IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 28 May 2024, accepted 28 July 2024, date of publication 2 August 2024, date of current version 12 August 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3437738

RESEARCH ARTICLE

A Hybrid Feature Selection Algorithm Based on Collision Principle and Adaptability

XIAOTONG BAI, YUEFENG ZHENG, AND YANG LU
School of Mathematics and Computer Science, Jilin Normal University, Siping 136000, China

Corresponding author: Yuefeng Zheng (honest_zyf@hotmail.com)

This work was supported by the Natural Science Foundation of Jilin Province under Grant 20210101176JC and Grant YDZJ202301ZYTS157.

ABSTRACT Feature selection plays a significant role in machine learning and data mining, where the goal is to screen out the most representative and relevant subset of features from a large collection of features to improve the performance and generalization ability of the model. In this paper, a hybrid feature selection algorithm that combines a filter algorithm and an improved particle swarm optimization algorithm is proposed, that is, the Information Gain and Maximum Pearson Minimum Mutual Information improved Adaptive Particle Swarm Optimization algorithm (IGMPMMIAPSO). First, combined with the characteristics of the Pearson correlation coefficient and mutual information, a filter algorithm called Maximum Pearson Minimum Mutual Information (MPMMI) is proposed. The algorithm balances the relevance and redundancy between the features by adjusting two weight parameters (w_{p1} and w_{p2}). Second, Adaptive Adjustment of Control (AAC) is introduced to update the particle swarm optimization algorithm, so that the particle velocity has a higher searching ability, and the diversity of population position changes is increased. The improved algorithm was used as the wrapper algorithm. Simultaneously, the concepts of the No Continuous Change (NCC) times and collision distance values are proposed. According to these, the IGMPMMIAPSO algorithm is proposed by combining the filter algorithm and wrapper algorithm. To verify the performance of the proposed algorithm, we experimented with other state-of-the-art hybrid algorithms using eight datasets. The experimental results show that the classification accuracy of the proposed algorithm is at least 0.1% higher than that of the other five algorithms, and the feature subset length is shorter.

INDEX TERMS Feature selection, filter, collision distance value, MPMMI, APSO.

I. INTRODUCTION

Feature selection(FS) [1], [2] is a critical step in machine learning and data analysis [3], [4], because it helps to identify the most informative and relevant features from a given dataset, while removing irrelevant or redundant features. The selected features not only improve the accuracy and efficiency of the model, but also provide insight into potential patterns and relationships in the data.

In recent years, additional feature selection algorithms have been proposed and applied to solve dimension disasters [5], [6], [7] problems in machine learning. Among them, the filter algorithms and wrapper algorithms are common feature selection methods. Filter methods (e.g., Maximum Relevance and Minimum Redundancy (mRMR) [8], Information Gain (IG) [9]) evaluate and rank the features to select the most relevant features independently of the specific machine learning algorithm. Wrapper methods combine meta-heuristic algorithms (e.g., Genetic Algorithm (GA) [10], Whale Optimization Algorithm (WOA) [11], Particle Swarm Optimization algorithm (PSO) [12]) with classifiers [13] (e.g., Support Vector Machine (SVM) [14], k-Nearest Neighbor (KNN) [15]), where the performance of the model is trained and evaluated using different subsets of features, and selecting the subset of features with the best performance. These feature selection methods can help to reduce the number of features in dataset and improve the model's efficiency. However, both the filter and wrapper algorithms have limitations. Filter algorithms may ignore the interactions and dependencies between features because they evaluate

The associate editor coordinating the review of this manuscript and approving it for publication was Yongqiang Cheng.

the features independently. However, wrapper algorithms are computationally extremely expensive and time consuming, especially for datasets with a large number of features.

Inspired by MFO, Chen et al. [16] proposed a new FS method based on PSO to solve the high-dimensional classification problem by sharing information between two related tasks generated from a dataset. This method combines feature selection and classification tasks by optimizing multiple classification tasks using an evolutionary algorithm to select the best subset of features. However, this method only considers sharing the global best solution in PSO, and does not consider other useful information such as local position and velocity. This may result in high computational complexity and require a lot of computing resources. Yang et al. [17] proposed a bidirectional feature-fixation (BDFF) framework based on PSO for large-scale feature-selection problems. Aiming at large-scale datasets, a bidirectional feature-fixation mechanism was introduced to make the algorithm more effective in feature selection. This method can effectively deal with high-dimensional feature spaces and reduce the computational complexity of the feature selection process. However, the evaluation time was long and might have been influenced by the local optimal solutions in some cases. Thaher et al. [18] proposed a feature selection method that combines the Boolean Particle Swarm Optimization (BPSO) with multiple evolutionary population dynamics methods. By utilizing the global search capability of the BPSO and the advantages of the evolutionary population dynamics method, the efficiency and accuracy of feature selection were improved. However, feature selection combining multiple optimization methods may require more computational resources, especially when dealing with large-scale datasets, which may lead to the problem of higher computational complexity.

Kaur et al. [19] proposed a feature selection method based on mutual information and adaptive Particle Swarm Optimization (PSO) for image steganalysis. Mutual information is used as a metric for feature selection to capture the relevance between features and target variables, which helps in selecting the most relevant features for image steganalysis tasks. The utilization of the adaptive PSO algorithm [12] for feature selection enhances the global search capability, thereby facilitating the identification of an optimal subset of features. Ye et al. [20] proposed a feature selection method using a Leader-Learning Adaptive Particle Swarm Optimization algorithm (LLAPSO). The LLAPSO method adopts the leader learning mechanism, making the particles based on the experience of leaders adjust their learning strategies, thus enhancing the global search ability of the algorithm. In addition, an adaptive weight factor and adaptive inertia weight were introduced to improve the adaptability and convergence speed of the algorithm, but they may fall into a local optimal solution in a high-dimensional feature space. Hu et al. [21] proposed a federal feature selection algorithm based on particle swarm optimization. The algorithm is performed under privacy protection, which is particularly important for scenarios involving sensitive data. Adopting the framework of federated learning for feature selection would enable model training and feature selection among multiple participants without sharing the original data, which is beneficial for protecting data privacy. However, it should be noted that privacy protection might have an impact on the performance and efficiency of the algorithm, and it is necessary to weigh the performance of privacy protection and feature selection. The parameter selection and convergence of the PSO algorithm may affect the performance of the algorithm, which must be appropriately tuned.

In summary, hybrid feature selection algorithms were analyzed and demonstrated based on an improved wrapper algorithm. In this paper, a hybrid feature selection algorithm was developed by combining a filter algorithm with a wrapper algorithm. To ensure that the filter algorithm provides a better subset of candidate features, the wrapper algorithm can better jump out of the local optimum and provide a better classification accuracy. In this paper, a new hybrid feature selection algorithm [22] was developed by combining the filter algorithm with the wrapper algorithm. The experimental results show that the proposed algorithm outperforms the traditional filter algorithms and wrapper algorithms in terms of feature selection accuracy and computational efficiency. Moreover, this set of experiments achieved better classification results than those of the compared algorithms on the eight datasets.

The contributions of this paper are as follows:

1) A filter algorithm based on the Pearson correlation coefficient and mutual information is proposed, which is called the Maximum Pearson Minimum Mutual Information (MPMMI). The weights of the relevance and redundancy were adjusted in the algorithm using two weighting parameters (w_{p1} and w_{p2}).

2) Modifying the particle swarm optimization algorithm. Adaptive Adjustment of Control (AAC) is introduced to update the velocity of the particle, and the position of the particle is updated by comparing the average probability (p) of the random number of all dimensions with the probability of any dimension. In addition, for the corresponding positions of the remaining features after adjustment by the PSO algorithm, the value that appears the most times in each iteration (the mode) is adopted for normalization.

3) No continuous change times or the concept of collision distance. The filter algorithm is invoked multiple times according to the relationship between the ball collision distance and No Consecutive Changes (NCC). The filter algorithm (i.e., univariate and multivariate filter algorithms) is determined by the number of collision-distance values (count_distance) generated during the iteration. Thus, the subset of candidate features provided by the filter algorithms was adjusted to facilitate the algorithms to jump out of the local optimum.

The structure of this paper is as follows. The second part introduces the related work of the algorithm. The third part describes the proposed hybrid feature selection algorithm in detail. The fourth part introduces the experimental setup and

the analysis of the experimental results. The fifth section discusses the advantages and disadvantages of the algorithm, as well as the direction of future research, and provides a further outlook.

II. RELATED WORKS

This part mainly introduces the relevant contents involved in the algorithm, including the principle of ball collision and collision types (Section A), the basic concept and calculation method of information gain (Section B), the concept of the Pearson correlation coefficient and relevant calculation method (Section C), and the calculation of mutual information (Section D). And particle swarm optimization algorithm is related to theoretical knowledge, formulas, and other content introduction (Section E).

A. BALL COLLISION(BC)

The principle of ball collision is based on Newton's third law [23], which states that the forces between any two bodies are reciprocal, equal in magnitude, and opposite in direction. Ball collision refers to the interaction between two or more small balls through the action of each other's forces colliding to change their speed and direction. According to the type of collision, ball collisions can be classified as elastic [24] and inelastic [25].

1) ELASTIC COLLISION

In elastic collisions, the interaction force between the balls causes their velocities to change; however, the total kinetic energy and momentum [26] remain constant before and after the collision, that is, the total energy and momentum are conserved. If the shape of the objects does not change during the collision, then it is a perfectly elastic collision.

Assuming that the masses of balls A and B are m_A and m_B, respectively; v_A and v_B are the initial velocities of balls A and B before the collision, respectively; and v_{Af} and v_{Bf} are the velocities of balls A and B after the collision, respectively; the energy conservation law can be expressed as follows:

$$\frac{1}{2}m_A \cdot v_A^2 + \frac{1}{2}m_B v_B^2 = \frac{1}{2}m_A v_{Af}^2 + \frac{1}{2}m_B v_{Bf}^2 \tag{1}$$

According to the momentum conservation law before and after the collision, we obtain.

$$m_A v_A + m_B v_B = m_A v_{Af} + m_B v_{Bf} \tag{2}$$

From the above two conservation formulas, which are perfectly elastic collisions, kinetic energy is conserved, and kinetic energy remains unchanged before and after the collision.

$$v_{Af} = \frac{m_A - m_B}{m_A + m_B} \cdot v_A + \frac{2m_B}{m_A + m_B} \cdot v_B \tag{3}$$

$$v_{Bf} = \frac{2m_A}{m_A + m_B} \cdot v_A - \frac{m_A - m_B}{m_A + m_B} \cdot v_B \tag{4}$$

2) INELASTIC COLLISION

Inelastic collision refers to the process of collision between some or all kinetic energy into other forms of energy, such as heat energy, sound energy, and deformation energy. In inelastic collisions, momentum is conserved, but the kinetic energy is no longer completely conserved.

B. INFORMATION GAIN(IG)

Information Gain [27] is a metric used in the field of machine learning and data mining to quantify the amount of information provided by specific features in the context of a given dataset. The calculation of the information gain [28] is based on the concept of information entropy. Information entropy is used to measure the uncertainty or purity of data. Information gain was obtained by comparing the parent node of information entropy and information entropy to measure the characteristics of the child node's contribution to the classification task.

