
8	7.613E−01	5.517E−02	7.298E−01	4.015E−02	7.341E−01	5.306E−02	7.345E−01	5.995E−02
9	5.215E−01	1.874E−02	5.187E−01	1.630E−02	5.118E−01	1.629E−02	5.414E−01	2.076E−02
10	8.658E−01	3.002E−02	8.421E−01	1.808E−02	8.239E−01	2.227E−02	8.672E−01	1.686E−02
11	8.683E−01	1.292E−02	8.615E−01	1.999E−02	8.401E−01	2.325E−02	8.923E−01	4.854E−03
12	**2.786E−01**	3.854E−02	2.231E−01	5.785E−02	2.051E−01	2.918E−02	2.689E−01	4.606E−02
13	8.577E−01	1.081E−02	6.526E−01	5.090E−02	5.019E−01	1.713E−02	7.530E−01	2.146E−02
14	7.151E−01	3.351E−02	7.251E−01	1.572E−02	7.114E−01	1.575E−02	7.481E−01	2.169E−02
15	6.492E−01	5.187E−02	6.409E−01	3.315E−02	5.872E−01	6.989E−02	6.462E−01	4.840E−02
16	8.114E−01	2.189E−02	8.154E−01	1.827E−02	7.893E−01	2.856E−02	8.184E−01	2.008E−02
p-value	2.923E−04		2.412E−04		2.412E−04		3.534E−04	

distinct search operator, specifically a probability-based one. Finally, BPSO-3 integrates an elite strategy in the final stage.

From Table 10, it is evident that ISPSO outperforms other algorithms on 10 datasets. The Wilcoxon signed-rank test was employed to determine if there are significant differences between ISPSO and its variants. The results confirm that ISPSO significantly surpasses the other algorithms on these datasets, demonstrating an improvement in performance due to these mechanisms. Notably, the Wilcoxon signed-rank test shows a value of 7.629E−04 for BPSO-1 compared to BPSO, well below the significance threshold of 0.05. This suggests a considerable enhancement in algorithm performance attributed to the use of an initialization method based on subfeature grouping.

(2) Selected features analysis: The primary objective of feature selection is eliminating redundant and irrelevant features. To verify the efficacy of the proposed mechanism in achieving this goal, this section calculates the proportions of selected features relative to the total number of features. These calculations are conducted for ISPSO, BPSO, and its three variants, with the results presented in Table 11.

From Table 11, it is observed that BPSO-2 selects the fewest features across all datasets. However, as shown in Table 10, it outperforms ISPSO on only three datasets and falls short on the others. Considerably, BPSO-2 exhibits markedly inferior performance compared to other algorithms on the sixth dataset. This implies that BPSO-2 might be omitting essential features, indicating that an excessively limited feature selection can detrimentally affect algorithm performance. Despite this, the results indicate that the novel search operator in BPSO-2 effectively reduces the search space. Excluding BPSO-2, ISPSO consistently selects the fewest features across all 12 datasets and delivers the best outcomes on nine of them. In the remaining two datasets, ISPSO secures second-best performance. To further ascertain the elimination of redundant and irrelevant features, we analyzed the Yale-32 dataset, which has the highest feature count. By employing the information gain ratio, we determined the most irrelevant feature. Over 30 runs, the frequency of the most irrelevant feature being selected by BPSO, BPSO-1, BPSO-3, and ISPSO were 20, 17, 16, and 13 times, respectively. The stepwise reduction in the number of selections indicates that the implemented mechanisms progressively minimize the inclusion of redundant and irrelevant features.

These results indicate that ISPSO enhances the algorithm's performance by generating a high-quality population through the proposed initialization method. This method considers the relationship between features and labels. The introduced probability-based search factor effectively narrows the search space, decreasing computational costs. Moreover, implementing the penalty term builds on this improvement, reducing the MCE and effectively decreasing the number of features selected by the algorithm.

5. Conclusion

In this study, we introduced ISPSO, a novel PSO approach. ISPSO integrates subfeature grouping, a probability-based search operator, and a penalty term, enabling it to surpass five other feature selection algorithms in performance. We evaluated ISPSO across 16 datasets, which varied in feature dimensions from 14 to 1024. The experimental results underscore ISPSO's effectiveness in eliminating redundant features through its integrated mechanisms.

Table 8
The average and standard deviation of AUC on the testing sets.

