A Sparse Genetic Algorithm to Solve Feature Selection of Sparse High-dimensional Data and Liver Totxicity Classification

Yu Liu, Jie-Sheng Wang*, Jia-Yao Wen, Yu-Tong Li, Peng-Guo Yan

Abstract—**The feature selection (FS) problem in real-life applications across various industries, particularly in the medical field, is a commonly encountered challenge. An improved Sparse Genetic Algorithm (SparseGA) was proposed to combine a sparse strategy with GA. SparseGA incorporates the genetic architecture while improving the greedy initialization strategy, dynamic scoring and elite retention strategy. Additionally, the K-nearest neighbors (KNN) classifier is integrated to select a representative feature subset from the given sparse high-dimensional dataset, improving both data representation ability and classification performance. Experimental results demonstrate that this method has achieved remarkable performance enhancements on various UCI sparse high-dimensional datasets and generally performs well across most datasets. Furthermore, a collection of medical data consisting of 1475 hepatotoxic compounds and 1038 non-hepatotoxic compounds was collected for classification purposes. A comparison of performance with commonly used classical meta-heuristic algorithms reveals that SparseGA performs favorably on the classification problems. The classification accuracy increased from 68.13% to 72.33%, showing the practicality and generalization ability of the algorithm. The findings of this research can assist in predicting the hepatotoxicity of compounds in the drug development stage.**

Index Terms—**feature selection, sparse genetic algorithm, liver toxicity, classification**

I. Introduction

In the real world, an increasing number of practical application scenarios involve the processing of sparse high-dimensional data, such as word frequency statistics in text data, drug molecules and gene expression data in genomics. Therefore, choosing the most representative subset of features from this data can reduce the detrimental effects of dimension curse and improve the expressive ability and classification performance. Meta-heuristic optimization (MHO) has demonstrated impressive and precise performance by stochastically solving optimization problems based on prior knowledge of stochastic search over the past few decades. These algorithms include Gray Wolf Optimizer (GWO) [1], Whale Optimization Algorithm (WOA) [2], Salp Swarm Algorithm (SSA) [3] and Pathfinder Algorithm (PFA) [4]. Striking a balance between exploring the search space and developing optimal solutions can lead to improved algorithm performance. Emina et al. Utilized GAs to extract crucial features from breast cancer datasets and employed various data mining techniques for classification decisions [5]. Huang et al. aimed to improve classification accuracy by simultaneously optimizing parameters and feature subsets through a GA-based method [6]. Stefano et al. proposed a FS algorithm based on GAs to evaluate feature subsets through a designed separability index. [7]. From different perspectives, Li et al. analyzed the differences and connections between individual sparse FS. Furthermore, they explored promising research directions and topics in sparse learning models related to FS [8]. Hou et al. proposed a novel unsupervised FS framework, that is Joint Embedding Learning and Sparse Regression (JELSR) [9]. Yang et al. studied the role of FS in face recognition from the perspective of sparse representation[10]. Maleki et al. adopted KNN technology, used GAs for FS, and experimentally determined the optimal value of K for diagnosing the patient's disease stage [11]. Abualigah et al. proposed a hybrid FS method based on the SCA and GA [12]. Singh et al. adopts Gravitational Search Optimization Algorithm (GSOA) for FS in machine learning. The focus of this study is to analyze retinal fundus images to extract 36 features of infected and healthy individuals. By training six machine learning models with the feature subset identified based on GSOA, the classification accuracy is 95.36% [13-14].

Drug development is a time-consuming and expensive process. On average, the development of a new drug takes 10 to 17 years and incurs a cost of approximately US$2.6 billion [15]. When a new drug enters the market, it can cause adverse drug reactions, potentially leading to restricted use or even withdrawal [16]. Therefore, ensuring

Manuscript received September 30, 2024; revised January 24, 2025. This work was supported by the Basic Scientific Research Project of Institution of Higher Learning of Liaoning Province (Grant No. LJ222410146054) and the Postgraduate Education Reform Project of Liaoning Province (Grant No. LNYJG2022137)..

Yu Liu is a Ph. D. student of School of Electronic and Information Engineering, University of Science and Technology Liaoning, Anshan, 114051, P. R. China (e-mail: lnasacl@126.com).

Jie-Sheng Wang is a professor of School of Electronic and Information Engineering, University of Science and Technology Liaoning, Anshan, 114051, P. R. China (Corresponding author, phone: 86-0412-2538246; fax: 86-0412-2538244; e-mail: wang_jiesheng@126.com).

Jia-Yao Wen is a postgraduate student of School of Electronic and Information Engineering, University of Science and Technology Liaoning, Anshan, 114051, P. R. China (e-mail: m17641243544@163.com).

Yu-Tong Li is a postgraduate student of School of Electronic and Information Engineering, University of Science and Technology Liaoning, Anshan, 114051, P. R. China (e-mail: 13942871038@163.com).

Peng-Guo Yan is a postgraduate student of School of Electronic and Information Engineering, University of Science and Technology Liaoning, Anshan, 114051, P. R. China (e-mail: yan407319226@163.com).

the safety of drug development while minimizing costs and time is crucial. Hepatotoxicity, a significant adverse drug reaction, contributes to the failure of clinical trials and the withdrawal of drugs from the market. Thus, predicting potential hepatotoxicity during the early stages of drug discovery is critical for minimizing costs and the likelihood of drug failure. However, current in vivo animal toxicity testing is both costly and time-consuming. As an alternative, several machine learning models have been developed to predict potential hepatotoxicity in humans [17]. Based on the detailed annotated drug labels of 387 FDA-approved drugs, Chen et al. constructed and validated quantitative structure-activity relationships for type 2 DILI prediction (no DILI and yes DILI) by using a decision forest algorithm and molecular descriptors calculated by Mold2 (QSAR) machine learning model [18]. Williams et al. [19] developed Bayesian machine learning to build model data by integrating mechanistically relevant liver safety assays, including data from in vitro assays. A recursive random forest method was developed to predict bioactivity data for 233 chemicals by using chemical descriptors and ToxCast to detect mouse liver toxicity [20].

This paper presents an improved algorithm, SparseGA, that combines sparse strategy and genetic algorithm to address the FS problem in high-dimensional sparse space. Additionally, the algorithm was tested on and compared to 15 data sets from the UCI. The results demonstrate the superiority of the proposed algorithm. Classification tasks utilize KNN to evaluate the quality of the selected feature subset. Finally, FS and classification were performed on the hepatotoxic compound data set.

II. Sparse Genetic Algorithm Based on Dynamic Scoring and Greedy Strategy

A. Dynamic Scoring Strategy

The SparseEA is an evolutionary algorithm developed for large-scale sparse multi-objective optimization problems (MOPs) [21]. However, there are two shortcomings, one is that it is only scored in the initialization stage, and the other is that the decision variables of each dimension are scored separately. So, in order to solve the above problems, to reduce the score of each decision variable has an impact on its optimization performance. Improved the static scoring strategy of SparseEA, proposed a dynamic scoring strategy for decision variables, and applied it to SparseGA. Algorithm 1 outlines the specific process. The real vector dec represents the value of each decision variable. Eq. (1) can be used to obtain each solution, denoted by x. Consequently, the best decision variables identified are stored in the dec vector. The mask vector records which decision variables should be set to 0, allowing for control of solution sparsity.

$$x = (mask_1 \times dec_1, \cdots, mask_D \times dec_D) \tag{1}$$

The algorithm takes as input the current population P and outputs a $1 \times D-$ dimensional matrix that represents the Scores of D decision variables. Score consists of two parts. The non-dominated level where the individual is located and the number of times the corresponding decision variable is selected in the population $Mask$ vector are weighted and accumulated. The lower the non-dominated level, carries a higher weight. In summary, for decision variable d, the higher its frequency in the population's the higher the score. The lower the non-dominated level it belongs to, the higher the score. The scores are updated as each generation of the population evolves.

For instance, consider a population with a size of $N = 5$ and a decision variable dimension of $D = 6$. The distribution of the population in the decision space and its non-dominated level are illustrated in Table I. According to Algorithm 1, the binary masking matrix $Mask$ of the population and the score of each indiviual, $ScoreP$, can be obtained as shown in Table II. Each decision variable's Score can be calculated by multiplying the binary value corresponding to its dimension in the $Mask$ matrix with the respective individual score.

The presented tables provide a comprehensive depiction of the decision variable (dec) matrix for a given population and the corresponding non- dominated levels assigned to each individual. Within the decision variable matrix, each row signifies an individual, with each column representing a specific decision variable. The matrix encompasses six columns to account for six decision variables. The decision variable values for each individual span the range from 0 to 1. Taking individual 1 as an exemplar, the decision variable values are as follows: 0.83 for the first variable, 0.67 for the second variable, 0 for the third and fourth variables, 0.92 for the fifth variable and 0 for the sixth variable. This representation is sequentially extended to encompass all individuals within the population. In addition to the decision variable matrix, each individual is ascribed a non-dominated level. In the exemplified scenario, individual 1 is positioned at non-dominated level 3, while individuals 2 and 5 share level 2. Individual 3 is designated at level 1, and individual 4 is situated at level 4. The non-dominated hierarchy serves as a crucial metric for elucidating the interrelations among individuals within the context of goal optimization. A lower hierarchy signifies superior individual performance, whereas identical hierarchies suggest a lack of discernible distinctions between individuals.

Algorithm 1 ***DyScoringStrategy(P)***

Input: P(population)

Output: $Score$ (score of each dimension of decision variable)

1: $D \leftarrow$ Decision variable dimension; //D is an integer

2: $N \leftarrow$ The size of the population P; //N is an integer

3: $Q \leftarrow$ Decision variable of population P; //Q is an $N \times D$ dimensional real matrix

4: $Mask \leftarrow Q \neq 0$; //$N \times D$ dimensional 01 matrix

5: $FrontNo \leftarrow$ Population P non-dominated sorting; //$1 \times N$ dimensional integer matrix, for each individual level

6: $ScoreP \leftarrow max(FrontNo)$ - $FrontNo$ + 1; //$1 \times N$ dimensional integer matrix, which is the score of each individual

7: $MaskScore \leftarrow Mask .\times repmat(ScoreP, [D\ 1])'$; //$N \times D$ dimensional matrix, which is the score of each dimension of each individual

8: $Score \leftarrow sum(MaskScore)$; //Sum by column, D-dimensional vector, score for each dimension of decision variables

9: **Return** $Score$; //Return the score of each dimension of decision variables

TABLE I. POPULATION DEC MATRIX AND INDIVIDUAL NON-DOMINATED HIERARCHY

	Dimension 1	Dimension 2	Dimension 3	Dimension 4	Dimension 5	Dimension 6	Non-dominated layer
Individual -1	0.83	0.67	0	0	0.92	0	3
Individual -2	0	0.36	0.54	0	0	0.21	2
Individual -3	0.56	0	0	0.14	0.24	0.85	4
Individual -4	0.96	0.27	0.71	0	0.27	0	1
Individual -5	0.23	0	0.36	0.25	0	0	2

TABLE II. POPULATION MASK MATRIX AND INDIVIDUAL SCORE

	Dimension 1	Dimension 2	Dimension 3	Dimension 4	Dimension 5	Dimension 6	Non-dominated layer
Individual -1	1	1	0	0	1	0	4-3+1=2
Individual -2	0	1	1	0	0	1	4-2+1=3
Individual -3	1	0	0	1	1	1	4-4+1=1
Individual -4	1	1	1	0	1	0	4-1+1=4
Individual -5	1	0	1	1	0	0	4-2+1=3
Decision variable score	2+0+1+4+3	2+3+0+4+0	0+3+0+4+3	0+0+1+0+3	2+0+1+4+0	0+3+1+0+0	