In information theory, information entropy is calculated as follows:

$$\boldsymbol{H}(\boldsymbol{T}) = -\sum \left(\boldsymbol{p}(\boldsymbol{x}) \cdot \boldsymbol{log}_2 \boldsymbol{p}(\boldsymbol{x})\right) \tag{5}$$

where H(T) represents the entropy (information uncertainty) of the target variable T, and p(x) represents the probability of each class in the target variable T.

For the conditional entropy H (T|A) under the condition of the characteristics of a target variable T condition entropy, the computation formula is as follows:

$$\boldsymbol{H}(\boldsymbol{T}\,|\boldsymbol{A}) = \sum (\boldsymbol{p}(\boldsymbol{a}) \cdot \boldsymbol{H}(\boldsymbol{T}\,|\boldsymbol{A} = \boldsymbol{a})) \tag{6}$$

where p(a) represents the probability of each value in feature A, and H(T|A=a) represents the entropy of the target variable T under the condition that feature A is a.

$$\boldsymbol{H}(\boldsymbol{T}\,|\boldsymbol{A} = \boldsymbol{a}) = -\sum \boldsymbol{p}(\boldsymbol{x}\,|\boldsymbol{A} = \boldsymbol{a}) \cdot \boldsymbol{log}_2 \boldsymbol{p}(\boldsymbol{x}\,|\boldsymbol{A} = \boldsymbol{a}) \tag{7}$$

where $p\ (x\ |A{=}a)$ represents the probability of each category in target variable T under the condition that feature A is a.

Algorithm 1 Pseudo-Code of the IG

Input: Dataset D, sample number n (each sample has m features and a label), number of features: K
Output: Sequence of features
Calculate IG
  Calculate the entropy H(D) of the dataset according to formula (5).
For each feature i:
  Calculate the entropy of feature H (D |i) according to formula (7).
  Calculate the IG of feature i according to formula (8).
Selected_Features is the top K features
Return Selected_Features
End

By calculating the information entropy and conditional entropy, it is possible to measure the information gain (IG) obtained by categorizing the target variable T under the condition of feature A. IG can be expressed as follows:

$$\boldsymbol{IG}(\boldsymbol{T}, \boldsymbol{A}) = \boldsymbol{H}(\boldsymbol{T}) - \boldsymbol{H}(\boldsymbol{T}\,|\boldsymbol{A}) \tag{8}$$

C. PEARSON CORRELATION COFFFICIENT(PEARSON)

Pearson's correlation coefficient [29] (Irene Rodriguze-Lujan et al.,2010), Pearson product-moment correlation coefficient (PPMCC or PCCs), also known as PPMCC or PCCS, is a method to measure the correlation between two variables [30]. It reflects the correlation of two variables; therefore, it is only sensitive to a linear relationship. The values range between [−1, 1], with one indicating a perfectly positive correlation, zero indicating no linear relationship, and −1 indicating a perfectly negative correlation. If there are variables X ($x_1, x_2, \ldots, x_n$) and Y ($y_1, y_2, \ldots, y_n$), then the Pearson correlation coefficient of variables X and Y can be expressed as

$$\rho(X, Y) = \frac{cov(X, Y)}{\sigma X \cdot \sigma Y} \tag{9}$$

where cov (X, Y) is the covariance of variables X and Y. σX and σY denote the standard deviations of variables X and Y, respectively.

Covariance represents the overall error between the two variables. The standard Deviation represents the degree of dispersion of the variables. Covariance and standard deviation were calculated using the following formulas:

$$cov(X, Y) = \frac{\sum_{i=1}^{N}(X - \mu X) \cdot (Y - \mu Y)}{N} \tag{10}$$

$$\sigma X = \sqrt{\frac{\sum_{i=1}^{N}(X - \mu X)}{N}} \tag{11}$$

$$\sigma Y = \sqrt{\frac{\sum_{i=1}^{N}(Y - \mu Y)}{N}} \tag{12}$$

where X and Y are the values of the variables X and Y, respectively. μX and μY are the mean values of variables X and Y, respectively, and N is the number of samples.

D. INFORMATION(MI)

Mutual Information (MI) [37], [38] is an indicator used to measure the correlation between two random variables. It can be used to measure the degree of interdependence between two variables, and the amount of information passed between them. It may be regarded as a random variable that contains information about another random variable, or a random variable with another known random variable that is not positive.

The calculation of mutual information is based on the concept of information entropy, which can be calculated by using the joint probability distribution of two random variables and their respective edge probability distributions. The higher the value of mutual information, the stronger the correlation between the two variables and the weaker the correlation between the two variables. The joint distribution of two random variables (X, Y) is denoted as p (x, y), and the marginal distribution is denoted as p(x) and p(y) respectively. Mutual information I (X: Y) is the relative entropy of the joint distribution p (x, y) and the marginal distribution p(x)p(y), that is, the mutual information of two discrete random variables X and Y can be defined as follows:

$$I(X, Y) = \sum_{x \in X}\sum_{y \in Y} p(x, y) \cdot log\frac{p(x, y)}{p(x)p(y)} \tag{13}$$

According to the chain rule of entropy, we have:

$$H(X) - H(X \mid Y) = H(Y) - H(Y \mid X) \tag{14}$$

The difference is called X and Y of mutual information, as I (X; Y).

Mutual information can be both positive and negative, depending on the relationship between random variables. Mutual information represents the degree of information dependence between two random variables, and its positive and negative values and magnitudes reflect the diverse types of dependence.

E. PARTICLE SWARM OPTIMIZATION (PSO)

PSO is an evolutionary computation technique developed in 1995 by Eberhart and Kennedy [31], from the study of bird predation behavior. The algorithm is a simplified model based on swarm intelligence inspired by the regularity of bird swarm activity. PSO is based on the observation of animal cluster activities and makes use of the information shared by individuals in the group to make the movement of the whole group evolve from disorder to order in the problem-solving space to obtain the optimal solution.

The basic idea of the PSO algorithm is to regard the problem to be optimized as a search problem in multidimensional space and regard each solution as a particle, which searches for the optimal solution by constantly adjusting its position and speed. A flowchart of the PSO algorithm is shown in Fig. 1.

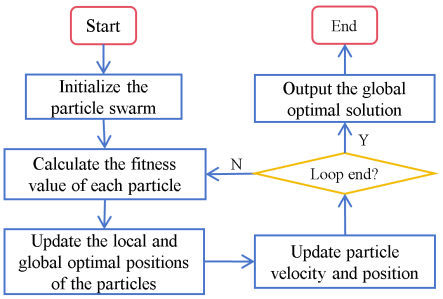

FIGURE 1. Flowchart of the PSO algorithm.

Each particle has its own position and velocity, which needs to be updated based on own optimum and the population optimum.

$$v_{t+1} = w \cdot v_t + c_1 \cdot rand() \cdot (pBest_t - x_t) + c_2 \cdot rand() \cdot (gBest - x_t) \tag{15}$$

$$x_{t+1} = x_t + v_{t+1} \tag{16}$$

where v_{t+1} is the velocity of the particle at t+1 iterations, w is the inertia weight, c_1 and c_2 are the learning factors, rand() is a random number between 0 and 1, pBest and gBest are the local and global best positions of the particle, respectively,

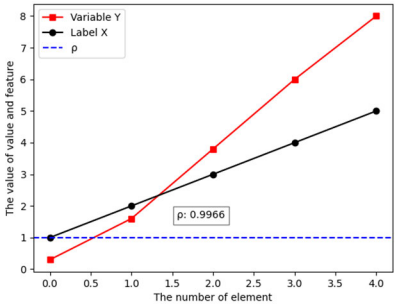

FIGURE 2. Relationship between label *X* and variable *Y*.

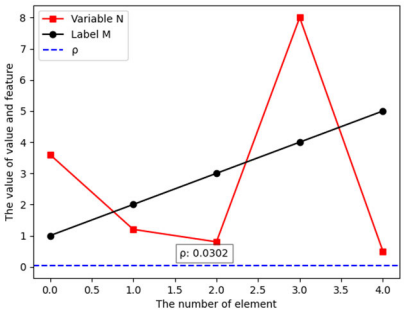

FIGURE 3. Relationship between label *M* and variable *N*.

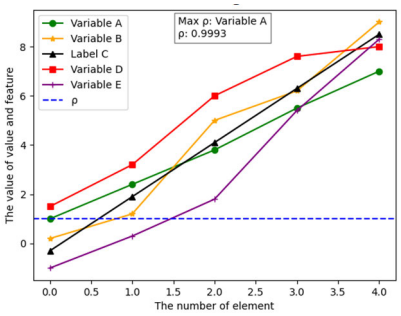

FIGURE 4. Relationship between multiple variables and labels.

and x_t, x_{t+1} are the positions of the particle at t and $t+1$ iterations, respectively.

The advantages of the PSO algorithm include simple implementation, strong global search ability, and good adaptability to the constraints of the problem [32].

A hybrid feature selection algorithm IGMPMMIAPSO, is proposed in this paper to avoid the filter algorithm providing a single subset of candidate features in feature selection, which causes the algorithm to fall into a local optimal solution. The algorithm is an improvement of related work and consists of IG, MPMMI, and the Adaptive Particle Swarm Optimization algorithm (APSO). In the third part of the article, the design of the algorithm is described in detail.

III. IGMPMMIAPSO ALGORITHM

This section introduces the framework, pseudo-code, and algorithm flowchart of MPMMI, APSO, and the proposed algorithm IGMPMMIAPSO.

A. MPMMI FILTER ALGORITHM

A new filter algorithm, MPMMI, is proposed with two parameters to adjust the relevance between labels and features using the Pearson correlation coefficient and the redundancy between features using mutual information.

1) MAXIMUM PEARSON (MP)

Through the introduction of the Pearson correlation coefficient in Part II C, the Pearson correlation coefficients of the two variables can be obtained by calculating the covariance and standard deviation.

Suppose we give the data of two variables x and y, x = [1, 2, 3, 4, 5], y = [0.3, 1.6, 3.8, 6, 8], and calculate ρ(x, y) = 0.9966 using formula (9). 0.9966 is very close to one, indicating that there is a strong positive correlation between variables x and y. As shown in Fig. 2.

If two variables are given as m = [1, 2, 3, 4, 5] and n = [3.6, 1.2, 0.8, 8, 0.5], ρ (m, n) =0.0302 is calculated using formula (9), which is close to 0, indicating that there is no linear relationship between variables m and n.

Figs. 2 and 3 show that the relationship between variables x and y is stronger, and the correlation between variables m and n is weak.

There are multiple features in the dataset, and the feature with the strongest correlation to the label can using calculated by the following formula:

$$\boldsymbol{MP}\left(\boldsymbol{F_{object}}, \boldsymbol{L}\right) = \boldsymbol{max}\left(\boldsymbol{\rho}\left(\boldsymbol{F_i}, \boldsymbol{L}\right)\right) \tag{17}$$

where MP(F_{object}, L) represents the maximum Pearson value from the feature set to the label, F_{object} represents the set of all features in the dataset, L represents the label vector in the dataset, max represents the maximum value function, ρ(F_i, L) is calculated according to formula (9), represents the Pearson correlation coefficient between variables F_i and L, F_i represents the ith feature in the set, $i = 1, 2, \ldots, n$.