No.	ISPSO		ESAPSO		SPACO		AAPSO	
	Avg.	Std.	Avg.	Std.	Avg.	Std.	Avg.	Std.
1	**8.825E−01**	1.436E−02	7.120E−01	1.659E−02	8.740E−01	2.012E−02	7.580E−01	8.965E−02
2	8.406E−01	2.453E−02	8.091E−01	0.000E+00	8.109E−01	6.943E−03	7.892E−01	4.110E−02
3	**6.630E−01**	2.880E−02	6.065E−01	1.286E−03	6.361E−01	2.577E−02	5.820E−01	4.651E−02
4	**8.518E−01**	2.501E−02	7.798E−01	2.219E−02	8.276E−01	2.879E−02	8.413E−01	4.033E−02
5	**9.855E−01**	7.369E−03	9.804E−01	3.089E−02	9.634E−01	4.091E−02	8.931E−01	6.550E−02
6	**9.760E−01**	4.853E−03	4.002E−01	5.646E−17	9.694E−01	5.224E−03	7.613E−01	7.992E−02
7	**9.158E−01**	6.312E−03	8.421E−01	1.149E−02	7.844E−01	3.781E−02	8.087E−01	5.245E−02
8	**7.843E−01**	4.278E−02	7.360E−01	4.427E−02	7.719E−01	3.670E−02	7.272E−01	3.639E−02
9	**5.544E−01**	1.626E−02	5.275E−01	1.133E−02	5.476E−01	1.468E−02	5.502E−01	1.877E−02
10	**8.715E−01**	2.345E−02	8.694E−01	1.951E−02	8.692E−01	2.051E−02	8.406E−01	2.776E−02
11	**9.412E−01**	4.710E−03	7.923E−01	1.363E−02	9.384E−01	5.861E−03	9.251E−01	8.477E−03
12	**6.074E−01**	1.429E−02	5.938E−01	1.504E−02	5.789E−01	1.095E−02	5.902E−01	2.936E−02
13	**8.592E−01**	1.834E−02	7.812E−01	1.834E−02	7.598E−01	1.219E−02	7.211E−01	1.008E−01
14	8.933E−01	9.003E−03	8.565E−01	1.696E−02	**8.961E−01**	8.257E−03	8.690E−01	7.249E−03
15	**8.788E−01**	3.068E−02	7.715E−01	4.412E−02	8.027E−01	3.623E−02	8.472E−01	3.949E−02
16	**9.028E−01**	9.888E−03	8.840E−01	1.587E−02	8.969E−01	8.917E−03	8.910E−01	1.269E−02
p-value	–		2.412E−04		3.534E−04		2.412E−04	
No.	VS-CCPSO		CPBGSA		VLPSO		WOA	
	Avg.	Std.	Avg.	Std.	Avg.	Std.	Avg.	Std.
1	7.549E−01	5.964E−02	7.195E−01	5.576E−02	7.200E−01	5.222E−02	8.612E−01	4.450E−02
2	7.996E−01	3.567E−02	8.058E−01	4.995E−02	7.475E−01	5.092E−02	**8.441E−01**	2.694E−02
3	6.359E−01	2.629E−02	5.828E−01	3.895E−02	6.287E−01	2.864E−02	6.359E−01	1.879E−02
4	8.003E−01	2.647E−02	7.822E−01	3.156E−02	8.186E−01	3.016E−02	8.225E−01	4.113E−02
5	9.723E−01	5.877E−02	9.039E−01	6.708E−02	9.202E−01	4.371E−02	9.793E−01	7.814E−03
6	9.642E−01	2.133E−02	7.633E−01	9.260E−02	6.605E−01	6.351E−02	9.516E−01	5.546E−02
7	9.107E−01	6.488E−03	8.135E−01	5.207E−02	7.994E−01	3.589E−02	9.084E−01	8.163E−03
8	7.636E−01	5.517E−02	7.299E−01	4.072E−02	7.359E−01	5.820E−02	7.385E−01	6.032E−02
9	5.215E−01	1.874E−02	5.185E−01	1.607E−02	5.117E−01	1.621E−02	5.413E−01	2.077E−02
10	8.628E−01	3.002E−02	8.418E−01	2.378E−02	8.243E−01	2.662E−02	8.649E−01	2.068E−02
11	9.265E−01	1.292E−02	9.236E−01	1.111E−02	9.108E−01	1.300E−02	9.409E−01	3.190E−03
12	6.044E−01	3.854E−02	5.731E−01	2.310E−02	5.681E−01	1.297E−02	5.967E−01	2.025E−02
13	8.576E−01	1.081E−02	6.522E−01	5.083E−02	5.019E−01	1.713E−02	7.528E−01	2.145E−02
14	8.671E−01	3.351E−02	8.726E−01	8.824E−03	8.653E−01	6.926E−03	8.836E−01	1.286E−02
15	8.561E−01	5.187E−02	8.554E−01	1.957E−02	8.168E−01	4.308E−02	8.583E−01	2.603E−02
16	8.940E−01	2.189E−02	8.956E−01	1.086E−02	8.796E−01	1.587E−02	8.963E−01	1.214E−02
p-value	2.412E−04		2.412E−04		2.412E−04		3.534E−04	