B. *Greedy Initialization Strategy*

The principle of greedy strategy initialization is to select the optimal individual according to the score and probability of the decision variable as much as possible during the population initialization stage. The greedy population initialization strategy includes two steps: decision variable scoring and population initialization. First, a population Q containing D individuals is generated, each individual contains a real vector Dec and a binary vector $Mark$. Through Algorithm 1, the score of each individual at the non-dominated level is calculated, which indicates the individual's strengths and weaknesses in the solution space. Since each dimension of the decision variable in population Q is marked as 1 an equal number of times in the $Mark$, the score of the decision variable here is only related to the non-dominated level at which the individual is located. The lower the level, the higher the score. The score of a decision variable represents the probability that it should be set to 1, with a higher score indicating a higher probability of the decision variable being set to 1. Following the evaluation of decision variables, the initialization phase of population P is each individual within P is endowed with a real-number vector, Dec, wherein each constituent is assigned a random value. Concurrently, an accompanying binary vector $Mark$, is established for every individual, with its elements initially set to 0. Subsequently, a stochastic process selects $rand()\times D$ elements randomly from the $Mark$ vector and assigns them a value of 1. Here, $rand()$ denotes a uniformly distributed random number within the interval [0,1]. It is imperative to note that the selection process, executed "with replacement", permits the possibility of duplicate elements being assigned a value of 1. Consequently, the resultant $Mark$ vector is characterized by a proportion of elements set to 1 that is anticipated to be less than 50%. A detailed procedural delineation is provided in Algorithm 2.

C. *Greedy Genetic Operator*

The pseudo-code for a greedy genetic operator is provided in Algorithm 3. P and q are randomly selected parents from the crossover pool, and each time a new offspring individual o is created. In the binary vector $Mark$ section, start by setting the $Mark$ of o to be the same as that of individual P, and then perform the following two operations with equal probability.

(1) Crossover operation. Based on the score of the decision variable (the smaller the better), choose one of the non-zero elements in $p.mask \cap \overline{q.mask}$ and set the element at this position in $o.mark$ to 0; Or, according to the score of the decision variable Score (bigger is better), select one of the non-zero elements of $p.mask \cap \overline{q.mask}$, and assign the element at this position in $o.mark$ to 1.

(2) Mutation operation. In step 1, Choose one of the non-zero elements in $o.mark$ and assign a value of 0 to the element in $o.mark$ at that position; Conversely, based on the score of the decision variable (with a larger value being more preferable), select one of the elements in $o.mark$ with a value of 0, and assign a value of 1 to the element in $o.mark$ at that position. The real vector dec of O is generated by using conventional genetic operations, specifically, simulated binary crossover and polynomial mutation.

Algorithm 2 ***InitializationGreedy(N)***

Input: N (population size)
Output: P (initial population), *Score* (score of each dimension of decision variables)

```
D ← decision variable dimension;
Dec ← D×D dimensional random real matrix;
Mark ← D-dimensional identity matrix;
Q ← SOLUTION(Dec.×Mask); //Construct a population for scoring
Score ← DyScoringStrategy(Q); //Calculate the score of each dimension of decision variables, Algorithm 1
Dec ← N × D random real matrix;
Mark ← N × D-dimensional all-zero matrix;
For i = 1 : N
    For j = 1 : D × rand() //The mathematical expectation of D × rand() is 0.5D
        x, y ←Randomly select two integers from [1, D];
        If Score[x] > Score[y]   Mark[i][x] = 1;
        Else                     Mark[i][y] = 1;
P ← SOLUTION (Dec.× Mask); //Initialization population
Return P, Score;
```

The greedy-like genetic operator is specifically designed for large-scale sparse MOPs. Based on the score of the decision variable, both non-zero elements and 0 in the mask vector will be flipped with equal probability. The offspring generated by this operator exhibit varying counts of 0 and 1 elements, to maintain the sparsity of decision variables in the offspring individuals.

D. SparseGA Algorithm Framework

SparseGA is proposed by improving the GA framework and combining the decision variable dynamic scoring strategy, greedy initialization strategy and greedy genetic operator to generate the initial population and crossover mutation offspring. It is a GA variant applied to FS problems. The core idea is to use strategies to process sparse high-dimensional data through the evolution process of the GA, and gradually optimize the quality of the feature subset to enhance the classification performance of the data. Firstly, initialize the population P with size N, and construct a suitable reference point based on the population size and target dimensions. In each iteration of the main loop, 2^N parents through binary tournaments to generate N offspring. Next, the parent population and the offspring population are merged, and duplicate solutions in the combined population are deleted. In the combined population, N individuals are selected and retained for the next generation through non-dominated sorting and by using associated reference points. Detailed steps can be found in Algorithm 4.

Fig. 1 presents the flowchart of SpaeseGA for the feature selection problem. Firstly, the dataset is read and divided into a training set and a test set. Subsequently, a binarization operation is implemented for the progeny individuals. Specifically, according to the pre-set threshold, when the importance index of a feature in the model training and evaluation process is higher than the threshold, its corresponding code is assigned to 1, which indicates that the feature is selected. On the contrary, if it is lower than the threshold, it is assigned to 0, which indicates that the feature is not selected. With this binarization, it is possible to clearly define the subset of features that are ultimately identified for each child individual. After training the KNN classifier based on the selected K-values by using the training set, the relevant evaluation metrics on the training set are calculated, such as accuracy, precision and recall in the classification task, to initially judge the model's goodness of fit to the training data. Next, test the KNN classifier and calculate the corresponding evaluation metrics.

Algorithm 3 *GAOperatorGreedy(P, Score)*

Input: P (crossover pooling), *Score* (score of each dimension of the decision variable)
Output: O (offspring population)
$O \leftarrow \emptyset$ //offspring population
While $P \neq \emptyset$
  $p, q \leftarrow$ Randomly select two individuals from P; //Determine the male and female parents
  $P \leftarrow P\backslash\{p, q\}$; // Eliminate p and q from P
  o.mask ← *p.mask*; //01 vector
  If *rand*() < 0.5 //cross operation
    x,y ← Randomly select two decision variable numbers from the non-0 elements of $p.mask \cap \overline{q.mask}$;
    If *Score*[x] > *Score*[y] *o.mask*[y] = 0;
    Else *o.mask*[x] = 0;
  Else
    x,y ← Randomly select two decision variable numbers from $p.mask \cap \overline{q.mask}$ 0 elements;
    If *Score*[x] > *Score*[y] *o.mask*[x] = 1;
    Else *o.mask*[y] = 1;
  If *rand*() < 0.5 //Mutation operation
    x, y ← Randomly select two decision variable numbers from the non-0 elements of *o.mask*;
    If *Score*[x] > *Score*[y] *o.mask*[y] = 0;
    Else *o.mask*[x] = 0;
  Else
    x, y ← Randomly select two decision variable numbers from elements where *o.mask* is 0;
    If *Score*[x] > *Score*[y] *o.mask*[x] = 1;
    Else *o.mask*[y] = 1;
  o.dec ← *GaOperator*(*p.dec*, *q.dec*); //The real number vector part uses traditional crossover mutation
  $O \leftarrow O \cup \{o\}$;
Return O;

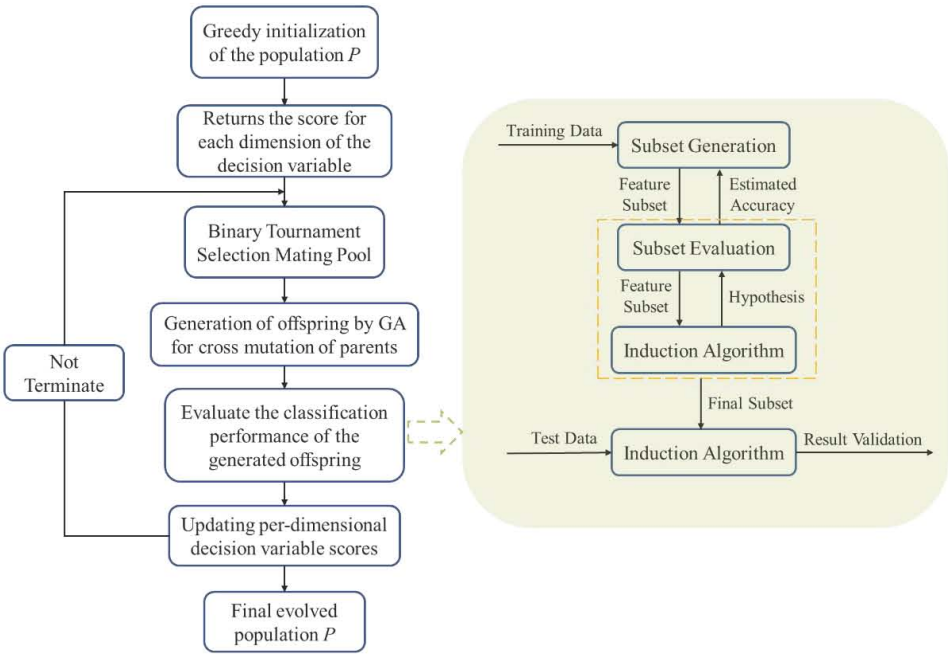


Fig. 1 Flowchart of SparseGA in feature selection problem.

III. Experimental Results and Discussion

A. Classification Dataset Description

This section evaluates and compares the proposed SparseGA with other algorithms, and provides an experimental results analysis. This article selected 15 benchmark data sets from the UCI database. The details of these datasets are presented in Table III.

B. K-Nearest Neighbor Classifier

This article utilizes the KNN classifier [22] to compute the Euclidean distance (D_E) between an unlabeled instance and its closest K instances. By doing so, it determines the K nearest neighbors of the sample for classification purposes. If the majority of the K most similar samples of a sample belong to a particular category in the given feature space, then the sample also belongs to that category. Its calculation formula is presented as:

$$D_E = \sqrt{\sum_{i=1}^{k} (Train_{F_i} - Test_{F_i})^2} \tag{2}$$

where, $Train_{F_i}$ is a feature in the training data, $Test_{F_i}$ is a feature in the test data, and k is the number of features.

C. Fitness Function

The objective of FS methods is to minimize the number of selected features while maintaining high classification accuracy. To address this conflicting goal, fitness function shown in Eq. (3) are employed.

$$fitness = h_1 \gamma_R(D) + h_2 \frac{|M|}{|N|} \tag{3}$$

where, r_R represents the classification error rate corresponding to the currently selected feature subset of the classifier, $|M|$ represents the number of currently selected features, $|N|$ is the total number of features, h_1 and h_2 are two weight coefficients of subset classification rate and length satisfy $h_1 + h_2 = 1$. In this article, h_1 and h_2 are set to 0.9 and 0.1, respectively.

D. Experimental Parameter Settings

This paper uses KNN with Euclidean distance and $k = 5$ to calculate the classification accuracy in the fitness function. The experiment is conducted 10 times with different random seeds for robustness. Additionally, to prevent over-fitting, five-fold cross-validation is employed to evaluate the effectiveness of the machine learning model. The final results are obtained by collecting the average statistical measurements from 10 independent runs. The population size of each algorithm is set to 30, the maximum number of iterations to 100, and the dimension of the search space is equal to the total number of features. Meanwhile, the detailed parameter settings of all algorithms are presented in Table IV.