If there are multiple features, such as five variables, then, A = [1, 2.4, 3.8, 5.5, 7], B = [0.2, 1.2, 5, 6.2, 9], C = [0.3, 1.9, 4.1, 6.3, 8.5], D = [1.5, 3.2, 6, 7.6, 8], E = [1, 0.3, 1.8, 5.4, 8.3]. The Pearson values between the four variables A, B, D, E, and the labeled variable C were calculated, and the maximum value was selected as the maximum Pearson value [33], which was calculated using formula (9) to obtain max(ρ) = 0.9993, as shown in Fig. 4.

2) MINIMUM MUTUAL INFORMATION (MMI)

Based on the concepts of maximum relevance and minimum redundancy, we need to consider not only the relevance between features and labels, but also the redundancy between features. Mutual information was used in this paper to evaluate the redundancy between features. This provides a comprehensive measure of the relationship between features, which leads to better trade-offs and judgments in the feature selection process.

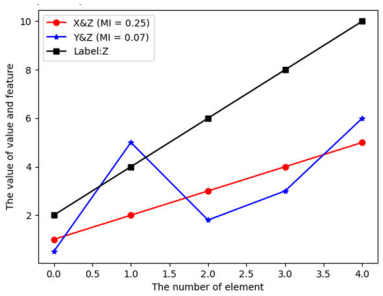

FIGURE 5. Relationship between variables X, Y, and label variable Z.

Similar to the correlation coefficient, if the two existing features are denoted as F_i and F_j respectively, the correlation coefficient is denoted as $\rho(\mathrm{F_i}, \mathrm{F_j})$, and the mutual information is denoted as $\mathrm{I}(\mathrm{F_i}, \mathrm{F_j})$.

However, both Pearson and mutual information denote relevance, so we take the value of minimum mutual information [34] as a measure of redundancy between features. For example, variables $X = [1, 2, 3, 4, 5]$, $Y = [0.5, 5, 1.8, 3, 6]$, $Z = [2, 4, 6, 8, 10]$, calculated by formula (13), $\mathrm{I}(\mathrm{X}, \mathrm{Z}) = 0.25$, $\mathrm{I}(\mathrm{Y}, \mathrm{Z}) = 0.07$, $\min(I) = 0.07$, as shown in Fig. 5.

In the feature selection process, it is necessary to select feature from a group, and the known feature is the redundancy of the smallest feature. The mutual information value between a feature and each feature in the feature group is calculated using the mutual information, and the feature with the minimum mutual information is selected.

A known feature is labeled F_{know}, and a set of features is labeled $F_{object} = \{F_1, F_2, \ldots, F_k, \ldots, F_n\}$, $k = 1, 2, \ldots, n$, then the minimum mutual information can be expressed as follows:

$$\boldsymbol{MMI}\left(\boldsymbol{F_{know}}, \boldsymbol{F_{object}}\right) = \boldsymbol{min}\left(\boldsymbol{MI}\left(\boldsymbol{F_{know}}, \boldsymbol{F_k}\right)\right) \tag{18}$$

where $MMI(F_{know}, F_{object})$ represents the value of the minimum mutual information from a feature to a set of features corresponding to a feature F_k, min represents the function of finding the minimum value, and $\mathrm{MI}(\mathrm{F_{know}}, \mathrm{F_k})$ is calculated using formula (13).

The mutual information value between two sets of features must also be calculated using the mutual information value between two variables. The set of known features is $G = \{g_1, g_2, \ldots, g_i, \ldots, g_m\}$, $i = 1, 2, \ldots, m$. Another goal set for $F_{object} = \{F_1, F_2, \ldots, F_k, \ldots F_n\}$, $k = 1, 2, \ldots, n$. Feature is selected from the target set such that the mutual information between this feature and the known feature set is minimum.

$$\boldsymbol{MMI}\left(\boldsymbol{G}, \boldsymbol{F_{object}}\right) = \boldsymbol{min}\left(\boldsymbol{I}\left(\boldsymbol{G}, \boldsymbol{F_k}\right)\right) \boldsymbol{k} = \boldsymbol{1}, \boldsymbol{2}, \ldots, \boldsymbol{n}. \tag{19}$$

where $\mathrm{MMI(G,F_{object})}$ represents the minimum mutual information between the target set and the known set corresponding to a feature F_k, min represents the minimization function, G represents the known feature set, F_{object} represents the target feature set, M is calculated according to formula (20), and F_k is the element in the set.

$$\boldsymbol{MI}\left(\boldsymbol{G}, \boldsymbol{F_k}\right) = \frac{\sum_{i=1}^{m} \boldsymbol{I}\left(\boldsymbol{g_i}, \boldsymbol{F_k}\right)}{\boldsymbol{m}} \tag{20}$$

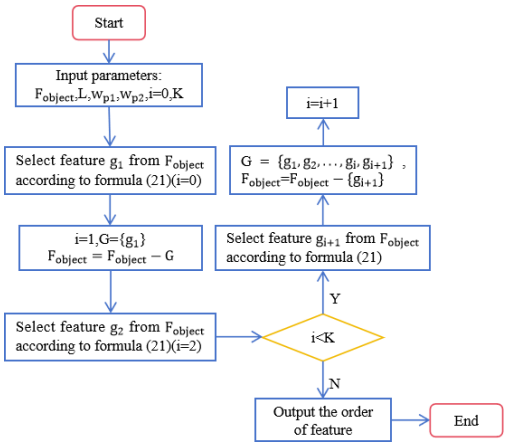

FIGURE 6. Flowchart of the MPMMI algorithm.

where m represents the number of elements in the feature set G, and g_i is the element in G. $\mathrm{I}(\mathrm{g_i}, \mathrm{F_k})$ is calculated by formula (13), which represents the mutual information of variables g_i and F_k, g_i and F_k represent the i and kth features in the set, respectively, $i = 1, 2, \ldots, m$, $k = 1, 2, \ldots, n$.

3) MAXIMUM PEARSON MINIMUM MUTUAL INFORMATION (MPMMI)

To achieve the "maximum relevance and minimum redundancy" criterion, we combined the maximum Pearson's correlation coefficient and the minimum mutual information mentioned earlier. We adjust the weight of both by introducing weighting factors in formula (21) to avoid the situation where both are equally important in the filter algorithm. From this, we can calculate the score of each feature and output the sequence of features in descending order.

$$\boldsymbol{MPMMI}\left(\boldsymbol{g_i}\right) = \begin{cases} \boldsymbol{w_{p1}} \cdot \boldsymbol{MP}\left(\boldsymbol{F_{object}}, \boldsymbol{L}\right) \boldsymbol{i} = \boldsymbol{0} \\ \boldsymbol{w_{p1}} \cdot \boldsymbol{MP}\left(\boldsymbol{F_{object}}, \boldsymbol{L}\right) + \boldsymbol{w_{p2}} \cdot \boldsymbol{MMI}\left(\boldsymbol{g_1}, \boldsymbol{F_{object}}\right) \boldsymbol{i} = \boldsymbol{1} \\ \boldsymbol{w_{p1}} \cdot \boldsymbol{MP}\left(\boldsymbol{F_{object}}, \boldsymbol{L}\right) + \boldsymbol{w_{p2}} \cdot \boldsymbol{MMI}\left(\boldsymbol{G}, \boldsymbol{F_{object}}\right) \boldsymbol{i} > \boldsymbol{1} \end{cases} \tag{21}$$

where w_{p1} and w_{p2} are weight coefficients, calculated by formulas (22) and (23), which are used to adjust the importance of relevance and redundancy. They must be dynamically adjusted to the size of the feature set and the number of selected features. F_{object} denotes the set of candidate features, L denotes the label vector of the dataset, G denotes the set of already selected features, where g_1 denotes the first already selected feature. $i = 0$ indicates that there is no feature in the G set, and $i = 1$ indicates that there is only one feature in the G set, $i > 1$ indicates that there are at least two features in the G set.

$$\boldsymbol{w_{p1}} = \boldsymbol{1} - \frac{\boldsymbol{1}}{\boldsymbol{1} + \boldsymbol{e}^{-\frac{t - T/2}{10}}} \tag{22}$$

$$\boldsymbol{w_{p2}} = \frac{\boldsymbol{1}}{\boldsymbol{1} + \boldsymbol{e}^{-\frac{t - T/2}{10}}} \tag{23}$$

where t represents the number of iterations and T represents the total number of iterations.

In Fig. 6, the input parameter F_{object} denotes the set of all features in the dataset, and the number of features in the set is gradually reduced to empty. L represents the label vector, i represents the number of features selected from F_{object}, and K represents the number of features in the dataset.

Algorithm 2 Pseudo-Code of the MPMMI

Input: w_{p1}, w_{p1}, number of features: K,
feature set: F_{object}, $i = 0$
Output: Sequence of features
Select feature g_1 from F_{object} according to formula (21) ($i = 0$)
G= {g_1}, F_{object}=F_{object}-G, i = i+1
Select feature g_2 from F_{object} according to formula (21) ($i = 1$)
G= {g_1, g_2}, F_{object}=F_{object}- {g_2}, i=i + 1
While i< K
Select feature g_{i+1} from F_{object} according to formula (21) (i >1)
$G = G$+{g_{i+1}}
$F_{object} = F_{object}$-{g_{i+1}}
$i = i$+1
End

B. APSO WRAPPER ALGORITHM

In this paper, the velocity and position update of the original PSO algorithm were improved. We named the new method "Adaptive Particle Swarm Optimization algorithm (APSO)".

1) UPDATES ON PARTICLE VELOCITIES

In the traditional PSO algorithm [31], the movement of particles is only guided by the local and global optima, while the interactions between particles are ignored. In addition, the acceleration factor for velocity update is usually fixed, which is generally taken as two. This setting may cause the particle to fall into the local optimal solution in the search space and thus fail to search globally for a better solution.

Adaptive Adjustment of Control (AAC) realizes flexible adjustment of particle velocity and position by dynamically adjusting the acceleration coefficient [35]. The introduction of AAC not only improves the convergence speed and global search ability of particle swarm optimization algorithm but also makes the algorithm more flexible and adaptive. This improvement enables the algorithm to cope better with the complex search space during the optimization process and effectively avoid falling into the dilemma of local optimal solutions.

The dynamic adjustment of the AAC is specified as follows:

$$AAC_t = \begin{cases} AAC_{t-1} + \delta & f_i^t < f_i^{t-1} \\ AAC_{t-1} - \delta & f_i^t > f_i^{t-1} \\ AAC_{t-1} & f_i^t = f_i^{t-1} \end{cases} \quad (24)$$

where f_i^t represents the fitness value of the ith particle in the tth iteration, $\mathrm{AAC_t}$ represents the value of the adaptively adjusted control parameter in tth iteration, and δ represents the adaptive increment, which is typically taken as 0.01.

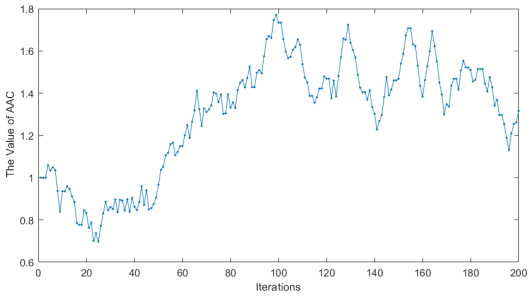

FIGURE 7. AAC trend with the number of iterations.