Table 9
The classification error on the testing sets for different feature subsets.

	F_4	F_5	F_8	F_9	F_{10}	None
Error	2.000E−01	2.333E−01	2.000E−01	3.000E−01	2.000E−01	**1.667E−01**

Given ISPSO's superior performance on high-dimensional datasets relative to other algorithms, future work involves integrating ISPSO with neural networks. This integration aims to apply ISPSO to real-world high-dimensional datasets, such as those in medical text analysis, for selecting input features, thereby enhancing network performance. Nonetheless, it is crucial to recognize the inherent limitations of using a single evaluation metric for feature assessment. To counter this, we plan to investigate the application of multiple metrics to comprehensively evaluate feature significance and redundancy.

CRediT authorship contribution statement

Jinrui Gao: Writing – original draft, Software, Methodology, Conceptualization. **Ziqian Wang:** Writing – review & editing, Software, Methodology. **Ting Jin:** Writing – review & editing, Supervision. **Jiujun Cheng:** Writing – review & editing, Supervision, Conceptualization. **Zhenyu Lei:** Writing – review & editing, Supervision, Conceptualization. **Shangce Gao:** Writing – review & editing, Supervision, Software, Methodology, Conceptualization.

Declaration of competing interest

The authors certify that they have NO affiliations with or involvement in any organization or entity with any financial interest (such as honoraria; educational grants; participation in speakers' bureaus; membership, employment, consultancies, stock ownership, or other equity interest; and expert testimony or patent-licensing arrangements), or non-financial interest (such as personal or professional relationships, affiliations, knowledge or beliefs) in the subject matter or materials discussed in this manuscript.

Data availability

Data will be made available on request.

Acknowledgments

This research was partially supported by the Japan Society for the Promotion of Science (JSPS) KAKENHI under Grant JP22H03643, Japan Science and Technology Agency (JST) Support for Pioneering Research Initiated by the Next Generation (SPRING) under Grant JPMJSP2145, JST through the Establishment of University Fellowships towards the Creation of Science Technology Innovation under Grant JPMJFS2115 and National Natural Science Foundation of China Grants (No. 12201304). All authors approved the version of the manuscript to be published.

Table 10
The mean and standard deviation of the MCE across the testing sets of the variants.