The crossover probability is generally chosen between 0.6 and 0.9, and the mutation probability is chosen between 0.001 and 0.1. In this paper, a crossover probability of 0.6 and a mutation probability of 0.01 are chosen. Firstly, selecting a higher crossover probability of 0.6 helps to generate new individuals in the population, increasing the diversity of the search and avoiding falling into local optimal solutions. However, if the crossover probability is too high, it may lead to the emergence of too many duplicate individuals, which will reduce the efficiency of the search and cause overcrowding in the survival space. Secondly, choosing a lower mutation probability of 0.01 helps to maintain the overall integrity of the population so as to avoid excessive disruption of the population structure. However, if the mutation probability is too high, it may disrupt the overall integrity of the population and increase the burden of the search. Therefore, by selecting a crossover probability of 0.6 and a mutation probability of 0.01, it is possible to fully utilize the crossover operation to generate new individuals during the search process, while maintaining the diversity of the population, thereby improving the convergence and search capabilities.

E. FS Performance Evaluation Criteria

The evaluation metrics for FS problems include fitness value, classification accuracy and average selection size. Eq. (4)-(8) represent the calculation methods for the average classification accuracy, the average number of selected features, the fitness mean and the standard deviation respectively.

Table III. Datasets used in the simulation experiments

Number	Datasets	Features	Instances	Classes
1	Arrhythmia	279	452	16
2	COIL20	1024	1440	20
3	CNAE_9	856	1080	9
4	Hill_Valley	100	1212	2
5	Secom	590	1567	2
6	Handwritten	256	1593	10
7	QSAR_androgen_receptor	1024	1687	2
8	Har	561	270	6
9	HAPTDataSet	561	360	12
10	Isolet5	617	1559	26
11	Semeion	256	1593	10
12	UJIIndoorLoc	522	279	3
13	Madelon	500	2600	2
14	Mfeat	649	2000	10
15	TUANDROMD	241	4464	2

Table IV. Algorithm parameters

Algorithm	Parameters	Values
SparseGA	CrossoverRate, mutationRate	0.6, 0.01
SA	Temperature, MarkovChain, AttenuationFactor, mutationRate	100 * dim, 5, 0.98, 0.01
GA	CrossoverRate, mutationRate	0.6, 0.01
SSA	Sparrow number	10
WOA	r	1
PFA	Population number	10
GWO	a	[2,0]
SSOA	w c1 c2 k	0.7 2 2 0.5

$$Mean\_accuracy = \frac{1}{10}\sum_{i=1}^{10} Accuracy_i \quad (4)$$

where, $Mean\_accuracy_i$ represents the average classification accuracy achieved by independently executing the algorithm 10 times, while $Accuracy_i$ denotes the classification accuracy attained in each iteration. The classification accuracy is calculated as follows:

$$Accuracy = \frac{1}{10}\sum_{i=1}^{10} match(Pl_i, Al_i) \quad (5)$$

where, N represents the number of test set points, that is, the number of data set instances; Pl_i represents the predicted class label of data point i, Al_i refers to the actual class in the annotated data. In other words, it represents the reference class label of i, match (Pl_i , Al_i) serves as a discriminant function for comparison. When $Pl_i == Al_i$, match (Pl_i , Al_i) =1, otherwise match (Pl_i , Al_i)=0.

$$Mean\_feature = \frac{1}{10}\sum_{i=1}^{10} feature_i \quad (6)$$

where, $Mean\_feature$ represents the average number of selected features obtained by running the algorithm M times independently, while $feature_i$ denotes the number of selected features in each run.

$$Mean\_fitness = \frac{1}{10}\sum_{i=1}^{10} fitness_i \quad (7)$$

where, $Mean\_fitness$ represents the average fitness value obtained by running the algorithm independently M times, and f_i signifies the optimal fitness value obtained by each operation.

$$Std_{fitness} = \sqrt{\frac{1}{10}\sum (fitness_i - Mean\_fitness)^2} \quad (8)$$

where, $std_{fitness}$ represents the standard deviation of the fitness value, $fitness_i$ corresponds to the fitness value obtained at the ith time, and $Mean\_fitness$ is calculated using Eq. (6). Furthermore, all models are evaluated by calculating true positives (*TP*), true negatives (*TN*), false positives (*FP*) and false negatives (*FN*) to obtain Precision, Sensitivity, Specificity, F1-score, and G-mean values. The evaluation indicators are defined in Table V.

F. Simulation Results Comparison

In order to verify the improvement effect of the proposed strategy on SparseGA, the contribution comparison of different strategies is tested, as shown in Fig. 2. When only adding greedy initialization strategy and tournament strategy to the algorithm, it cannot achieve optimal results, and dynamic scoring will have a small effect compared to the original algorithm. But when these three strategies are combined, they are initialized through Initialization Greedy, the genetic operator is combined with the binary tournament to generate offspring, and finally the scores of each dimension of the decision variables are updated through dynamic scoring. By conducting repeated iterations in sequence, SparseGA achieved excellent classification accuracy in the model.

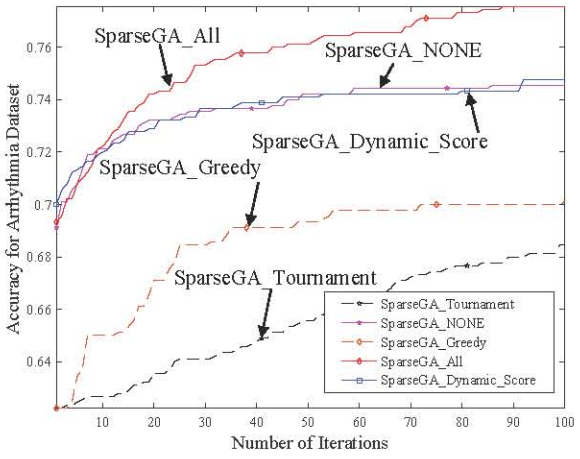
The convergence curves of classification accuracy of SparseGA and comparison algorithms on the UCI datasets are shown in Fig. 3, and the non-parametric Wilcoxon test results of classification performance indicators and classification accuracy are shown in Table VI-XVI. The time complexity is shown in Table 16. As can be seen from Table VI, compared with other algorithms, SparseGA has higher accuracy in classification problems. Although GOA performed well on the Hill_Valley dataset, Secom dataset and Madelon dataset, SparseGA achieved higher accuracy than other 8 comparison algorithms on the remaining 13 datasets, and reached 100% accuracy on the UJI-IndoorLoc dataset. When comparing the Precision, Sensitivity, Specificity, F1-score and G-mean metrics, SparseGA performed well in most data sets, indicating that the algorithm has good model accuracy. In terms of FS, although SparseGA does not select the minimum number of features compared with other 6 algorithms, it retains about half of the features compared with the original data set while ensuring the accuracy of classification.

The Wilcoxon test is a non-parametric statistical test method used to compare the differences between two related samples or paired samples. Based on the Wilcoxon analysis results of classification accuracy in Table XV, the following conclusions can be drawn. For the arrhythmia dataset, COIL20 dataset, and Semeion dataset, GA, PFA and SSOA do not meet the significance level of less than 0.05. However, for the QSAR_androgen_receptor dataset, only the p-value compared with SA is greater than 0.05, while all other comparison p-values are less than 0.05, indicating that SparseGA exhibits excellent performance in terms of classification accuracy. Overall, SparseGA has demonstrated excellent performance in handling these classification datasets.

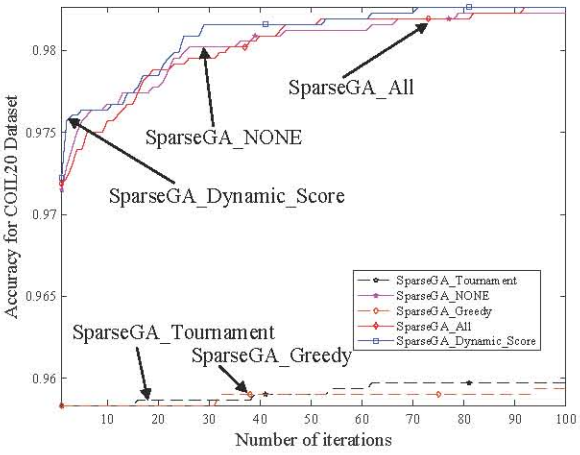
According to the time complexity analysis results in Table XVI, although SparseGA is not the fastest in the algorithm comparison, it performs well in terms of speed. For smaller data sets, the calculation can usually be completed in tens to tens of seconds, while on large data sets, it is usually around 150 seconds. On the COIL20 and Hill_Valley data sets, SparseGA has significant performance improvements compared with the original GA, and can achieve ideal classification accuracy on most data sets. Compared with the SSOA, although SSOA has superior performance in terms of time complexity, it is slightly insufficient in classification effect, which may be due to insufficient search due to too fast running speed. In comparison, SparseGA can still achieve excellent classification results while running quickly, showing excellent performance.

TABLE V. TEST PERFORMANCE INDEX

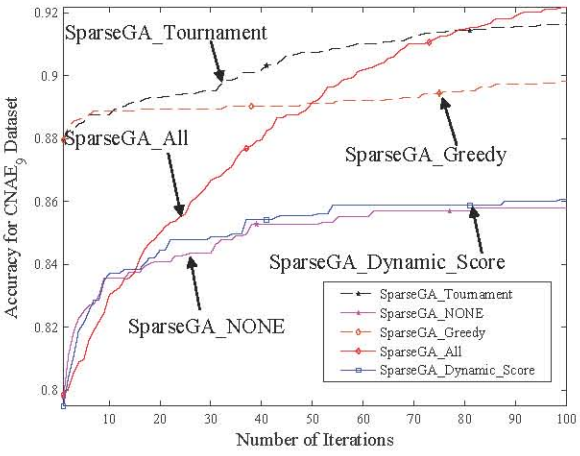
Measure	Definition
Accuracy	$(TN + TP)/(TP + FP + TN + FN)$
Sensitivity	$TP/(FN + TP)$
Specificity	$TN/(FP + TN)$
Precision	$TP/(TP + FP)$
F1-score	$2*(\frac{Precision*Senstivity}{Precision+Senstivity})$
G-mean	$\sqrt{Sensitivity*Specificity}$


SparseGA_All
SparseGA_NONE
SparseGA_Dynamic_Score
SparseGA_Greedy
SparseGA_Tournament
Accuracy for Arrhythmia Dataset
Number of Iterations
SparseGA_Tournament
SparseGA_NONE
SparseGA_Greedy
SparseGA_All
SparseGA_Dynamic_Score

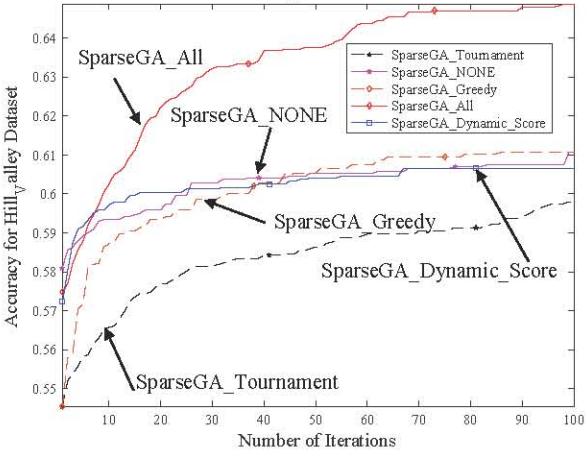
(1)