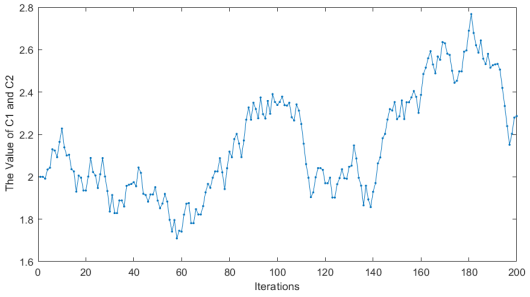

FIGURE 8. c_1 and c_2 trends with the change in the number of iterations.

Update the values of the learning factors $\boldsymbol{c_1}, \boldsymbol{c_2}$ by the values of AAC:

$$\boldsymbol{c_1} = \frac{\boldsymbol{\alpha}}{\boldsymbol{2}} \cdot (\boldsymbol{AAC_t} + \boldsymbol{1}) \quad (25)$$

$$\boldsymbol{c_2} = \boldsymbol{c_1} \quad (26)$$

where α is a constant, usually taken as two.

Because t is in the range of [1,200], formula (24) is used from the second iteration, and the initial value of AAC is set to one. Therefore, the value range of AAC is $[-1, 3]$, which indirectly affects the value ranges of $\boldsymbol{c_1}$ and $\boldsymbol{c_2}$ to be $[0, 4]$. The AAC value changes with the number of iterations, as shown in Fig. 7. Fig. 8 shows the changes in the learning factor with the number of iterations under the effect of AAC.

2) UPDATES ON PARTICLE POSITIONS

In the continuous PSO algorithm, the probability p-value of selecting the content [33] is an important parameter used to control whether the particle adds the change in current velocity to the position update. The discrete PSO algorithm can be used to better define this probability parameter.

Specifically, for each feature dimension of each particle, a random number feap(i) was generated. These random numbers should take values in the range [0, 1], indicating the magnitude of the likelihood that the feature will be selected. The random numbers for all feature dimensions are then averaged to obtain an overall probability p-value, that is,

$$\boldsymbol{p} = \frac{\sum_{i=1}^{n} \boldsymbol{feap}\ (\boldsymbol{i})}{\boldsymbol{n}} \quad (27)$$

where n is the number of features in the dataset.

The p-value defined in this manner allows for a more comprehensive decision on what to select by considering

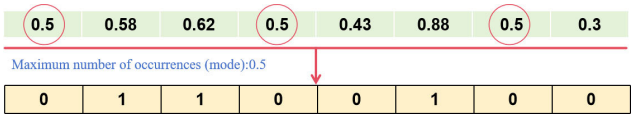

FIGURE 9. Mode adjustment method.

the randomness of the particle in each dimension. When performing particle position updates, we can decide whether to add the amount of change in the current velocity to the position update based on a random number and probability p-value for each feature dimension. Specifically, for each feature dimension i, if feap(i) $\leq p$, it means that the feature dimension is "selected," and the change in current velocity needs to be added to the position update. On the contrary, if feap(i) $> p$, the feature dimension is not "selected," and the position does not need to be updated.

$$x_{t+1} = \begin{cases} x_t + v_{t+1} & feap\,(i) \leq p \\ x_t & feap\,(i) > p \end{cases} \tag{28}$$

where x_{t+1} and v_{t+1} represent the position and velocity at the $(t+1)$ th iteration, respectively, which can be obtained using formula (15).

Algorithm 3 Pseudocode for APSO Algorithm

Input: $t = 1$, $T = 100$, w, selected_features= []
Output: selected_features
Initialize parameters
Initialize the particle swarm
While $t<T$
  Calculate the fitness function of the population and update the local optimal solution
  Update the value of AAC according to formula (24)
  Update the learning factors c_1 and c_2 according to formula (25,26)
  Update the velocity according to formula (15)
  Update the position according to formula (28)
  According to the number of the PSO adjusted not normalized location down to 0 or 1
  Update the global optimal solution
  $t=t+1$
Output selected_features
End

This method based on discrete PSO can better define the probability p-value for selecting content in the continuous PSO algorithm, thus increasing the flexibility and diversity of the algorithm. By introducing randomness and integrating the probabilities of each feature dimension, the particles can explore more comprehensively in the search space and can avoid falling into local optimal solutions.

Finally, our improved APSO algorithm was combined with SVM [39] to form a wrapper algorithm. In addition, this algorithm is used to perform feature selection to find the optimal solution, that is, to find the optimal particle location in the search space.

3) LEGACY POSITION ADJUSTMENT METHOD

In traditional methods, we perform the algorithm after the corresponding position of those features is not 0 or 1 value, according to a random number judgment, which is attributed to 0 or 1. However, this cannot ensure the accuracy of the value and may even appear for all 0 or 1 results. Our ultimate goal is to make the data more convincing, the results more accurate, and the swarm intelligence optimization algorithm can better jump out of the local optimal and find the global optimal. Therefore, the value adjusted by the algorithm is the value that appears the most times in each iteration (the mode) as the judgment basis. If the value of the corresponding position of the feature is greater than that of the mode, it is set to one, and vice versa, it is set to 0, as shown in Fig. 9.

By choosing the value that occurs most times in each iteration as the basis for judgment, we can better reflect the results of the algorithm after execution. Compared with random number judgment, the use mode can provide more accurate feature values. Random numbers may lead to unstable results, whereas the mode can reduce this uncertainty to some extent, so that the value of the corresponding position of the feature can be determined more accurately. In addition, using this mode as a basis for judgment can reduce data volatility. Choosing the value that occurs most times in each iteration can avoid drastic changes in feature values between iterations, making the data more stable, which in turn facilitates PSO to better jump out of the local optimal.

C. FRAMEWORK OF THE IGMPMMIAPSO ALGORITHM

This section introduces the framework of the algorithm, including the change in the distance value, population initialization method, pseudo-code, and algorithm flowchart. The algorithm is combined with filter algorithms and wrapper algorithms by introducing the collision distance value (distance) and No Consecutive Changes (NCC) to prevent the algorithm from falling into local optimal solutions.

1) ALGORITHM FRAMEWORK

In optimization algorithms, we often face the problem of "not being able to jump out of the local optimum," which causes the algorithm to fall into the limitation of the optimal solution and fail to find the global optimal solution. To avoid this situation, this section introduces a framework based on collision distance and No Consecutive Changes (NCC), as shown in Fig. 10. The algorithm uses a nested loop structure, with each module showing the specifics of the loop.

The key to the entire algorithm is the judgment of the filter algorithm. The filter algorithm is executed only if the collision distance is greater than the number of NCC. However, the filter algorithms in this paper were executed in two ways: univariate filter (IG algorithm in Section B of Part II) and bivariate filter (MPMMI algorithm in Section A of Part III), depending on the number of changes in the collision distance value. After the execution of the filter algorithm, a subset of candidate features is generated and provided to the wrapper

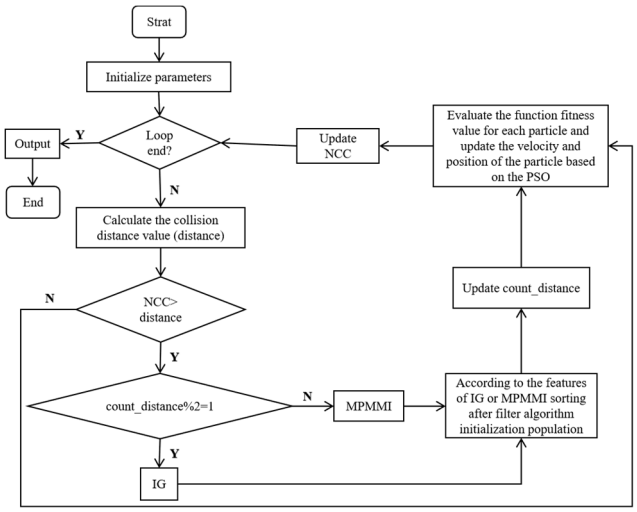

FIGURE 10. Framework of the IGMPMMIAPSO algorithm.

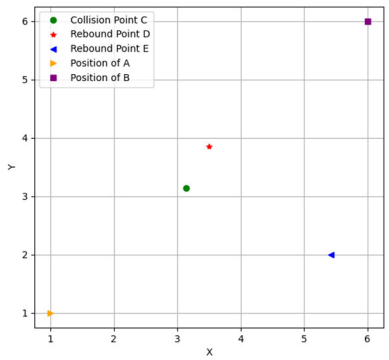

FIGURE 11. Location diagram of collision point.

algorithm. Based on the candidate feature subset, the wrapper algorithm determines the local optimum iteratively. By executing the filter algorithm multiple times, a local optimum is found in the newly generated feature subset each time. In this manner, the algorithm can overcome the limitations of the local optimum and determine the global optimal solution. At the end of the loop, the global optimum was the output.

The collision distance values were calculated based on the small ball collisions in Section A of Part II. When two objects collide, according to the law of conservation of momentum and the law of conservation of energy, we can deduce the position of the collision point and change in the velocity of the objects after the collision. In this paper, we assume a perfectly elastic collision between spheres A and B, that is, kinetic energy is conserved before and after the collision and momentum is conserved before and after the collision.

In this paper, we simulate a ball collision in a two-dimensional space where rebound occurs and stops, as shown in Fig. 11, which is a two-dimensional coordinate system with X-Y axes. Assuming that the positions of points A and B are the initial positions, they are denoted as P_A and P_B, where the coordinates of point A can be denoted as $P_A(A_x, A_y)$(the subsequent positions of B, C, D and E are denoted similarly). During the iteration process, the mass of the ball was constant; however, the velocity of the two balls changed with the number of iterations.

In Fig. 11, the two balls depart from points A and B with different velocities, collide at point C(denoted as P_C), and rebound after the collision with some energy loss, possibly stopping at the positions of points D and E (denoted as P_D and P_E) after running for a period of time.

First, the position of collision point C was calculated. It can be obtained from the initial positions of balls A and B and their velocities before collision, that is,

$$\boldsymbol{P_C} = \boldsymbol{P_A} + (\boldsymbol{P_B} - \boldsymbol{P_A}) \cdot \frac{\boldsymbol{V_A}}{\boldsymbol{V_A} + \boldsymbol{V_B}} \tag{29}$$

where P_A and P_B are the initial positions of balls A and B, respectively, and V_A and V_B are the velocities, that is, the initial velocities.

Second, the rebound stopping points D and E, which are the positions of balls A and B when they stop. We denote the velocities of balls A and B as v_{Af} and v_{Bf} after the collision. According to formulas (1)-(4), v_{Afx}, v_{Afy}, v_{Bfx} and v_{Bfy} can be obtained, so we can calculate the position of the two points P_D and P_E using the position of the collision point C and the velocity and energy loss after the collision, as follows:

$$\boldsymbol{P_D} = \boldsymbol{P_C} - \left(v_{Afx}\frac{\boldsymbol{v_{Afx}}}{\boldsymbol{loss}}, v_{Afy}\frac{\boldsymbol{v_{Afy}}}{\boldsymbol{loss}}\right) \tag{30}$$

$$\boldsymbol{P_E} = \boldsymbol{P_C} + \left(v_{Bfx}\frac{\boldsymbol{v_{Bfx}}}{\boldsymbol{loss}}, v_{Bfy}\frac{\boldsymbol{v_{Bfy}}}{\boldsymbol{loss}}\right) \tag{31}$$

where, *loss* is the energy loss value with value in the range of [0, 1].