No.	ISPSO		BPSO-3		BPSO-2		BPSO-1		BPSO	
	Avg.	Std.	Avg.	Std.	Avg.	Std.	Avg.	Std.	Avg.	Std.
1	**1.180E−01**	1.643E−02	1.185E−01	1.568E−02	1.510E−01	7.218E−02	1.222E−01	1.717E−02	1.237E−01	1.719E−02
2	**2.344E−01**	1.854E−02	**2.344E−01**	2.050E−02	2.378E−01	1.152E−02	2.367E−01	1.602E−02	2.378E−01	1.691E−02
3	**2.503E−01**	2.177E−02	2.557E−01	1.755E−02	2.583E−01	2.659E−02	2.588E−01	2.353E−02	2.657E−01	1.742E−02
4	1.159E−01	1.971E−02	**1.117E−01**	2.061E−02	1.127E−01	2.636E−02	1.276E−01	1.384E−02	1.317E−01	1.502E−02
5	3.272E−02	1.122E−02	3.272E−02	8.235E−03	**2.722E−02**	1.192E−02	3.333E−02	1.036E−02	3.853E−02	1.115E−02
6	2.387E−02	4.871E−03	**2.349E−02**	4.155E−03	8.921E−02	8.272E−02	2.463E−02	3.145E−03	2.829E−02	5.856E−03
7	8.164E−02	5.835E−03	8.237E−02	4.628E−03	8.483E−02	4.314E−03	**8.140E−02**	4.901E−03	8.507E−02	5.774E−03
8	**2.156E−01**	4.095E−02	2.210E−01	3.999E−02	2.360E−01	3.375E−02	2.323E−01	3.404E−02	2.323E−01	3.896E−02
9	**4.421E−01**	1.603E−02	4.467E−01	1.704E−02	4.485E−01	1.984E−02	4.682E−01	1.566E−02	4.666E−01	1.566E−02
10	**1.237E−01**	1.910E−02	1.317E−01	2.069E−02	1.397E−01	1.706E−02	1.343E−01	1.983E−02	1.336E−01	1.835E−02
11	**1.061E−01**	7.776E−03	1.077E−01	8.100E−03	1.152E−01	1.117E−02	1.152E−01	1.002E−02	1.177E−01	9.663E−03
12	3.669E−01	1.791E−02	3.701E−01	1.767E−02	**3.583E−01**	2.555E−02	3.877E−01	1.968E−02	3.916E−01	2.250E−02
13	1.408E−01	1.834E−02	2.054E−01	1.482E−02	**1.270E−01**	1.042E−02	2.315E−01	1.433E−02	2.391E−01	1.230E−02
14	**2.218E−01**	1.481E−02	2.286E−01	1.026E−02	2.421E−01	1.717E−02	2.353E−01	1.316E−02	2.440E−01	1.319E−02
15	**3.061E−01**	5.306E−02	3.170E−01	5.049E−02	3.361E−01	6.602E−02	3.313E−01	3.156E−02	3.347E−01	2.965E−02
16	**1.681E−01**	1.486E−02	1.778E−01	1.671E−02	1.856E−01	2.040E−02	1.758E−01	1.809E−02	1.753E−01	1.375E−02
p-value	–		1.160E−03		8.098E−03		2.923E−04		2.412E−04	

Table 11
The feature numbers across the testing sets of the variants.

No.	ISPSO	BPSO-3	BPSO-2	BPSO-1	BPSO
1	1.600E−01	1.644E−01	9.333E−02	1.711E−01	1.722E−01
2	5.521E−01	5.667E−01	4.792E−01	5.563E−01	5.542E−01
3	3.622E−01	3.689E−01	1.767E−01	3.633E−01	3.922E−01
4	1.588E−01	1.657E−01	1.078E−01	2.882E−01	2.961E−01
5	6.529E−01	6.627E−01	4.451E−01	5.912E−01	6.069E−01
6	5.602E−01	5.620E−01	2.380E−01	5.269E−01	5.472E−01
7	6.509E−01	5.959E−01	2.737E−01	5.269E−01	5.526E−01
8	3.178E−01	3.883E−01	2.467E−01	4.550E−01	4.889E−01
9	2.497E−01	3.110E−01	1.260E−01	4.830E−01	4.630E−01
10	4.482E−01	5.301E−01	2.992E−01	5.008E−01	4.970E−01
11	7.895E−01	7.870E−01	4.728E−01	5.493E−01	5.452E−01
12	2.700E−01	3.325E−01	7.192E−02	4.843E−01	5.008E−01
13	9.107E−02	2.828E−01	1.733E−02	4.837E−01	4.845E−01
14	3.387E−01	3.808E−01	2.558E−01	4.909E−01	4.922E−01
15	2.048E−01	2.576E−01	9.229E−02	5.014E−01	5.006E−01
16	4.267E−01	5.109E−01	2.349E−01	5.052E−01	4.984E−01

References

[1] J. Li, H. Liu, Challenges of feature selection for big data analytics, IEEE Intell. Syst. 32 (2) (2017) 9–15, http://dx.doi.org/10.1109/MIS.2017.38.

[2] J. Li, K. Cheng, S. Wang, F. Morstatter, R.P. Trevino, J. Tang, H. Liu, Feature selection: A data perspective, ACM Comput. Surv. 50 (6) (2017) 1–45, http://dx.doi.org/10.1145/3136625.