SparseGA_All
SparseGA_NONE
SparseGA_Dynamic_Score
SparseGA_Tournament
SparseGA_Greedy
Accuracy for COIL20 Dataset
Number of iterations
SparseGA_Tournament
SparseGA_NONE
SparseGA_Greedy
SparseGA_All
SparseGA_Dynamic_Score

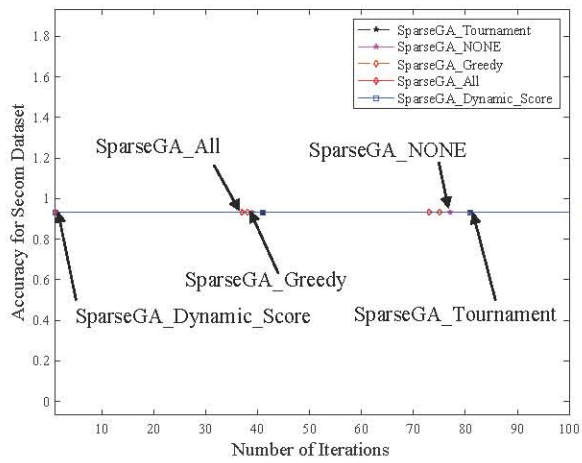
(2)


SparseGA_Tournament
SparseGA_All
SparseGA_Greedy
SparseGA_Dynamic_Score
SparseGA_NONE
Accuracy for CNAE$_9$ Dataset
Number of Iterations
SparseGA_Tournament
SparseGA_NONE
SparseGA_Greedy
SparseGA_All
SparseGA_Dynamic_Score

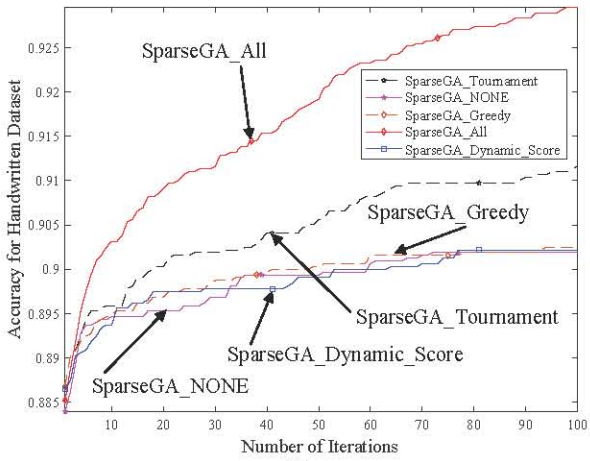
(3)


SparseGA_All
SparseGA_NONE
SparseGA_Greedy
SparseGA_Dynamic_Score
SparseGA_Tournament
Accuracy for Hill$_V$alley Dataset
Number of Iterations
SparseGA_Tournament
SparseGA_NONE
SparseGA_Greedy
SparseGA_All
SparseGA_Dynamic_Score

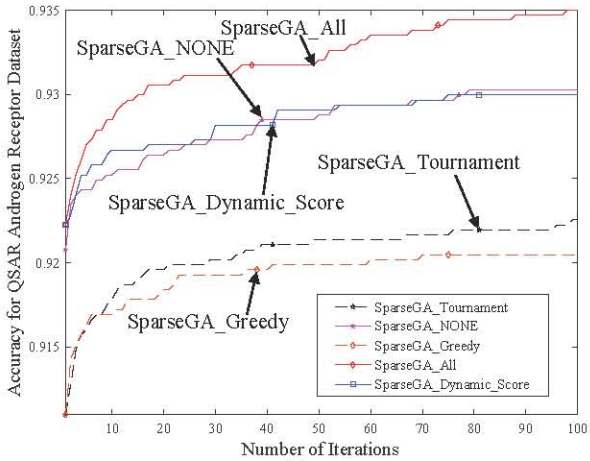
(4)


SparseGA_Tournament
SparseGA_NONE
SparseGA_Greedy
SparseGA_All
SparseGA_Dynamic_Score
SparseGA_All
SparseGA_NONE
SparseGA_Greedy
SparseGA_Dynamic_Score
SparseGA_Tournament
Accuracy for Secom Dataset
Number of Iterations

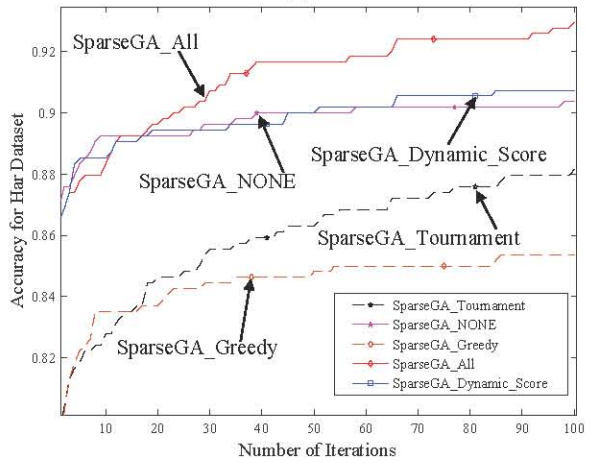
(5)


SparseGA_All
SparseGA_Greedy
SparseGA_Tournament
SparseGA_Dynamic_Score
SparseGA_NONE
Accuracy for Handwritten Dataset
Number of Iterations
SparseGA_Tournament
SparseGA_NONE
SparseGA_Greedy
SparseGA_All
SparseGA_Dynamic_Score

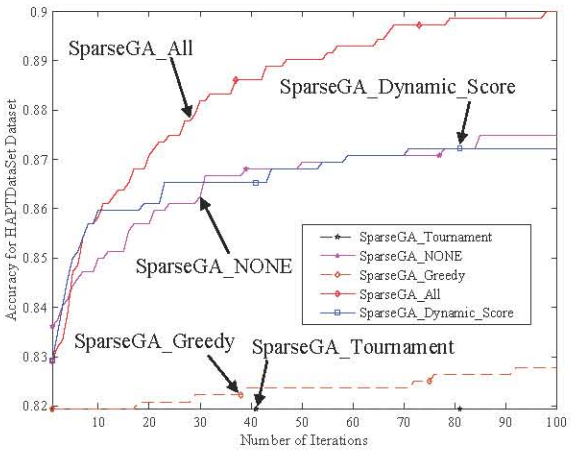
(6)


SparseGA_All
SparseGA_NONE
SparseGA_Tournament
SparseGA_Dynamic_Score
SparseGA_Greedy
Accuracy for QSAR Androgen Receptor Dataset
Number of Iterations
SparseGA_Tournament
SparseGA_NONE
SparseGA_Greedy
SparseGA_All
SparseGA_Dynamic_Score

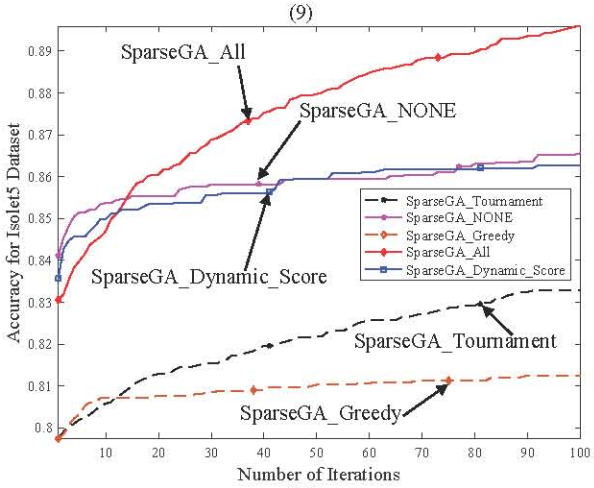
(7)


SparseGA_All
SparseGA_NONE
SparseGA_Dynamic_Score
SparseGA_Tournament
SparseGA_Greedy
Accuracy for Har Dataset
Number of Iterations
SparseGA_Tournament
SparseGA_NONE
SparseGA_Greedy
SparseGA_All
SparseGA_Dynamic_Score

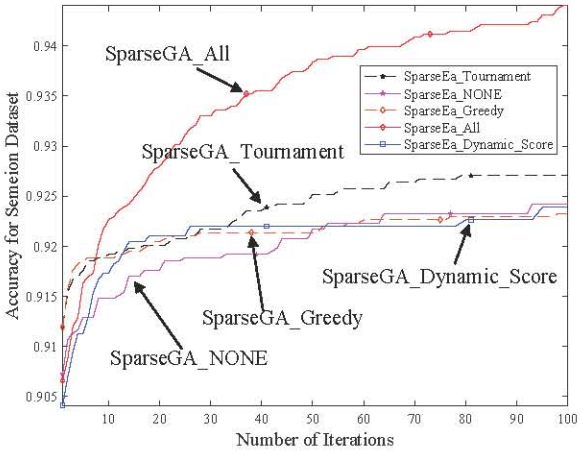
(8)



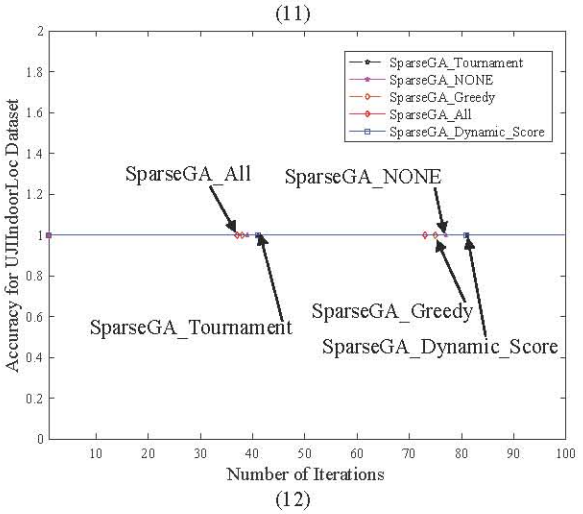
(9)



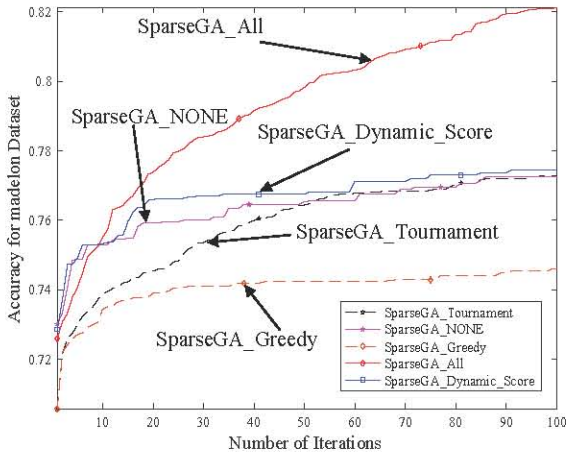
(10)



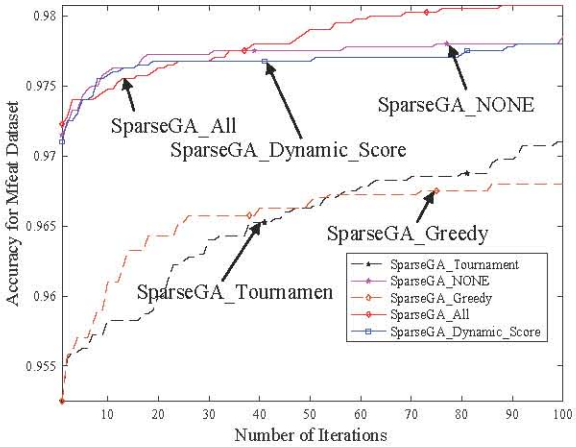
(11)