The positions of the collision points C, D and E were calculated using formulas (29∼31). The distance from point C to points D and E can be calculated by the following formulas.

$$\boldsymbol{CD} = \sqrt{(\boldsymbol{P_C} - \boldsymbol{P_D})^2} \tag{32}$$

$$\boldsymbol{CE} = \sqrt{(\boldsymbol{P_C} - \boldsymbol{P_E})^2} \tag{33}$$

where P_C, P_D and P_E are the coordinates of the points in two-dimensional space, and the distance between two points is based on the coordinates of the corresponding points. For example, in Fig. 9, points A and B are (1,1) and (6,6), respectively; therefore, the distance of AB is $\sqrt{(6-1)^2 + (6-1)^2} = 5\sqrt{2}$. Therefore, both CD and CE are available.

In each iteration, because the coordinates of the points at the initial positions of the two balls and the initial velocities are taken randomly, the value of the distance resulting from the collision also changes with the number of iterations. However, there may be cases where the values are all zero, and the resulting CD, and CE may also be zero. To prevent such a result, a natural constant e was chosen to compute the distance further.

In Fig. 12, the NCC with an initial value of 1 is greater than that with an initial value of 0. Therefore, it is determined in the first iteration. The collision distance

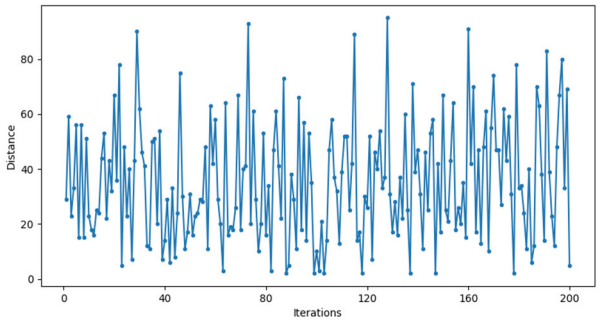

FIGURE 12. The change trend of distance with the number of iterations.

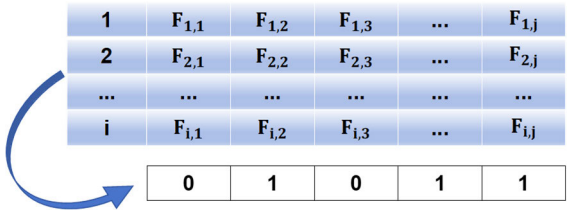

FIGURE 13. Randomized method initialized.

was calculated as follows:

$$\boldsymbol{distance} = \lfloor \boldsymbol{max}\,(\boldsymbol{CD}, \boldsymbol{CE}, \boldsymbol{e}) \rfloor \tag{34}$$

where ⌊max(CD, CE, e)⌋ represents the larger value between CD, CE and e rounded down.

When the collision distance is large, the optimal value can be determined through several iterations. When the collision distance value is small, a filter algorithm is typically used, and the number of changes in the collision distance value is used to determine whether it is a univariate filter algorithm (IG) or a bivariate filter algorithm (MPMMI). New popu-lations are formed by readjusting different combinations of features such that the algorithm achieves the goal of jumping out of the local optimum and finding the global optimum.

2) SELECTION OF THE NUMBER OF FEATURES

In the feature selection process, we usually need to consider the dataset, vector labels, and number of features to be selected, denoted as K. In general, because the dataset is not considered, the value of K is fixed. In this case, the best subset of features may not be selected on different datasets or different problems. Therefore, determining the value of K to be selected based on the information in the dataset may be more flexible and effective.

In this paper, we wish to select as few features as possible to avoid overfitting and to improve the interpretability of the algorithm. However, we must retain sufficient information to maintain the predictive power of the IGMPMMIAPSO algorithm. To significantly reduce the dimensionality of the data, we must select an appropriate number of features in high-dimensional datasets. Therefore, we used logarithm [36] to construct a function to determine the value of K to achieve dimensionality reduction of the data.

$$\boldsymbol{y} = \boldsymbol{log}_{\varepsilon}\boldsymbol{x} \tag{35}$$

TABLE 1. Range of y variation under the action of different values of ε.

Number	ε	Span of y	Range of y
1	1.05	41.30	155.79~197.08
2	1.06	34.58	130.45~165.02
3	1.07	29.78	112.33~142.12
4	1.08	26.18	98.76~124.94
5	1.09	23.38	88.21~11.58
6	1.1	21.14	79.75~100.89
7	1.11	19.31	72.83~92.14
8	1.12	17.78	67.07~84.85

where x represents the number of features in the dataset and y represents the number of selected features. For the value of ε, we use 0.01 as an interval, which ranges from 1.01 to 1.19. The value of y changes when different values are provided, as listed in Table 1. Here, we chose any number from 1.06 to 1.12.

In a given dataset, the number of features x is determined, thus determining the range of values for the response variable y. However, to select the optimal subset of features to improve the algorithm's performance, two random numbers are introduced to adjust the value of K. The selection of the value of K provides flexibility in selecting a subset of features, which further optimizes its predictive power.

$$\boldsymbol{K} = \left|\boldsymbol{int}\left(\boldsymbol{y} + \boldsymbol{i} \cdot \boldsymbol{rand}\,() \cdot \mathbf{0.1} \cdot \boldsymbol{q}\right) - \boldsymbol{randint}\right| \tag{36}$$

where i represents the number of particles in the dataset; q is randomly determined, representing positive or negative numbers; rand() is a random number; randint is a random integer, usually from 1 to 10, y is calculated by formula (35); || indicates that the absolute value of the result is taken to avoid K being negative.

During multiple dimensionality reductions, different K values may be obtained; thus, the repeatability of the dimensionality reduction is low. Simultaneously, we can obtain a feature subset with high classification accuracy and low length.

3) INITIALIZATION METHOD

In swarm intelligent optimization methods, population initialization is especially important because it directly affects the search space and the final optimization performance of the algorithm.

In this paper, a randomized initialization method was adopted after considering the change in the filter algorithm on the ranking of features [36]. For the random initialization of each individual in the population, when the value of the corresponding position of the feature is greater than or equal to 0.5, it is set to 1, which indicates that the feature is selected; otherwise, it is set to 0, which indicates that the feature is not selected. As shown in Fig. 13, this method can introduce a certain amount of randomness, simulate a real

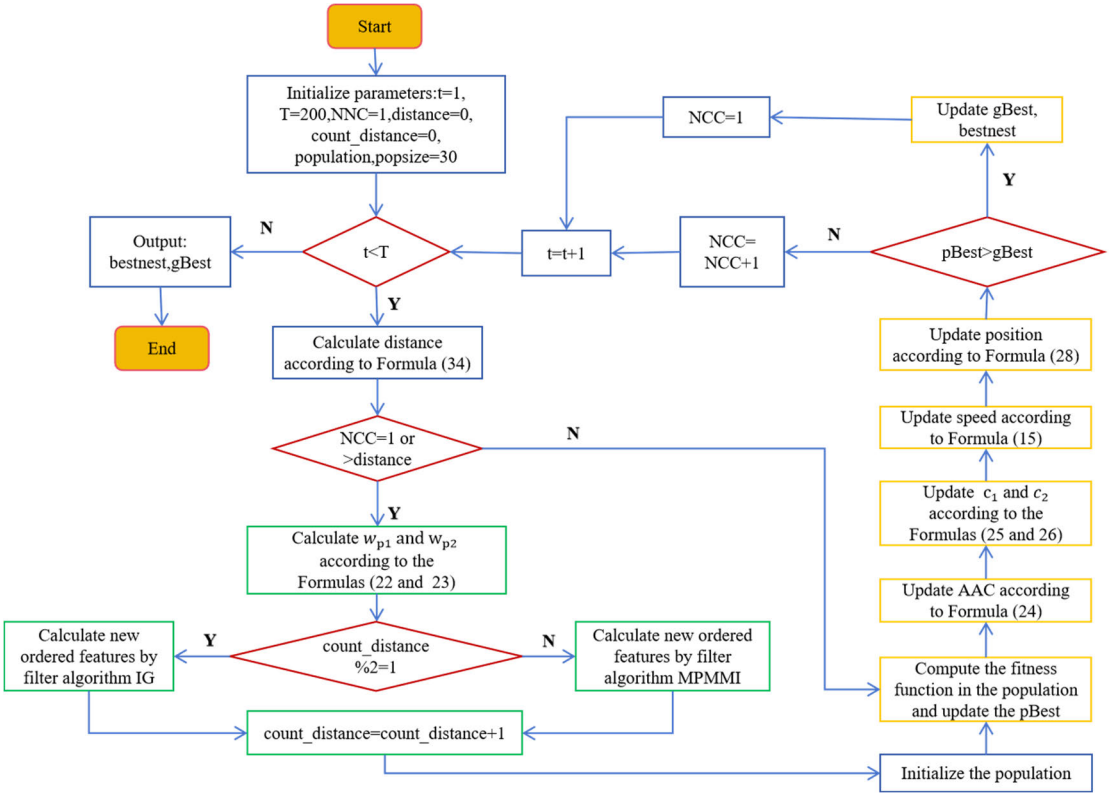

FIGURE 14. Flowchart of the IGMPMMIAPSO algorithm.

situation, increase the exploration ability of the search space, and allow flexibility in adjusting the probability of features being selected.

4) PSEUDOCODE AND FLOWCHART OF THE ALGORITHM

Fig. 14 shows the flowchart of the proposed algorithm IGMPMMIAPSO, and Algorithm 4 is its pseudocode.

IV. EXPERIMENTAL RESULTS AND ANALYSIS

This section introduces the datasets used for the experiments, the experimental parameter settings of the proposed algorithm and comparison algorithms, and the experimental results and analysis.

A. DATASETS AND PARAMETER SETTINGS

1) DATASETS

To validate the superiority of the IGMPMMIAPSO algorithm, we performed a series of tests on eight datasets selected from the Gene Expression Model Selector, which were evaluated based on the following datasets: Breast, CNS, Detect, DLBCL, GLI-85, Leukemia, secom, and SMK-CAN-187.

These datasets cover the number of classes, samples, and features for each dataset. In our experiments, these datasets were analyzed using the IGMPMMIAPSO algorithm and its performance are evaluated based on the details listed in Table 2.

Algorithm 4 Pseudo-Code IGMPMMIAPSO

Input: dataset, $t = 1$, $T = 200$, NNC=1, distance=0, population, popsize =30, count_distance=0
Output: bestnest, gBest
While t<T
  Calculate the distance according to formula (34)
  If NCC=1 or NNC>distance
    Update w_{p1} and w_{p2} according to formulas (22) and (23)
    If the number of collision distance value changes is odd
      Calculate the new ordered features by the filter algorithm IG
    Else
      Calculate the new ordered features by the filter algorithm MPMMI
    Endif
    count_distance=count_distance+1
  Endif
  Calculate the population fitness function and update pBest
  Update AAC according to formula (24)
  Update c_1 and c_2 according to the formula (25), (26)
  Update velocity according to formula (15)
  Update position according to formula (28)
  If pBest > gBest
    Update gBest, bestnest
    NCC=1
  else
    NCC=NCC+1
  Endif
  t=$t+1$
End

2) PARAMETER SETTINGS

To compare the hybrid feature selection algorithm (hybrid algorithm) with the proposed IGMPMMIAPSO algorithm,

TABLE 2. Introduction to the datasets.