[3] J. Cai, J. Luo, S. Wang, S. Yang, Feature selection in machine learning: A new perspective, Neurocomputing 300 (2018) 70–79, http://dx.doi.org/10.1016/j.neucom.2017.11.077.

[4] I. Guyon, A. Elisseeff, An introduction to variable and feature selection, J. Mach. Learn. Res. 3 (Mar) (2003) 1157–1182.

[5] V. Kumar, S. Minz, Feature selection: a literature review, SmartCR 4 (3) (2014) 211–229.

[6] S. Gao, M. Zhou, Z. Wang, D. Sugiyama, J. Cheng, J. Wang, Y. Todo, Fully complex-valued dendritic neuron model, IEEE Trans. Neural Netw. Learn. Syst. 34 (4) (2023) 2105–2118, http://dx.doi.org/10.1109/TNNLS.2021.3105901.

[7] Y. Mingqiang, K. Kidiyo, R. Joseph, et al., A survey of shape feature extraction techniques, Pattern Recognit. 15 (7) (2008) 43–90.

[8] Z. Wang, S. Gao, M. Zhou, S. Sato, J. Cheng, J. Wang, Information-theory-based nondominated sorting ant colony optimization for multiobjective feature selection in classification, IEEE Trans. Cybern. 53 (8) (2023) 5276–5289, http://dx.doi.org/10.1109/TCYB.2022.3185554.

[9] Y. Xue, C. Zhang, F. Neri, M. Gabbouj, Y. Zhang, An external attention-based feature ranker for large-scale feature selection, Knowl.-Based Syst. 281 (2023) 111084, http://dx.doi.org/10.1016/j.knosys.2023.111084.

[10] B. Ahadzadeh, M. Abdar, F. Safara, A. Khosravi, M.B. Menhaj, P.N. Suganthan, SFE: A simple, fast, and efficient feature selection algorithm for high-dimensional data, IEEE Trans. Evol. Comput. 27 (6) (2023) 1896–1911, http://dx.doi.org/10.1109/TEVC.2023.3238420.

[11] A. Thakkar, R. Lohiya, Fusion of statistical importance for feature selection in deep neural network-based intrusion detection system, Inf. Fusion 90 (2023) 353–363, http://dx.doi.org/10.1016/j.inffus.2022.09.026.

[12] L. Abualigah, A. Diabat, Chaotic binary group search optimizer for feature selection, Expert Syst. Appl. 192 (2022) 116368, http://dx.doi.org/10.1016/j.eswa.2021.116368.

[13] R. Jiao, B.H. Nguyen, B. Xue, M. Zhang, A survey on evolutionary multiobjective feature selection in classification: approaches, applications, and challenges, IEEE Trans. Evol. Comput. (2023) http://dx.doi.org/10.1109/TEVC.2023.3292527.

[14] B.H. Nguyen, B. Xue, M. Zhang, A survey on swarm intelligence approaches to feature selection in data mining, Swarm Evol. Comput. 54 (2020) 100663, http://dx.doi.org/10.1016/j.swevo.2020.100663.

[15] F. Kılıç, Y. Kaya, S. Yildirim, A novel multi population based particle swarm optimization for feature selection, Knowl.-Based Syst. 219 (2021) 106894, http://dx.doi.org/10.1016/j.knosys.2021.106894.

[16] Y. Zhu, W. Li, T. Li, A hybrid artificial immune optimization for high-dimensional feature selection, Knowl.-Based Syst. 260 (2023) 110111, http://dx.doi.org/10.1016/j.knosys.2022.110111.

[17] X.-f. Song, Y. Zhang, D.-w. Gong, X.-y. Sun, Feature selection using bare-bones particle swarm optimization with mutual information, Pattern Recognit. 112 (2021) 107804, http://dx.doi.org/10.1016/j.patcog.2020.107804.

[18] X.-F. Song, Y. Zhang, D.-W. Gong, X.-Z. Gao, A fast hybrid feature selection based on correlation-guided clustering and particle swarm optimization for high-dimensional data, IEEE Trans. Cybern. 52 (9) (2021) 9573–9586, http://dx.doi.org/10.1109/TCYB.2021.3061152.

[19] L. Fang, X. Liang, A novel method based on nonlinear binary grasshopper whale optimization algorithm for feature selection, J. Bionic Eng. 20 (1) (2023) 237–252, http://dx.doi.org/10.1007/s42235-022-00253-6.