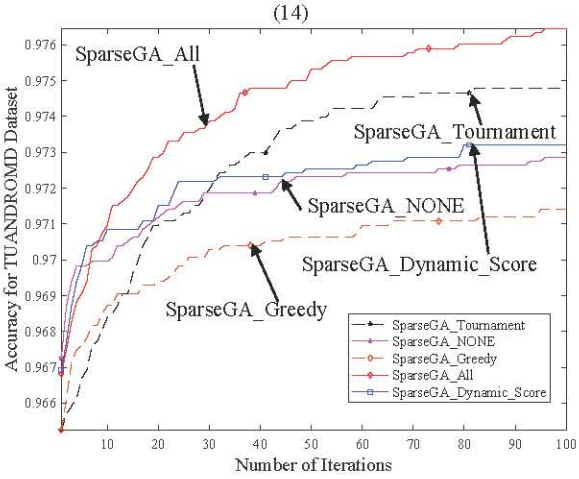
(12)



(13)

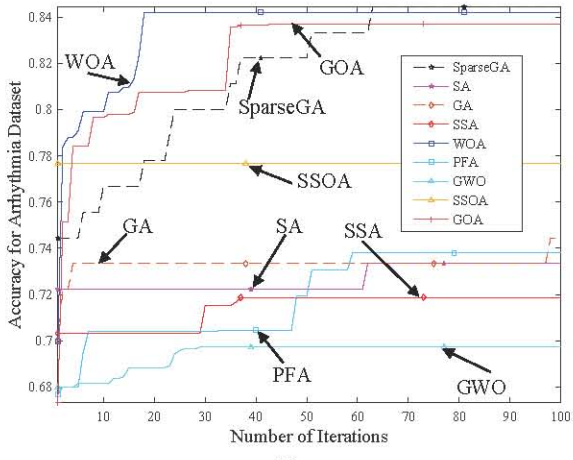


(14)

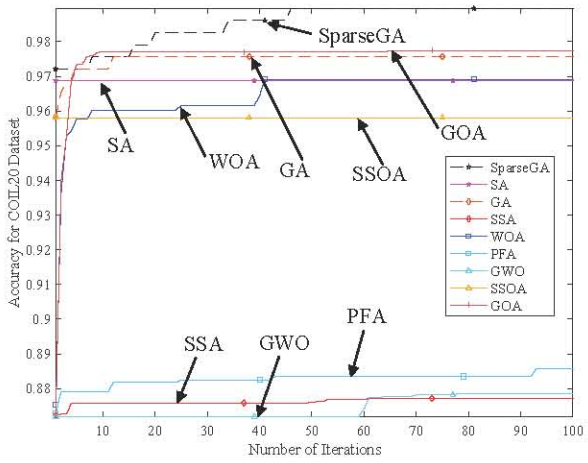


(15)

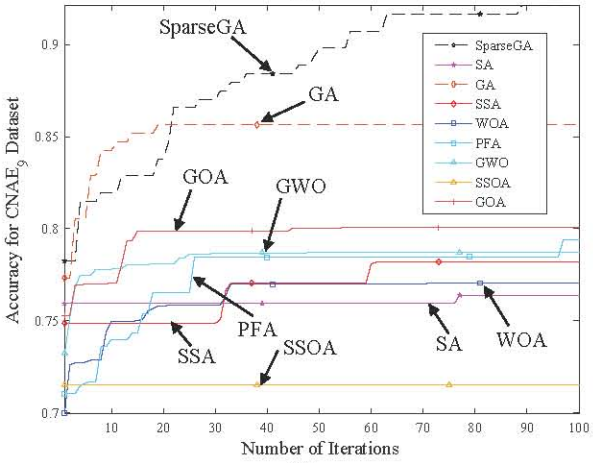
Fig. 2 Comparison of classification accuracy of different strategies.



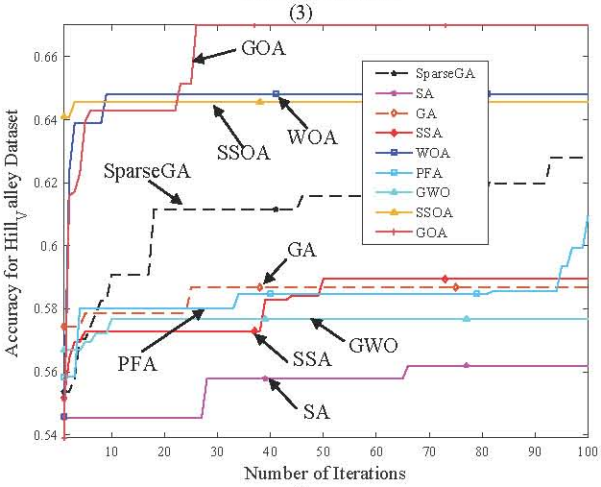
(1)


SparseGA
SA
WOA
GA
SSOA
GOA
SSA
GWO
PFA
Accuracy for COIL20 Dataset
Number of Iterations

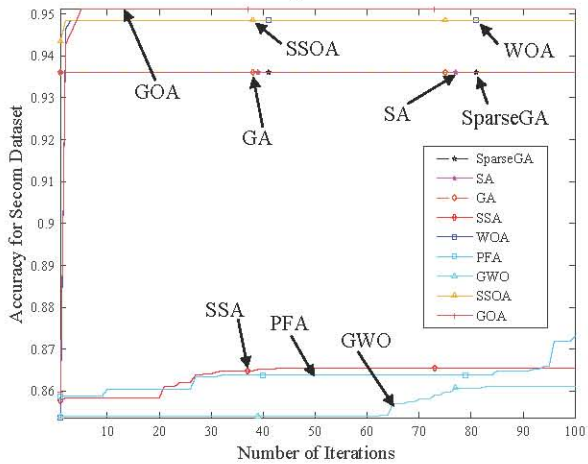
(2)


SparseGA
GA
GOA
GWO
PFA
SSA
SSOA
SA
WOA
Accuracy for CNAE$_9$ Dataset
Number of Iterations

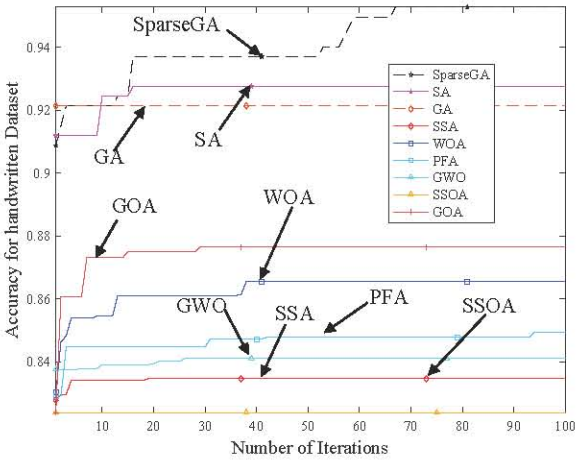
(3)


GOA
SSOA
WOA
SparseGA
GA
PFA
SSA
GWO
SA
Accuracy for Hill$_y$alley Dataset
Number of Iterations

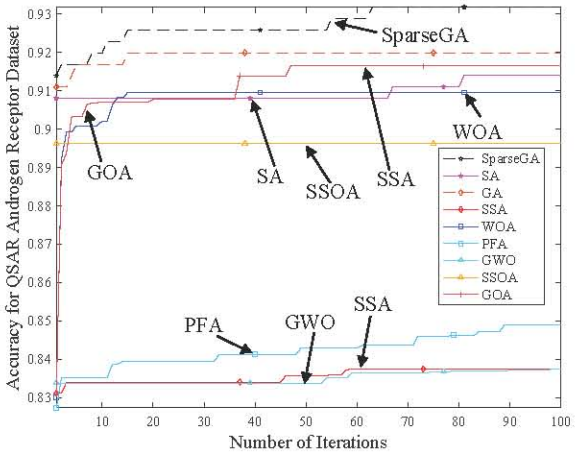
(4)


SSOA
WOA
GOA
SA
SparseGA
GA
SSA
PFA
GWO
Accuracy for Secom Dataset
Number of Iterations

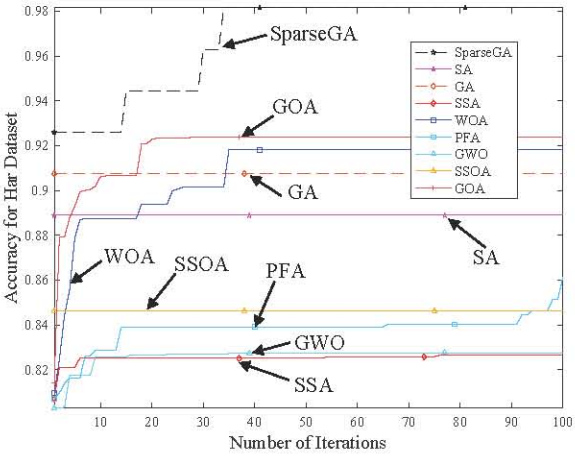
(5)


SparseGA
GA
SA
GOA
WOA
GWO
SSA
PFA
SSOA
Accuracy for handwritten Dataset
Number of Iterations

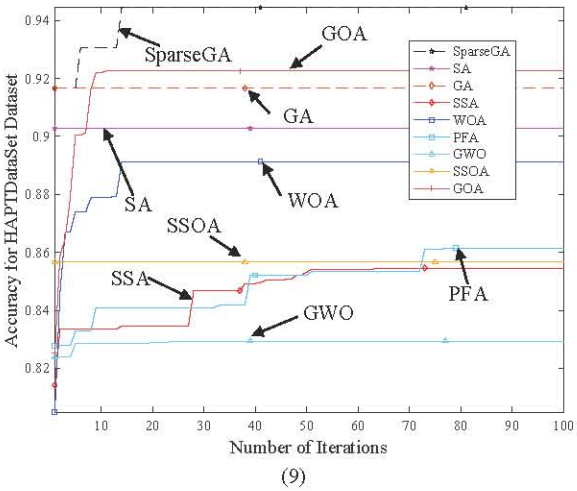
(6)


SparseGA
GOA
WOA
SA
SSOA
SSA
PFA
GWO
Accuracy for QSAR Androgen Receptor Dataset
Number of Iterations

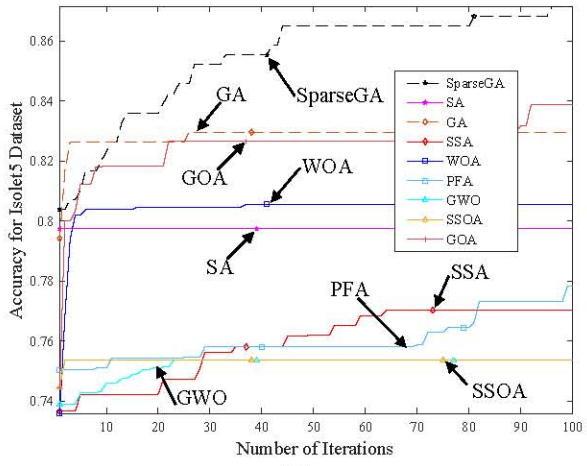
(7)