No.	Datasets	Class	Instances	Features	Abbreviation
1	Breast	2	97	24481	Bre
2	CNS	2	60	7129	CNS
3	Detect	3	373	531	Det
4	DLBCL	2	77	5469	DLB
5	GLI-85	2	85	22283	GLI
6	Leukemia	2	72	7070	Leu
7	Secom	2	1569	590	sec
8	SMK-CAN-187	2	187	19997	SMK

we used similar classifiers and parameter settings. Among the five hybrid algorithms with which they were compared, the performance depended on the selected classifiers and parameter settings. In our proposed IGMPMMIAPSO algorithm, we followed settings similar to those listed in Table 3 to ensure the accuracy and reliability of the experimental results.

In this paper, by using similar parameter settings, we can more accurately reveal the advantages of the proposed IGMPMMIAPSO algorithm over the hybrid algorithms being compared. In the proposed algorithm, we discarded the traditional maximum number of iterations of 100 and changed it to 200. However, it will still be executed ten times for each dataset. An increase in the number of iterations effectively prevents the algorithm from falling into a local optimum, which in turn enables the algorithm in this paper to better find the global optimum and obtain the best classification accuracy (ACC).

The fitness function is computed using an SVM classifier. The solution to this problem is to select the feature subset with the highest classification accuracy based on the SVM classifier. The penalty parameter and RBF parameters were selected using the grid search method.

The classification accuracies of the datasets in Table 4 used the ten-fold cross-validation technique, which is a common model evaluation method. By dividing the dataset into ten loops, each loop was divided into ten groups, one for testing and nine for training. Each loop produces classification accuracy by taking the average of these accuracies as the result of the fitness function.

B. EXPERIMENTAL RESULTS AND COMPARISON

Eight datasets and six algorithms were used in this experiment. Each algorithm runs ten times on each dataset. Table 4 shows the average classification accuracy and average feature subset length (LEN) achieved by the five hybrid algorithms and the proposed algorithm on the eight datasets. The results of the proposed algorithm are shown to outperform other algorithms on the eight datasets.

Table 4 shows that the IGMPMMIAPSO algorithm achieves the highest average classification accuracy in eight datasets. In Detect, DLBCL, GLI-85 and Leukemia datasets, the classification accuracy of IGMPMMIAPSO algorithm is greater than 95%. Compared to other compared algorithms, the IGMPMMIAPSO algorithm obtained the shortest feature subset length in most cases.

In the Detect, DLBCL, secom and SMK-CAN-187 datasets, the IGMPMMIAPSO algorithm has higher ACC than all other algorithms and the shortest feature subset length (LEN). On the Breast dataset, the value of ACC is higher than that of the other algorithms, but the feature subset length is longer than that of the mRMR+PSO and mRMR+GWO algorithms. In CNS and Leukemia datasets, the values of ACC are equal to those of other algorithms, but in terms of feature subset length, the CNS and Leukemia datasets are longer than those of the mRMR+CS algorithm. In addition, for the GLI-85 dataset, although the value of ACC was lower than that of the other algorithms, feature subset length was the shortest. It also proves that the proposed algorithm can obtain a smaller feature subset length than other algorithms.

As can be seen, our proposed algorithm shows advantages in both classification accuracy and optimal feature subset length on most datasets. Compared to the other five hybrid algorithms, the proposed algorithm has a higher classification accuracy, and the selected feature subset length is smaller. This also proves that the proposed algorithm can effectively improve the classification accuracy when dealing with various datasets, and can select representative features more effectively. Thus, reducing the length of the feature subset while improving the classification accuracy.

The algorithm introduces the concepts of collision distance values and NCC, thus effectively integrating filter algorithms and wrapper algorithms. When the local optimum value does not change after many iterations, the algorithm performs a filter algorithm to generate another subset of candidate features. The wrapper algorithm then obtains a new local optimum based on a new subset of candidate features. Thus, the algorithm can obtain the global optimum while actively breaking through the multiple local optima.

C. EXPERIMENTAL ANALYSIS

1) EXECUTION OF FILTER ALGORITHM FOR COLLISION DISTANCE VALUE

In the experiment, the IGMPMMIAPSO algorithm was executed ten times on each dataset and the number of filter algorithm execution was counted. Then the generation times of the collision distance value can be used to judge whether the filter algorithm is univariate or bivariate. Finally, the statistical results are shown in Fig. 15.

In Fig. 15, the X-axis represents the number of runs of the dataset, ranging from 1 to 10. The Y-axis represents eight different datasets. The Z-axis represents the number of times the filter algorithm is executed during the run of the dataset. When the dataset is executed ten times, each time the filter algorithm is executed a different number of times. Maximum number of executions is twenty-two and minimum is eleven. As can be seen from Fig. 15, the number of iterations (T) is a

TABLE 3. Experimental parameter settings.

No.	Algorithm	Experimental parameter
1	mRMR+PSO	Particle: 30, maximum weight value (w_{max}):0.9, minimum weight value (w_{min})：0.4, Maximum iterations:200
2	mRMR+GWO	Wolves: 30, initial contraction coefficient: 0.5, final contraction coefficient: 0.01, Maximum iterations:200
3	mRMR+BBA	Bat: 30, loudness: 1.5, pulse rate: 0.5, minimum frequency value: 0, maximum frequency value 1, Maximum iterations:200
4	mRMR+CS	Cuckoo: 30, discovery probability: 0.25, Levi's flight parameters: 1.5, Maximum iterations:200
5	mRMR+GA	Chromosome number: 30, crossover probability (P_c): 0.7, the mutation probability (P_m): 0.5, Maximum iterations:200
6	IGMPMMIAPSO	Particle: 30, maximum weight value (w_{max}):0.9, minimum weight value (w_{min})：0.4, Maximum iterations:200

TABLE 4. Average classification accuracy and feature subset length of six algorithms on eight datasets.

No.	Dataset	mRMR+PSO		mRMR+GWO		mRMR+BBA		mRMR+CS		mRMR+GA		IGMPMMI APSO	
		ACC	LEN	ACC	LEN	ACC	LEN	ACC	LEN	ACC	LEN	ACC	LEN
1	Bre	56.6	24.1	56.5	**22.7**	56.4	50.2	56.6	59.2	55.2	51.9	**56.9**	44.0
2	CNS	65.0	32.2	65.0	31.3	65.0	47.6	65.0	**17.0**	65.0	39.3	**65.0**	27.7
3	Det	99.6	32.7	99.9	43.5	99.8	49.5	99.8	52.5	99.5	53.7	**99.9**	**18.0**
4	DLB	100.0	49.9	100.0	60.5	100.0	52.7	100.0	40.4	100.0	41.1	**100.0**	**16.5**
5	GLI	99.8	25.9	**100.0**	44.5	99.4	48.3	99.3	51.7	98.2	50.1	96.6	**25.5**
6	Leu	100.0	50.0	100.0	68.8	100.0	53.0	100.0	18.9	100.0	40.5	**100.0**	**38.7**
7	sec	93.4	26.3	93.4	37.0	93.4	51.0	93.4	19.3	93.4	40.8	**93.5**	**17.2**
8	SMK	58.1	52.3	71.7	79.3	68.0	64.2	68.6	52.0	59.6	57.6	**72.7**	**48.0**

Bold indicates the optimal value.

fixed value, and its value is 200. The more times the filter algorithm is executed, the greater the number of different feature subsets it provides. Therefore, the more local optimal values the wrapper algorithm can obtain. Conversely, the fewer times the filter algorithm is called, the more often the wrapper algorithm is updated in the local optimum.

In summary, the wrapper algorithm is not affected by the number of calls to the filter algorithm and can obtain more optimal values, thereby obtaining the global optimal value.

2) RELATIONSHIP BETWEEN COLLISION DISTANCE VALUE AND NUMBER OF NO CONSECUTIVE CHANGES

The variation in the collision distance is closely related to the value of NCC. From Fig. 15, we can see the effect of the filter algorithm being executed 10 times on eight datasets. In Fig. 16 to 23, we take one of them for a detailed analysis. Through the scatterplot, it is easy to find that the changes in the collision distance value and NCC show a certain pattern as the number of iterations increases.

When the value of the collision distance changes, the value of NCC also changes accordingly, indicating that they are closely related. At the same time, we can also see that at some specific iteration points, the collision distance values and NCC values change abruptly, which may correspond to the moments when the filter algorithm is reinvoked.

In Fig. 16 to 23, the X-axis represents the number of iterations, from 1 to 200; and the Y-axis represents the values of distance and NCC. Each execution of the filter algorithm was performed with NCC reset to one. There are two possible scenarios for the execution of the filter algorithm: 1) the maximum value changes, and the NCC needs to be recalculated; and 2) the value of NCC exceeds the collision distance value, and the filter algorithm needs to be invoked to reform a subset of the candidate features and start a new round of exploration.

As can be seen from Fig. 16 to 23, the distribution of scatters is denser in the Breast, CNS, Detect, DLBCL, and SMK-CAN-187 datasets, which proves that the filter algorithm has been invoked many times on these five datasets in the current loop. In contrast, for the three datasets, GLI-85, Leukemia, and secom, the distribution of scatters is sparser, indicating that the filter algorithm has been invoked fewer times in the current loop.

This also proves that the execution times of the filter algorithm are random, and the experimental results are not accidental and the algorithm falls into the local optimal situation. In addition, the filter algorithm is invoked to determine

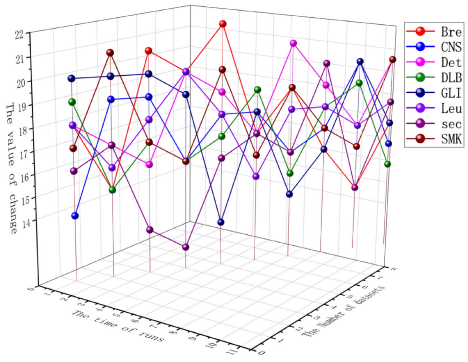

FIGURE 15. Filter algorithm in each dataset is called the number of times.

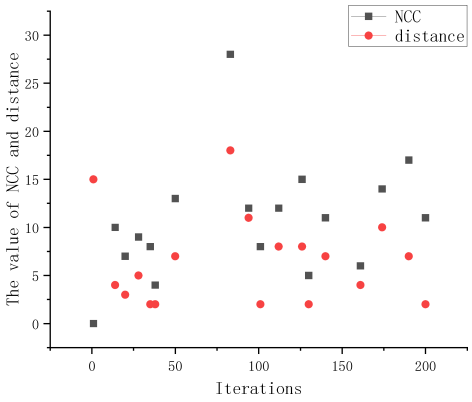

FIGURE 16. Trend of NCC and distance on Bre.

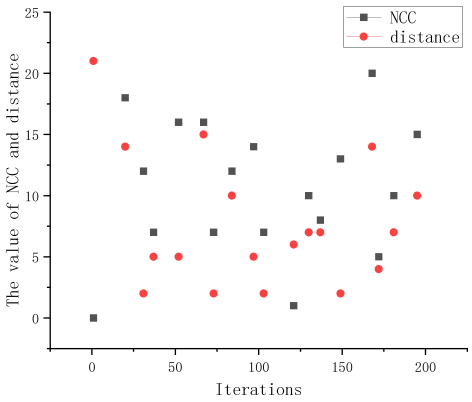

FIGURE 17. Trend of NCC and distance on CNS.