[20] L. Ma, N. Li, G. Yu, X. Geng, S. Cheng, X. Wang, M. Huang, Y. Jin, Pareto-wise ranking classifier for multi-objective evolutionary neural architecture search, IEEE Trans. Evol. Comput. (2023) 1, http://dx.doi.org/10.1109/TEVC.2023.3314766.

[21] Z. Lei, S. Gao, Z. Zhang, M. Zhou, J. Cheng, MO4: A many-objective evolutionary algorithm for protein structure prediction, IEEE Trans. Evol. Comput. 26 (3) (2021) 417–430, http://dx.doi.org/10.1109/TEVC.2021.3095481.

[22] H. Xu, B. Xue, M. Zhang, A duplication analysis-based evolutionary algorithm for biobjective feature selection, IEEE Trans. Evol. Comput. 25 (2) (2020) 205–218, http://dx.doi.org/10.1109/TEVC.2020.3016049.

[23] E. Hancer, B. Xue, M. Zhang, Fuzzy filter cost-sensitive feature selection with differential evolution, Knowl.-Based Syst. 241 (2022) 108259, http://dx.doi.org/10.1016/j.knosys.2022.108259.

[24] Z. Wang, S. Gao, Y. Zhang, L. Guo, Symmetric uncertainty-incorporated probabilistic sequence-based ant colony optimization for feature selection in classification, Knowl.-Based Syst. 256 (2022) 109874, http://dx.doi.org/10.1016/j.knosys.2022.109874.

[25] F. Karimi, M.B. Dowlatshahi, A. Hashemi, SemiACO: A semi-supervised feature selection based on ant colony optimization, Expert Syst. Appl. 214 (2023) 119130, http://dx.doi.org/10.1016/j.eswa.2022.119130.

[26] A.M. Vommi, T.K. Battula, A hybrid filter-wrapper feature selection using fuzzy KNN based on Bonferroni mean for medical datasets classification: A COVID-19 case study, Expert Syst. Appl. 218 (2023) 119612, http://dx.doi.org/10.1016/j.eswa.2023.119612.

[27] L. Qu, W. He, J. Li, H. Zhang, C. Yang, B. Xie, Explicit and size-adaptive PSO-based feature selection for classification, Swarm Evol. Comput. 77 (2023) 101249, http://dx.doi.org/10.1016/j.swevo.2023.101249.

[28] B. Tran, B. Xue, M. Zhang, A new representation in PSO for discretization-based feature selection, IEEE Trans. Cybern. 48 (6) (2017) 1733–1746, http://dx.doi.org/10.1109/TCYB.2017.2714145.

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

[30] S. Fong, R. Wong, A.V. Vasilakos, Accelerated PSO swarm search feature selection for data stream mining big data, IEEE Trans. Serv. Comput. 9 (1) (2015) 33–45, http://dx.doi.org/10.1109/TSC.2015.2439695.

[31] B. Tran, B. Xue, M. Zhang, Variable-length particle swarm optimization for feature selection on high-dimensional classification, IEEE Trans. Evol. Comput. 23 (3) (2018) 473–487, http://dx.doi.org/10.1109/TEVC.2018.2869405.

[32] X.-F. Song, Y. Zhang, Y.-N. Guo, X.-Y. Sun, Y.-L. Wang, Variable-size cooperative coevolutionary particle swarm optimization for feature selection on high-dimensional data, IEEE Trans. Evol. Comput. 24 (5) (2020) 882–895, http://dx.doi.org/10.1109/TEVC.2020.2968743.

[33] A. Gutiérrez, M. Lanza, I. Barriuso, L. Valle, M. Domingo, J. Perez, J. Basterrechea, Comparison of different pso initialization techniques for high dimensional search space problems: A test with fss and antenna arrays, in: Proceedings of the 5th European Conference on Antennas and Propagation (EUCAP), IEEE, 2011, pp. 965–969.

[34] X. Wu, J. Han, D. Wang, P. Gao, Q. Cui, L. Chen, Y. Liang, H. Huang, H.P. Lee, C. Miao, et al., Incorporating surprisingly popular algorithm and euclidean distance-based adaptive topology into PSO, Swarm Evol. Comput. 76 (2023) 101222, http://dx.doi.org/10.1016/j.swevo.2022.101222.