SparseGA
GOA
GA
SA
WOA
SSOA
PFA
GWO
SSA
Accuracy for Har Dataset
Number of Iterations

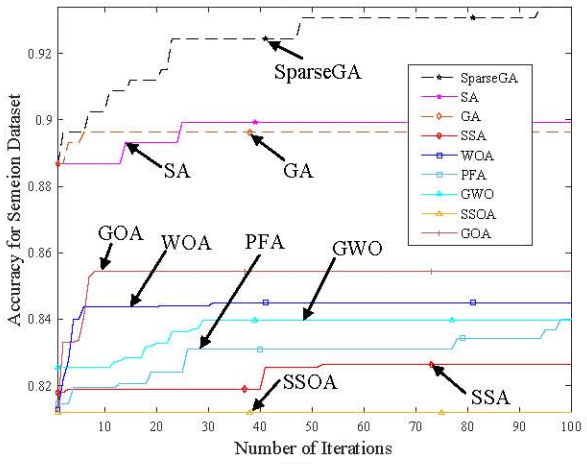
(8)


SparseGA
GOA
GA
SA
WOA
SSOA
PFA
GWO
SSA
Accuracy for HAPTDataSet Dataset
Number of Iterations

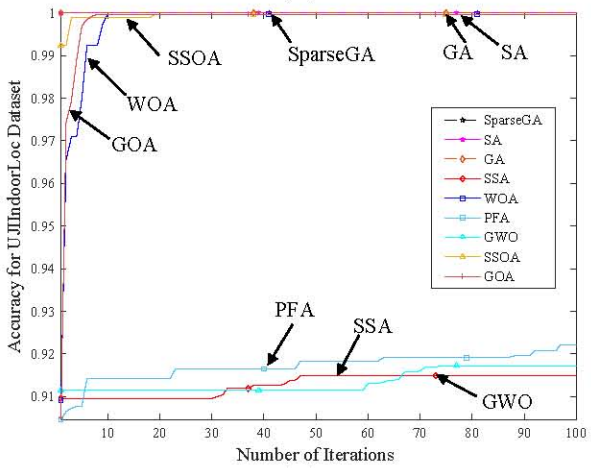
(9)



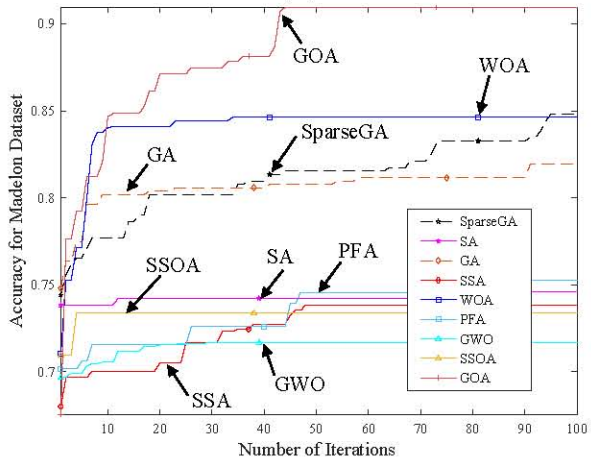
(10)



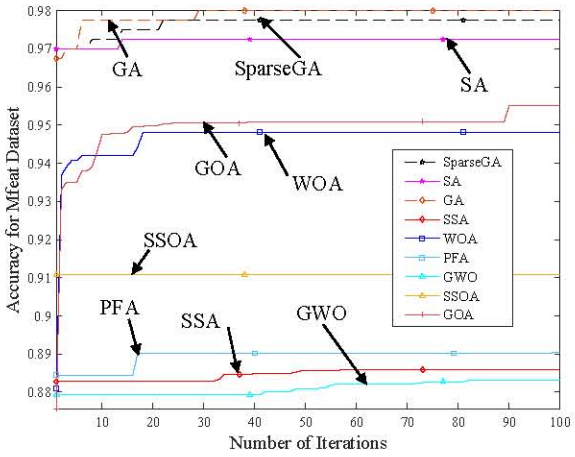
(11)



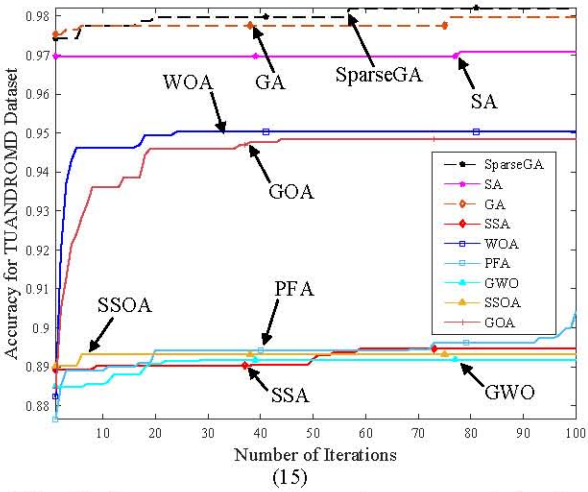
(12)



(13)



(14)



(15)

Fig. 3 Classification accuracy comparison and convergence chart under UCI datasets.

IV. SparseGA to Solve Feature Selection Problem and Classification of Hepatotoxicity Data

Hepatotoxicity is a prevalent adverse drug reaction and a primary cause for some drug discontinuation post-marketing because the liver plays a crucial role in metabolizing and eliminating xenobiotics [23-25]. Predicting the risk associated with hepatotoxic compounds is extremely challenging due to the complexity of liver mechanisms. Even human clinical trials are not always helpful as liver damage can occur rarely or take a long time to develop [26]. Consequently, hepatotoxicity is the most frequent cause of drug wastage [27]. It is crucial to identify and eliminate potentially problematic compounds at the early stages of drug discovery. This paper utilizes the SparseGA combined with the KNN classifier to classify hepatotoxic and non-hepatotoxic compounds.

A. Data Sources

By consulting the literature, we gathered medical data on 1475 hepatotoxic compounds and 1038 non-hepatotoxic compounds sourced from the SIDER, LiverTox, and Drugs.com databases. SIDER contains information about marketed drugs and their documented adverse drug reactions, derived from public documents and instructions. It provides details on side effect frequency, drug and side effect classification, and links to additional information, such as drug-target relationships. LiverTox offers current,

unbiased and accessible information on the diagnosis, causes, frequency, clinical patterns and treatment of liver injury induced by prescription and over-the-counter drugs, as well as specific herbal and dietary supplements.

During the process of data collection, keywords related to liver toxicity such as "hepatic," "liver," and the prefix "hepat" were utilized to extract liver toxicity information. To ensure data quality and reliability, the following steps were performed [38]. (1) Removal of compounds or molecular structures without specific Adverse Hepatic Effects (AHE); (2) Standardization of all compounds by using MOE (Molecular Operating Environment Software, 2016 version, Chemical Computing Group, Montreal, Quebec, Canada) to eliminate group metals in simple salts, retain only the largest molecular fragments, deprotonate strong acids, protonate strong bases and add explicit hydrogens; (3) Manual integration of hepatotoxicity information to maintain consistency. For instance, only one AHE was retained for duplicated AHEs such as jaundice. Subsequently, the 2D structure of the molecule was optimized by Merck Molecules to obtain a 3D structural force field, with a potential energy gradient threshold set at 0.001 $kcal^{-1}$ mol^{-1}. Finally, a total of 15873 compound-AHE pairs were associated with 2017 compounds and 403 AHEs for further in-depth analysis, and their associated frequency information was collected.

In the text, negative samples are defined as compounds not associated with hepatotoxicity or any AHE. Hence, after removing inconsistent data, 1038 negative samples from the previous study were ultimately collected. Among them, positive samples exhibited significant numerical differences across the three databases. Duplicate compounds accounted for 40% of the total data in the three databases, suggesting that drug-induced hepatotoxicity is generally diagnosed in clinical practice following regulatory approval of the drug. The overlap between LiverTox and Drugs.com is notably higher than the overlap between SIDER and these databases. As mentioned earlier, this overlap can be explained by the fact that SIDER's data originates from FDA drug labels, while LiverTox and Drugs.com extract hepatotoxicity information during the data collection process based on diverse clinical observations. Ultimately, a total of 1475 compounds with various Adverse Hepatic Effects (AGEs) are collected.

B. Data Classification Experiment and Analysis

In this experimental study, the parameter values and test conditions are the same as those under the UCI datasets. The comparison results of classification indicators under the confusion matrix, as well as the average fitness value, fitness value variance, number of sub-feature sets and accuracy non-parametric Wilcoxon test are shown in Table XVII-XVIII. Fig. 4 shows the comparison of the average classification error rate. Fig. 5 shows the visualization of the compared algorithms in terms of fitness, sensitivity, F1_Score, precision, G_mean, classification accuracy and specificity values.

As can be seen from Table XVI, compared with other algorithms, SparseGA has a higher accuracy rate in classification problems. Although it is about 2 percentage points different from GOA, it can be seen from the convergence diagram that GOA reached the current classification accuracy result at 10 iterations, and SparseGA still showed an upward trend at 59 iterations. Therefore, as the number of iterations of SparseGA increases, the classification accuracy will gradually increase. Compared with the original GA, it increased by about 4 percentage points. At the same time, it performs well in Precision, Sensitivity, Specificity, F1-score and G-mean indicators, indicating that this algorithm has good model accuracy and fitting effect. As can be seen from Table XVII, in terms of selecting the number of features, SparseGA selected an average of about 116 feature values from 206 feature values, retaining 56.31% of the features while ensuring classification accuracy. SSOA and GOA only retain about 3% of the number of features, which will make the model lack diversity. Therefore, when performing FS, it is necessary to weigh the number of retained features to avoid under-fitting and information loss problems caused by too few features. According to the results of Wilcoxon analysis, it is shown that SparseGA has a significant advantage in classification accuracy because the statistical significance level (p-value) is less than 0.05. This means that SparseGA can more accurately classify samples into the correct categories when performing classification tasks, showing excellent classification performance.

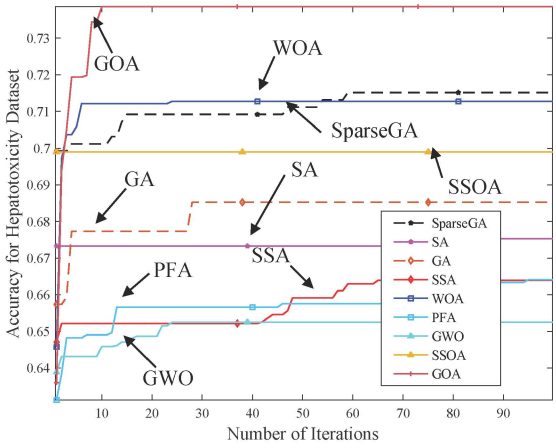


Fig. 4 Comparative convergence curves of classification accuracy under hepatotoxicity data.