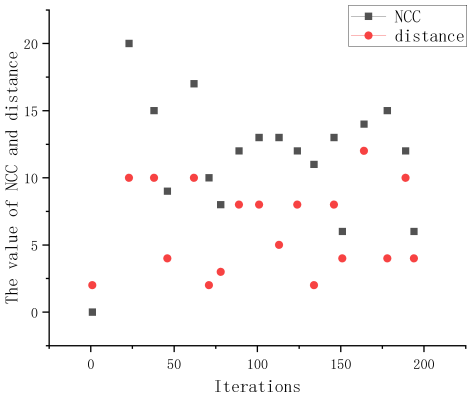

FIGURE 18. Trend of NCC and distance on Det.

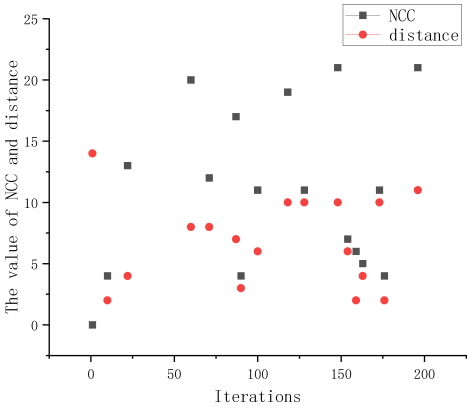

FIGURE 19. Trend of NCC and distance on DLB.

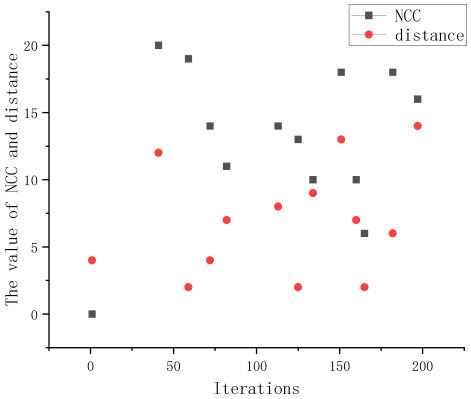

FIGURE 20. Trend of NCC and distance on GLI.

whether it is a univariate or multivariate, based on the number of times the collision distance value is generated.

3) CHANGES IN FEATURE SUBSETS

The proposed algorithm was executed ten times on each dataset, but the value of K was taken differently for these ten times. This means that for each dataset, we performed several experiments with different parameters.

Fig. 24 shows the results of this multiple-loop experiment on eight datasets for dimensionality reduction. From this figure, we can clearly observe the number of features in each dataset as well as the changes in the value of K and the value of LEN (the first K features selected by the filter algorithm and then selected by executing the wrapper algorithm, which indicates the length of the optimal subset of features). Thus, it is possible to better understand the performance and effect of the algorithm on different datasets.

For the Detect dataset, Fig. 25 shows the relationship between K and LEN after executing the IGMPMMIAPSO algorithm ten times. There were four times when the value of K exceeded 40 and six times when it was less than 40. There were five times when LEN was a one-digit value. Multiple values of K enhance randomness and prevent chance. At the same time, it results in better convergence for dimensionality reduction of the wrapper algorithm.

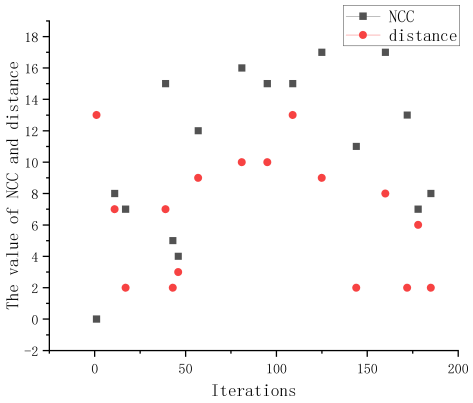

FIGURE 21. Trend of NCC and distance on Leu.

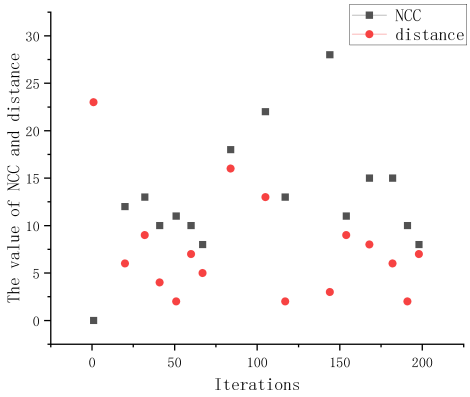

FIGURE 22. Trend of NCC and distance on sec.

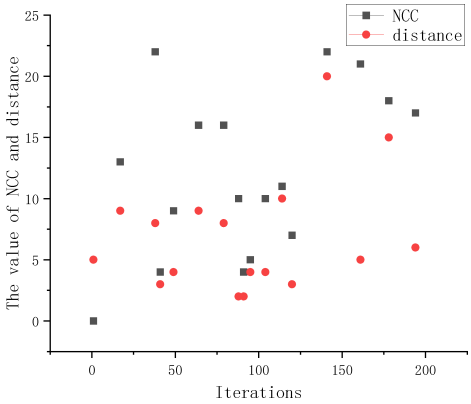

FIGURE 23. Trend of NCC and distance on SMK.

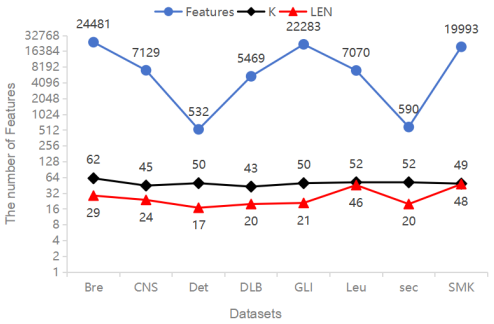

FIGURE 24. Dimensionality reduction effect on 8 datasets.

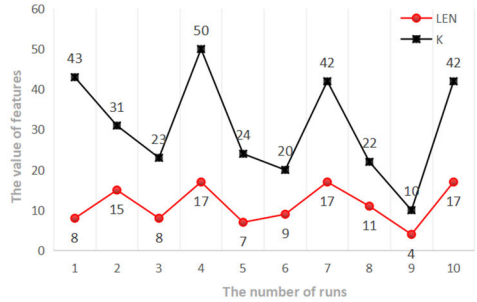

FIGURE 25. Dimension reduction effect performed 10 times on Detect.

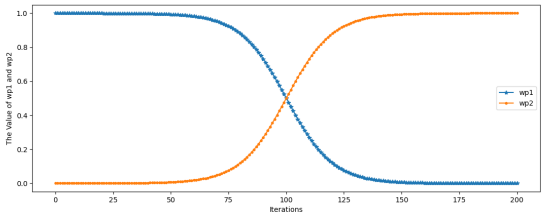

FIGURE 26. Trend of and on secom.

and w_{p2} are equal at t=100. w_{p1} and w_{p2} values affect the ratio of relevance to redundancy, and Pearson's relevance is more influential at $t<100$, which indicates that when the number of features is small, the relevance between the features and labels is predominant in the selection of the features. At $t>100$, Mutual Information redundancy is more influential, indicating that redundancy between features dominates feature selection when the number of features gradually increases.

4) CHANGES IN WEIGHT COEFFICIENTS

In the bivariate filter algorithm (MPMMI), to provide a subset of candidate features, we introduce two weight factors as a measure of relevance and redundancy, that is, w_{p1} and w_{p2}. As the values of the two weight factors vary with the number of iterations, the bivariate filter algorithm provides a feature subset rich in diversity.

Fig. 26 illustrates the variation in the two weight parameters, w_{p1} and w_{p2}, with the number of iterations on the secom dataset.

From the figure, it can be observed that w_{p1} gradually increases and w_{p2} gradually decreases, and the values of w_{p1}

5) CHANGES IN ADAPTIVE INCREMENTS

On each dataset, the proposed algorithm was executed 10 times, each time with 200 iterations. However, the adaptive increment δ was randomly generated with the number of iterations. Its value range is controlled in the range of [0,0.1] to prevent the algorithm from converging too fast and falling into a local optimum. The change in the value of δ in a particular loop over 200 iterations on the eight datasets is plotted in Fig. 27.

In Fig. 27, the X-axis represents the number of iterations, and the Y-axis represents the value of the adaptive increment Delta(δ). In 200 iterations, the value of δ is randomly taken to

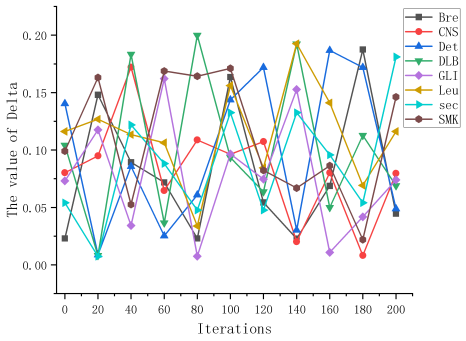

FIGURE 27. Changes in δ during the iteration of eight datasets.

TABLE 5. Computational complexity of the comparison algorithm and the proposed algorithm.

No.	Algorithm	Computational Complexity
1	mRMR+PSO	$O(T \times n^2 \times S)$
2	mRMR+GWO	$O(T \times n^2 \times S)$
3	mRMR+BBA	$O(T \times n^2 \times S)$
4	mRMR+CS	$O(T \times n^2 \times S)$
5	mRMR+GA	$O(T \times n^2 \times S)$
6	IGMPMMIAPSO	$O(T \times (n \times m) \times S)$

ensure that the δ is not fixed during each iteration. However, δ affects the adjustment of AAC, which in turn affects the change in learning factors. In this paper, the dynamic adjustment method was used to adjust the learning factors to update the particle velocity. This is more flexible than the update of the traditional PSO, and the particles can better find the global optimal solution.

D. COMPUTATIONAL COMPLEXITY

The computational complexity is the amount of resources required to execution of an algorithm. These resources include time (the number of steps required for execution) and space (the amount of memory required). In this paper, the computational complexity of the five algorithms used for comparison with the proposed IGMPMMAPSO algorithm is presented in tabular form in Table 5.

where T denotes the maximum number of iterations, m denotes the number of features we selected, n denotes the number of features in the dataset, and S denotes the time required to execute the SVM classifier.

From Table 2, the number of features we selected is far less than the original number and m is much less than n. That means that $n \times m$ is less than n^2. So, when compared with other algorithms, the proposed algorithm has a lower computational complexity than the other five hybrid algorithms and takes less time. This also means that the IGMPMMIAPSO algorithm can process more data or perform more iterations in the same time, which not only improves the performance but also increases the efficiency of the algorithm. In addition, the low computational complexity also means that the IGMPMMIAPSO algorithm may be better suited to resource-limited situations, as it can complete computational tasks in a shorter period of time.

V. CONCLUSION

In the feature selection process, a hybrid feature selection algorithm IGMPMMIAPSO, was proposed to prevent the algorithm from falling into local optima. The algorithm mainly consists of IG, MPMMI, and APSO, which adjust the order and frequency of execution by changing the NCC and collision distance values. Compared to other hybrid algorithms, the proposed algorithm provides a larger subset of candidates while preventing filter algorithms from being called frequently. Through experiments, we verified the effectiveness of the IGMPMMIAPSO algorithm for feature selection. The algorithm fully demonstrates the multiple relationships between features and labels, features and features. Such multifaceted considerations improve the accuracy and robustness of feature selection. The experimental results show that the classification accuracy is at least 0.1% higher than that of the other algorithms on some datasets. However, on other datasets, IGMPMMIAPSO provides a shorter subset of features.