[35] J. Dong, Q. Li, L. Deng, Design of fragment-type antenna structure using an improved BPSO, IEEE Trans. Antennas and Propagation 66 (2) (2017) 564–571, http://dx.doi.org/10.1109/TAP.2017.2778763.

[36] Q.-S. Zhang, S.-Y. Jiang, A note on information entropy measures for vague sets and its applications, Inform. Sci. 178 (21) (2008) 4184–4191, http://dx.doi.org/10.1016/j.ins.2008.07.003.

[37] Q. Hu, M. Guo, D. Yu, J. Liu, Information entropy for ordinal classification, Sci. China Inf. Sci. 53 (2010) 1188–1200, http://dx.doi.org/10.1007/s11432-010-3117-7.

[38] C. Shang, M. Li, S. Feng, Q. Jiang, J. Fan, Feature selection via maximizing global information gain for text classification, Knowl.-Based Syst. 54 (2013) 298–309, http://dx.doi.org/10.1016/j.knosys.2013.09.019.

[39] J.R. Quinlan, C4. 5: Programs for Machine Learning, Elsevier, 2014.

[40] J. Dai, Q. Xu, Attribute selection based on information gain ratio in fuzzy rough set theory with application to tumor classification, Appl. Soft Comput. 13 (1) (2013) 211–221, http://dx.doi.org/10.1016/j.asoc.2012.07.029.

[41] E. Zorarpacı, S.A. Özel, A hybrid approach of differential evolution and artificial bee colony for feature selection, Expert Syst. Appl. 62 (2016) 91–103, http://dx.doi.org/10.1016/j.eswa.2016.06.004.

[42] Y. Zhang, D.-w. Gong, X.-z. Gao, T. Tian, X.-y. Sun, Binary differential evolution with self-learning for multi-objective feature selection, Inform. Sci. 507 (2020) 67–85, http://dx.doi.org/10.1016/j.ins.2019.08.040.

[43] T. Bhadra, S. Bandyopadhyay, Unsupervised feature selection using an improved version of differential evolution, Expert Syst. Appl. 42 (8) (2015) 4042–4053, http://dx.doi.org/10.1016/j.eswa.2014.12.010.

[44] R. Pramanik, S. Sarkar, R. Sarkar, An adaptive and altruistic PSO-based deep feature selection method for pneumonia detection from chest X-rays, Appl. Soft Comput. 128 (2022) 109464, http://dx.doi.org/10.1016/j.asoc.2022.109464.

[45] R. Guha, M. Ghosh, A. Chakrabarti, R. Sarkar, S. Mirjalili, Introducing clustering based population in binary gravitational search algorithm for feature selection, Appl. Soft Comput. 93 (2020) 106341, http://dx.doi.org/10.1016/j.asoc.2020.106341.

[46] M. Mafarja, S. Mirjalili, Whale optimization approaches for wrapper feature selection, Appl. Soft Comput. 62 (2018) 441–453, http://dx.doi.org/10.1016/j.asoc.2017.11.006.

[47] E. Hancer, New filter approaches for feature selection using differential evolution and fuzzy rough set theory, Neural Comput. Appl. 32 (2020) 2929–2944, http://dx.doi.org/10.1007/s00521-020-04744-7.

[48] P. Wang, B. Xue, J. Liang, M. Zhang, Feature selection using diversity-based multi-objective binary differential evolution, Inform. Sci. 626 (2023) 586–606, http://dx.doi.org/10.1016/j.ins.2022.12.117.

[49] B. Rosner, R.J. Glynn, M.-L.T. Lee, The wilcoxon signed rank test for paired comparisons of clustered data, Biometrics 62 (1) (2006) 185–192.

[50] R.S.M. de Barros, J.I.G. Hidalgo, D.R. de Lima Cabral, Wilcoxon rank sum test drift detector, Neurocomputing 275 (2018) 1954–1963, http://dx.doi.org/10.1016/j.neucom.2017.10.051.

[51] P. Wang, B. Xue, J. Liang, M. Zhang, Differential evolution-based feature selection: A niching-based multiobjective approach, IEEE Trans. Evol. Comput. 27 (2) (2022) 296–310, http://dx.doi.org/10.1109/TEVC.2022.3168052.

[52] R. Forsyth, Zoo, UCI machine learning repository, 1990, http://dx.doi.org/10.24432/C5R59V.