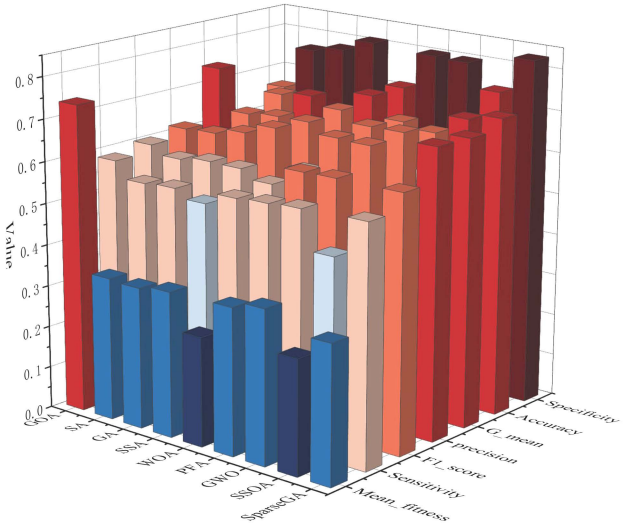


Fig. 5 3D visualization of classification indicators.

TABLE VI. THE AVERAGE CLASSIFICATION ACCURACY OF THE ALGORITHM UNDER THE UCI DATA SETS

	SparseGA	SA	GA	SSA	WOA	PFA	GWO	SSOA	GOA
Arrhythmia	**8.4444E-01**	7.3333E-01	7.4444E-01	7.1871E-01	8.4201E-01	7.3792E-01	6.9720E-01	7.7663E-01	8.3713E-01
COIL20	**9.8958E-01**	9.6875E-01	9.7569E-01	8.7719E-01	9.6914E-01	8.8570E-01	8.7854E-01	9.5807E-01	9.7747E-01
CNAE_9	**9.2130E-01**	7.6389E-01	8.5648E-01	7.8191E-01	7.7046E-01	7.9419E-01	7.8700E-01	7.1501E-01	8.0088E-01
Hill_Valley	6.2810E-01	5.6198E-01	5.8678E-01	5.8942E-01	6.4812E-01	6.0942E-01	5.7673E-01	6.4559E-01	**6.7003E-01**
Secom	9.3610E-01	9.3610E-01	9.3610E-01	8.6549E-01	9.4854E-01	8.7295E-01	8.6109E-01	9.4854E-01	**9.5110E-01**
Handwritten	**9.5283E-01**	9.2767E-01	9.2138E-01	8.3459E-01	8.6550E-01	8.4951E-01	8.4100E-01	8.2384E-01	8.7643E-01
QSAR_androgen_receptor	**9.3175E-01**	9.1395E-01	9.1988E-01	8.3747E-01	9.0954E-01	8.4908E-01	8.3736E-01	8.9629E-01	9.1648E-01
Har	**9.8148E-01**	8.8889E-01	9.0741E-01	8.2646E-01	9.1808E-01	8.6108E-01	8.2757E-01	8.4615E-01	9.2379E-01
HAPTDataSet	**9.4444E-01**	9.0278E-01	9.1667E-01	8.5460E-01	8.9127E-01	8.6144E-01	8.2935E-01	8.5668E-01	9.2264E-01
Isolet5	**8.7138E-01**	7.9743E-01	8.2958E-01	7.7025E-01	8.0557E-01	7.7831E-01	7.5357E-01	7.5360E-01	8.3879E-01
Semeion	**9.3396E-01**	8.9937E-01	8.9623E-01	8.2636E-01	8.4493E-01	8.4006E-01	8.3961E-01	8.1178E-01	8.5440E-01
UJIIndoorLoc	**1.0000E+00**	**1.0000E+00**	**1.0000E+00**	9.1494E-01	9.9962E-01	9.2222E-01	9.1724E-01	9.9962E-01	9.9962E-01
Madelon	8.4808E-01	7.4615E-01	8.1923E-01	7.3852E-01	8.4637E-01	7.5262E-01	7.1662E-01	7.3382E-01	**9.0951E-01**
Mfeat	9.7750E-01	9.7250E-01	**9.8000E-01**	8.8585E-01	9.4813E-01	8.9048E-01	8.8308E-01	9.1076E-01	9.5522E-01
TUANDROMD	**9.8206E-01**	9.7085E-01	9.7982E-01	8.9466E-01	9.5029E-01	9.0412E-01	8.9181E-01	8.9318E-01	9.4845E-01

TABLE VII. THE AVERAGE CLASSIFICATION SENSITIVITY VALUE OF THE ALGORITHM UNDER THE UCI DATA SETS

	SparseGA	SA	GA	SSA	WOA	PFA	GWO	SSOA	GOA
Arrhythmia	**1.0000E+00**	9.4000E-01	9.4000E-01	9.6000E-01	9.6000E-01	9.8000E-01	9.6000E-01	8.8000E-01	8.8000E-01
COIL20	**1.0000E+00**	**1.0000E+00**	**1.0000E+00**	**1.0000E+00**	**1.0000E+00**	**1.0000E+00**	**1.0000E+00**	**1.0000E+00**	**1.0000E+00**
CNAE_9	**1.0000E+00**	8.7500E-01	**1.0000E+00**	**1.0000E+00**	9.5833E-01	9.5455E-01	**1.0000E+00**	**1.0000E+00**	**1.0000E+00**
Hill_Valley	**6.2500E-01**	5.5000E-01	5.9167E-01	5.7500E-01	6.0833E-01	5.5000E-01	5.9167E-01	5.5000E-01	6.0833E-01
Secom	**1.0000E+00**	**1.0000E+00**	**1.0000E+00**	**1.0000E+00**	**1.0000E+00**	**1.0000E+00**	**1.0000E+00**	**1.0000E+00**	**1.0000E+00**
Handwritten	**1.0000E+00**	**1.0000E+00**	**1.0000E+00**	**1.0000E+00**	**1.0000E+00**	**1.0000E+00**	**1.0000E+00**	**1.0000E+00**	**1.0000E+00**
QSAR_androgen_receptor	**1.0000E+00**	9.8990E-01	9.9663E-01	**1.0000E+00**	9.8653E-01	**9.9663E-01**	**1.0000E+00**	9.6296E-01	9.6970E-01
Har	**1.0000E+00**	**1.0000E+00**	8.8889E-01	**1.0000E+00**	9.0000E-01	**1.0000E+00**	**1.0000E+00**	8.8889E-01	**1.0000E+00**
HAPTDataSet	**1.0000E+00**	**1.0000E+00**	**1.0000E+00**	**1.0000E+00**	**1.0000E+00**	**1.0000E+00**	**1.0000E+00**	**1.0000E+00**	**1.0000E+00**
Isolet5	**1.0000E+00**	**1.0000E+00**	**1.0000E+00**	**1.0000E+00**	**1.0000E+00**	**1.0000E+00**	**1.0000E+00**	**1.0000E+00**	**1.0000E+00**
Semeion	**1.0000E+00**	**1.0000E+00**	**1.0000E+00**	**1.0000E+00**	**1.0000E+00**	**1.0000E+00**	**1.0000E+00**	**1.0000E+00**	**1.0000E+00**
UJIIndoorLoc	**1.0000E+00**	**1.0000E+00**	**1.0000E+00**	**1.0000E+00**	**1.0000E+00**	**1.0000E+00**	**1.0000E+00**	**1.0000E+00**	**1.0000E+00**
Madelon	8.6923E-01	7.5385E-01	8.3077E-01	8.0385E-01	**8.2692E-01**	8.0385E-01	8.5769E-01	6.8846E-01	8.7692E-01
Mfeat	**1.0000E+00**	**1.0000E+00**	**1.0000E+00**	**1.0000E+00**	**1.0000E+00**	**1.0000E+00**	**1.0000E+00**	**1.0000E+00**	**1.0000E+00**
TUANDROMD	**1.0000E+00**	**1.0000E+00**	**1.0000E+00**	**1.0000E+00**	9.9719E-01	**1.0000E+00**	**1.0000E+00**	**1.0000E+00**	**1.0000E+00**

TABLE VIII. THE AVERAGE CLASSIFICATION SPECIFICITY VALUE OF THE ALGORITHM UNDER THE UCI DATA SETS

	SparseGA	SA	GA	SSA	WOA	PFA	GWO	SSOA	GOA
Arrhythmia	7.2222E-01	5.2778E-01	5.5556E-01	5.8333E-01	7.2222E-01	6.1111E-01	5.8333E-01	6.1111E-01	**7.7778E-01**
COIL20	**1.0000E+00**	**1.0000E+00**	**1.0000E+00**	**1.0000E+00**	**1.0000E+00**	**1.0000E+00**	**1.0000E+00**	**1.0000E+00**	**1.0000E+00**
CNAE_9	**1.0000E+00**	9.5833E-01	**1.0000E+00**	8.7500E-01	**1.0000E+00**	**1.0000E+00**	**1.0000E+00**	9.0000E-01	**1.0000E+00**
Hill_Valley	**6.3115E-01**	5.7377E-01	5.8197E-01	5.9836E-01	5.5738E-01	6.2295E-01	**5.9016E-01**	5.7377E-01	5.8197E-01
Secom	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	**5.0000E-02**
Handwritten	**1.0000E+00**	**1.0000E+00**	**1.0000E+00**	**1.0000E+00**	**1.0000E+00**	**1.0000E+00**	**1.0000E+00**	**1.0000E+00**	**1.0000E+00**
QSAR_androgen_receptor	4.5000E-01	3.5000E-01	3.5000E-01	3.2500E-01	4.2500E-01	**4.2500E-01**	4.0000E-01	3.0000E-01	4.7500E-01
Har	**1.0000E+00**	8.7500E-01	1.0000E+00	1.0000E+00	1.0000E+00	**1.0000E+00**	**1.0000E+00**	**1.0000E+00**	**1.0000E+00**
HAPTDataSet	**1.0000E+00**	**1.0000E+00**	**1.0000E+00**	**1.0000E+00**	**1.0000E+00**	**1.0000E+00**	**1.0000E+00**	8.8889E-01	**1.0000E+00**
Isolet5	**1.0000E+00**	**1.0000E+00**	**1.0000E+00**	**1.0000E+00**	8.8889E-01	**1.0000E+00**	**1.0000E+00**	9.0000E-01	**1.0000E+00**
Semeion	**1.0000E+00**	**1.0000E+00**	**1.0000E+00**	**1.0000E+00**	**1.0000E+00**	**1.0000E+00**	**1.0000E+00**	**1.0000E+00**	**1.0000E+00**
UJIIndoorLoc	**1.0000E+00**	**1.0000E+00**	**1.0000E+00**	**1.0000E+00**	**1.0000E+00**	**1.0000E+00**	**1.0000E+00**	**1.0000E+00**	**1.0000E+00**
Madelon	8.2692E-01	7.3846E-01	8.0769E-01	7.8846E-01	8.5000E-01	8.0769E-01	8.0385E-01	6.7308E-01	**9.0385E-01**
Mfeat	**1.0000E+00**	**1.0000E+00**	**1.0000E+00**	**1.0000E+00**	**1.0000E+00**	**1.0000E+00**	**1.0000E+00**	**1.0000E+00**	**1.0000E+00**
TUANDROMD	**9.1061E-01**	8.5475E-01	8.9944E-01	8.6592E-01	8.1006E-01	8.9385E-01	9.0503E-01	3.9665E-01	7.1508E-01