Comprehensive experimental results show that our algorithm achieves satisfactory results in feature selection and provides strong support for feature selection and data modelling in practical applications. With the widespread use of multimodal datasets in practical applications, we can consider how to extend the algorithm to deal with the feature selection problem of multimodal datasets and how to realize dynamic feature selection in the future. This method can cope with the situation that the data feature distribution changes with time, so as to achieve a more flexible and adaptive feature selection method.

REFERENCES

[1] X.-A. Ma, H. Xu, and C. Ju, "Class-specific feature selection via maximal dynamic correlation change and minimal redundancy," *Expert Syst. Appl.*, vol. 229, Nov. 2023, Art. no. 120455, doi: 10.1016/j.eswa.2023.120455.

[2] T. Wu, Y. Hao, B. Yang, and L. Peng, "ECM-EFS: An ensemble feature selection based on enhanced co-association matrix," *Pattern Recognit.*, vol. 139, Jul. 2023, Art. no. 109449, doi: 10.1016/j.patcog.2023.109449.

[3] Z. Fei, Y. Ryeznik, O. Sverdlov, C. W. Tan, and W. K. Wong, "An overview of healthcare data analytics with applications to the COVID-19 pandemic," *IEEE Trans. Big Data*, vol. 8, no. 6, pp. 1463–1480, Dec. 2022, doi: 10.1109/TBDATA.2021.3103458.

[4] H. Khalajzadeh, M. Abdelrazek, J. Grundy, J. Hosking, and Q. He, "Survey and analysis of current end-user data analytics tool support," *IEEE Trans. Big Data*, vol. 8, no. 1, pp. 152–165, Feb. 2022.

[5] Z. Tang, W. Jia, X. Zhou, W. Yang, and Y. You, "Representation and reinforcement learning for task scheduling in edge computing," *IEEE Trans. Big Data*, vol. 8, no. 3, pp. 795–808, Jun. 2022.

[6] X. Wen and Z. Xu, "Wind turbine fault diagnosis based on ReliefF-PCA and DNN," *Expert Syst. Appl.*, vol. 178, Sep. 2021, Art. no. 115016, doi: 10.1016/j.eswa.2021.115016.

[7] X. Qiao, T. Peng, N. Sun, C. Zhang, Q. Liu, Y. Zhang, Y. Wang, and M. Shahzad Nazir, "Metaheuristic evolutionary deep learning model based on temporal convolutional network, improved Aquila optimizer and random forest for rainfall-runoff simulation and multi-step runoff prediction," *Expert Syst. Appl.*, vol. 229, Nov. 2023, Art. no. 120616, doi: 10.1016/j.eswa.2023.120616.

[8] C. Ding and H. Peng, "Minimum redundancy feature selection from microarray gene expression data," *J. Bioinf. Comput. Biol.*, vol. 3, no. 2, pp. 185–206, May 2005.
[9] K. Qu, J. Xu, Q. Hou, K. Qu, and Y. Sun, "Feature selection using information gain and decision information in neighborhood decision system," *Appl. Soft Comput.*, vol. 136, Mar. 2023, Art. no. 110100, doi: 10.1016/j.asoc.2023.110100.
[10] P. Tao, Z. Sun, and Z. Sun, "An improved intrusion detection algorithm based on GA and SVM," *IEEE Access*, vol. 6, pp. 13624–13631, 2018.
[11] S. Mirjalili and A. Lewis, "The whale optimization algorithm," *Adv. Eng. Softw.*, vol. 95, pp. 51–67, May 2016.
[12] L. Song, Q. Liang, H. Chen, H. Hu, Y. Luo, and Y. Luo, "A new approach to optimize SVM for insulator state identification based on improved PSO algorithm," *Sensors*, vol. 23, no. 1, p. 272, Dec. 2022, doi: 10.3390/s23010272.
[13] Z. Song, S. Liu, M. Jiang, and S. Yao, "Research on the settlement prediction model of foundation pit based on the improved PSO-SVM model," *Sci. Program.*, vol. 2022, pp. 1–9, Mar. 2022, doi: 10.1155/2022/1921378.
[14] T. Gao and H. Chen, "Multicycle disassembly-based decomposition algorithm to train multiclass support vector machines," *Pattern Recognit.*, vol. 140, Aug. 2023, Art. no. 109479, doi: 10.1016/j.patcog.2023.109479.
[15] J. Shi, X. Chen, Y. Xie, H. Zhang, and Y. Sun, "Delicately reinforced k-nearest neighbor classifier combined with expert knowledge applied to abnormity forecast in electrolytic cell," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 35, no. 3, pp. 3027–3037, Mar. 2024.
[16] K. Chen, B. Xue, M. Zhang, and F. Zhou, "An evolutionary multitasking-based feature selection method for high-dimensional classification," *IEEE Trans. Cybern.*, vol. 52, no. 7, pp. 7172–7186, Jul. 2022.
[17] J.-Q. Yang, Q.-T. Yang, K.-J. Du, C.-H. Chen, H. Wang, S.-W. Jeon, J. Zhang, and Z.-H. Zhan, "Bi-directional feature fixation-based particle swarm optimization for large-scale feature selection," *IEEE Trans. Big Data*, vol. 9, no. 3, pp. 1004–1017, Jun. 2023.
[18] T. Thaher, H. Chantar, J. Too, M. Mafarja, H. Turabieh, and E. H. Houssein, "Boolean particle swarm optimization with various evolutionary population dynamics approaches for feature selection problems," *Expert Syst. Appl.*, vol. 195, Jun. 2022, Art. no. 116550.
[19] J. Kaur and S. Singh, "Feature selection using mutual information and adaptive particle swarm optimization for image steganalysis," in *Proc. 7th Int. Conf. Rel., INFOCOM Technol. Optim. (Trends Future Directions) (ICRITO)*, Aug. 2018, pp. 538–544, doi: 10.1109/ICRITO.2018.8748522.
[20] Z. Ye, Y. Xu, Q. He, M. Wang, W. Bai, and H. Xiao, "Feature selection based on adaptive particle swarm optimization with leadership learning," *Comput. Intell. Neurosci.*, vol. 2022, pp. 1–18, Aug. 2022, doi: 10.1155/2022/1825341.
[21] Y. Hu, Y. Zhang, X. Gao, D. Gong, X. Song, Y. Guo, and J. Wang, "A federated feature selection algorithm based on particle swarm optimization under privacy protection," *Knowl.-Based Syst.*, vol. 260, Jan. 2023, Art. no. 110122.
[22] X. Li and J. Ren, "MICQ-IPSO: An effective two-stage hybrid feature selection algorithm for high-dimensional data," *Neurocomputing*, vol. 501, pp. 328–342, Aug. 2022.
[23] X. Han, "On the understanding and application of Newton's third law," *Sci. Consulting J.*, vol. 11, no. 1, pp. 102–103, Jan. 2017.
[24] L. Zheng and X. Li, "Teaching research of mathematica assisted exploration of physics and mechanics problems in high school," *Phys. Bull.*, vol. 7, no. 5, pp. 105–109, Apr. 2023.
[25] L. Li, P. Ma, and L. Liang, "Application of phase diagram method of elastic collision double conservation equation in physics competition," *Phys. Teach.*, vol. 44, no. 9, pp. 67–72, Sep. 2022.
[26] S. Ma, "Collision-like collision in the law of conservation of momentum," *Teach. Examination*, no. 13, pp. 37–43, Mar. 2023.
[27] C. Deng, "Research on feature selection of mutual information in Chinese text classification," *Southwest Univ.*, no. 9, Apr. 2011.
[28] X. Li, J. Chong, Y. Lu, and Z. Li, "Application of information gain in the selection of factors for regional slope stability evaluation," *Bull. Eng. Geol. Environ.*, vol. 81, no. 11, Oct. 2022, Art. no. 470.
[29] I. Rodríguez-Luján, R. Huerta, C. P. Elkan, and C. S. Cruz, "Quadratic programming feature selection," *J. Mach. Learn. Res.*, vol. 11, pp. 1491–1516, Mar. 2010.
[30] E. J. G. Pitman, "Significance tests which may be applied to samples from any populations. II. The correlation coefficient test," *J. Roy. Stat. Soc. B, Stat. Methodol.*, vol. 4, no. 2, pp. 225–232, Jul. 1937.
[31] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proc. Int. Conf. Neural Netw. (ICNN)*, vol. 4, Dec. 1995, pp. 1942–1948.
[32] X. Liu, G.-G. Wang, and L. Wang, "LSFQPSO: Quantum particle swarm optimization with optimal guided Lévy flight and straight flight for solving optimization problems," *Eng. with Comput.*, vol. 38, no. S5, pp. 4651–4682, Dec. 2022.
[33] Y. Zheng, Y. Li, G. Wang, Y. Chen, Q. Xu, J. Fan, and X. Cui, "A novel hybrid algorithm for feature selection based on whale optimization algorithm," *IEEE Access*, vol. 7, pp. 14908–14923, 2019.
[34] Y. Zheng, Y. Li, G. Wang, Y. Chen, Q. Xu, J. Fan, and X. Cui, "A novel hybrid algorithm for feature selection," *Pers. Ubiquitous Comput.*, vol. 22, pp. 971–985, May 2018.
[35] Z. Qu and J. Yin, "Optimized LSTM networks with improved PSO for the teaching quality evaluation model of physical education," *Int. Trans. Electr. Energy Syst.*, vol. 2022, pp. 1–12, Sep. 2022, doi: 10.1155/2022/8743694.
[36] Y. Zheng, Y. Li, G. Wang, Y. Chen, Q. Xu, J. Fan, and X. Cui, "A hybrid feature selection algorithm for microarray data," *J. Supercomput.*, vol. 76, no. 5, pp. 3494–3526, May 2020.
[37] D. J. MacKay, *Information Theory, Inference and Learning Algorithms*. Cambridge, U.K.: Cambridge Univ. Press, 2003.
[38] P. Latham and Y. Roudi, "Mutual information," *Scholarpedia*, vol. 4, no. 1, p. 1658, 2009, doi: 10.4249/scholarpedia.1658.
[39] L. Yin, D. Li, and J. Xu, "Support vector machine was optimized based on particle swarm algorithm of video flame detection," *China New Technol. New Products*, no. 13, pp. 146–148, Jul. 2023, doi: 10.13612/j.cnki.cntp.2023.13.004.

XIAOTONG BAI was born in Hebei, China, in 1999. She received the B.S. degree in software engineering from the Boda College, Jilin Normal University, in 2022, where she is currently pursuing the M.S. degree.

Her research interest includes feature selection.

YUEFENG ZHENG received the B.S. degree in computer science and technology and the M.S. degree in computer application from Jilin Normal University and the Ph.D. degree from Jilin University.

He is currently an Associate Professor with the School of Mathematics and Computer Science, Jilin Normal University. His research interests include feature selection and machine learning.

YANG LU received the B.S. degree in computer science and technology from Jilin Normal University, the M.S. degree in computer application from Jilin University, and the Ph.D. degree from Jiangsu University.

She is currently a Professor with the School of Mathematics and Computer Science, Jilin Normal University. Her research interests include image manipulation and machine learning.