TABLE IX. THE AVERAGE CLASSIFICATION PRECISION VALUE OF THE ALGORITHM UNDER THE UCI DATA SETS

	SparseGA	SA	GA	SSA	WOA	PFA	GWO	SSOA	GOA
Arrhythmia	8.3333E-01	7.3438E-01	7.4603E-01	7.6190E-01	8.2759E-01	7.7778E-01	7.6190E-01	7.5862E-01	**8.4615E-01**
COIL20	**1.0000E+00**	**1.0000E+00**	**1.0000E+00**	**1.0000E+00**	**1.0000E+00**	**1.0000E+00**	**1.0000E+00**	**1.0000E+00**	**1.0000E+00**
CNAE_9	**1.0000E+00**	9.5455E-01	**1.0000E+00**	8.8889E-01	**1.0000E+00**	**1.0000E+00**	**1.0000E+00**	9.2000E-01	**1.0000E+00**
Hill_Valley	**6.2500E-01**	5.5932E-01	5.8197E-01	5.8475E-01	5.7480E-01	5.8929E-01	5.8678E-01	5.5932E-01	5.8871E-01
Secom	9.3610E-01	9.3610E-01	9.3610E-01	9.3610E-01	9.3610E-01	9.3610E-01	9.3610E-01	9.3610E-01	**9.3910E-01**
Handwritten	**1.0000E+00**	**1.0000E+00**	**1.0000E+00**	**1.0000E+00**	**1.0000E+00**	**1.0000E+00**	**1.0000E+00**	**1.0000E+00**	**1.0000E+00**
QSAR_androgen _receptor	**9.3082E-01**	9.1875E-01	9.1925E-01	9.1667E-01	9.2722E-01	9.2790E-01	9.2523E-01	9.1083E-01	9.3204E-01
Har	**1.0000E+00**	**9.0000E-01**	**1.0000E+00**	**1.0000E+00**	**1.0000E+00**	**1.0000E+00**	**1.0000E+00**	**1.0000E+00**	**1.0000E+00**
HAPTDataSet	**1.0000E+00**	**1.0000E+00**	**1.0000E+00**	**1.0000E+00**	**1.0000E+00**	**1.0000E+00**	**1.0000E+00**	9.1667E-01	**1.0000E+00**
Isolet5	**1.0000E+00**	**1.0000E+00**	**1.0000E+00**	**1.0000E+00**	9.1667E-01	**1.0000E+00**	**1.0000E+00**	9.2308E-01	**1.0000E+00**
Semeion	**1.0000E+00**	**1.0000E+00**	**1.0000E+00**	**1.0000E+00**	**1.0000E+00**	**1.0000E+00**	**1.0000E+00**	**1.0000E+00**	**1.0000E+00**
UJIIndoorLoc	**1.0000E+00**	**1.0000E+00**	**1.0000E+00**	**1.0000E+00**	**1.0000E+00**	**1.0000E+00**	**1.0000E+00**	**1.0000E+00**	**1.0000E+00**
Madelon	8.3395E-01	7.4242E-01	8.1203E-01	7.9167E-01	8.4646E-01	8.0695E-01	8.1387E-01	6.7803E-01	**9.0119E-01**
Mfeat	**1.0000E+00**	**1.0000E+00**	**1.0000E+00**	**1.0000E+00**	**1.0000E+00**	**1.0000E+00**	**1.0000E+00**	**1.0000E+00**	**1.0000E+00**
TUANDROMD	**9.7805E-01**	9.6482E-01	9.7538E-01	9.6744E-01	9.5436E-01	9.7404E-01	9.7671E-01	8.6845E-01	9.3325E-01

TABLE X. THE AVERAGE F1_SCORE VALUE OF THE ALGORITHM UNDER THE UCI DATA SETS

	SparseGA	SA	GA	SSA	WOA	PFA	GWO	SSOA	GOA
Arrhythmia	**9.0909E-01**	8.2456E-01	8.3186E-01	8.4956E-01	8.8889E-01	8.6726E-01	8.4956E-01	8.1481E-01	8.6275E-01
COIL20	**1.0000E+00**	**1.0000E+00**	**1.0000E+00**	**1.0000E+00**	**1.0000E+00**	**1.0000E+00**	**1.0000E+00**	**1.0000E+00**	**1.0000E+00**
CNAE_9	**1.0000E+00**	9.1304E-01	**1.0000E+00**	9.4118E-01	9.7872E-01	9.7674E-01	**1.0000E+00**	9.5833E-01	**1.0000E+00**
Hill_Valley	**6.2500E-01**	5.5462E-01	5.8678E-01	5.7983E-01	5.9109E-01	5.6897E-01	5.8921E-01	5.5462E-01	5.9836E-01
Secom	9.6700E-01	9.6700E-01	9.6700E-01	9.6700E-01	9.6700E-01	9.6700E-01	9.6700E-01	9.6700E-01	**9.6860E-01**
Handwritten	**1.0000E+00**	**1.0000E+00**	**1.0000E+00**	**1.0000E+00**	**1.0000E+00**	**1.0000E+00**	**1.0000E+00**	**1.0000E+00**	**1.0000E+00**
QSAR_androgen _receptor	**9.6260E-01**	9.5300E-01	9.5638E-01	9.5652E-01	9.5595E-01	9.6104E-01	9.6117E-01	9.3617E-01	9.5050E-01
Har	**1.0000E+00**	9.4737E-01	9.4118E-01	**1.0000E+00**	9.4737E-01	**1.0000E+00**	**1.0000E+00**	9.4118E-01	**1.0000E+00**
HAPTDataSet	**1.0000E+00**	**1.0000E+00**	**1.0000E+00**	**1.0000E+00**	**1.0000E+00**	**1.0000E+00**	**1.0000E+00**	9.5652E-01	**1.0000E+00**
Isolet5	**1.0000E+00**	**1.0000E+00**	**1.0000E+00**	**1.0000E+00**	9.5652E-01	**1.0000E+00**	**1.0000E+00**	9.6000E-01	**1.0000E+00**
Semeion	**1.0000E+00**	**1.0000E+00**	**1.0000E+00**	**1.0000E+00**	**1.0000E+00**	**1.0000E+00**	**1.0000E+00**	**1.0000E+00**	**1.0000E+00**
UJIIndoorLoc	**1.0000E+00**	**1.0000E+00**	**1.0000E+00**	**1.0000E+00**	**1.0000E+00**	**1.0000E+00**	**1.0000E+00**	**1.0000E+00**	**1.0000E+00**
Madelon	8.5122E-01	7.4809E-01	8.2129E-01	7.9771E-01	**8.3658E-01**	8.0539E-01	8.3521E-01	6.8321E-01	8.8889E-01
Mfeat	**1.0000E+00**	**1.0000E+00**	**1.0000E+00**	**1.0000E+00**	**1.0000E+00**	**1.0000E+00**	**1.0000E+00**	**1.0000E+00**	**1.0000E+00**
TUANDROMD	**9.8890E-01**	9.8209E-01	9.8753E-01	9.8345E-01	9.7531E-01	9.8685E-01	9.8822E-01	9.2960E-01	9.6547E-01

TABLE XI. THE AVERAGE G-MEAN VALUE OF THE ALGORITHM UNDER THE UCI DATA SETS

	SparseGA	SA	GA	SSA	WOA	PFA	GWO	SSOA	GOA
Arrhythmia	**8.4984E-01**	7.0435E-01	7.2265E-01	7.4833E-01	8.3267E-01	7.7388E-01	7.4833E-01	7.3333E-01	8.2731E-01
COIL20	**1.0000E+00**	**1.0000E+00**	**1.0000E+00**	**1.0000E+00**	**1.0000E+00**	**1.0000E+00**	**1.0000E+00**	**1.0000E+00**	**1.0000E+00**
CNAE_9	**1.0000E+00**	9.1572E-01	1.0000E+00	9.3541E-01	9.7895E-01	9.7701E-01	**1.0000E+00**	9.4868E-01	**1.0000E+00**
Hill_Valley	**6.2807E-01**	5.6176E-01	5.8680E-01	5.8656E-01	5.8230E-01	5.8534E-01	5.9091E-01	5.6176E-01	5.9500E-01
Secom	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	**2.2361E-01**
Handwritten	**1.0000E+00**	**1.0000E+00**	**1.0000E+00**	**1.0000E+00**	**1.0000E+00**	**1.0000E+00**	**1.0000E+00**	**1.0000E+00**	**1.0000E+00**
QSAR_androgen _receptor	6.7082E-01	5.8861E-01	5.9061E-01	5.7009E-01	6.4752E-01	6.5082E-01	6.3246E-01	5.3748E-01	**6.7868E-01**
Har	**1.0000E+00**	9.3541E-01	9.4281E-01	**1.0000E+00**	9.4868E-01	**1.0000E+00**	**1.0000E+00**	9.4281E-01	**1.0000E+00**
HAPTDataSet	**1.0000E+00**	**1.0000E+00**	**1.0000E+00**	**1.0000E+00**	**1.0000E+00**	**1.0000E+00**	**1.0000E+00**	9.4281E-01	**1.0000E+00**
Isolet5	**1.0000E+00**	**1.0000E+00**	**1.0000E+00**	**1.0000E+00**	9.4281E-01	**1.0000E+00**	**1.0000E+00**	9.4868E-01	**1.0000E+00**
Semeion	**1.0000E+00**	**1.0000E+00**	**1.0000E+00**	**1.0000E+00**	**1.0000E+00**	**1.0000E+00**	**1.0000E+00**	**1.0000E+00**	**1.0000E+00**
UJIIndoorLoc	**1.0000E+00**	**1.0000E+00**	**1.0000E+00**	**1.0000E+00**	**1.0000E+00**	**1.0000E+00**	**1.0000E+00**	1.0000E+00	1.0000E+00
Madelon	8.4781E-01	7.4611E-01	8.1915E-01	7.9612E-01	8.3838E-01	8.0577E-01	8.3033E-01	6.8073E-01	**8.9028E-01**
Mfeat	**1.0000E+00**	**1.0000E+00**	**1.0000E+00**	**1.0000E+00**	**1.0000E+00**	**1.0000E+00**	**1.0000E+00**	**1.0000E+00**	**1.0000E+00**
TUANDROMD	**9.5426E-01**	9.2453E-01	9.4839E-01	9.3055E-01	8.9877E-01	9.4544E-01	9.5133E-01	6.2980E-01	8.4563E-01

TABLE XII. THE AVERAGE FITNESS VALUE OF THE ALGORITHM UNDER THE UCI DATA SETS

	SparseGA	SA	GA	SSA	WOA	PFA	GWO	SSOA	GOA
Arrhythmia	2.1839E-01	2.9556E-01	2.7803E-01	3.0047E-01	1.5008E-01	2.8247E-01	3.3058E-01	2.0605E-01	**1.4801E-01**
COIL20	6.2500E-02	7.8711E-02	7.3340E-02	1.6082E-01	3.0703E-02	1.4613E-01	1.6449E-01	4.2031E-02	**2.3203E-02**
CNAE_9	**1.2815E-01**	2.6332E-01	1.7917E-01	2.4791E-01	2.7692E-01	2.3628E-01	2.6857E-01	3.0824E-01	2.0654E-01
Hill_Valley	3.8587E-01	4.4521E-01	4.2090E-01	4.0952E-01	3.2570E-01	3.8152E-01	4.2895E-01	3.2097E-01	**2.9997E-01**
Secom	1.1022E-01	1.0683E-01	1.0683E-01	1.6275E-01	4.6481E-02	1.5231E-01	1.6892E-01	4.6481E-02	**4.4181E-02**
Handwritten	**9.3330E-02**	1.1822E-01	1.2075E-01	1.9887E-01	1.5308E-01	1.8427E-01	2.0248E-01	2.0386E-01	1.3777E-01
QSAR_androgen_receptor	1.1989E-01	1.2755E-01	1.2113E-01	1.9550E-01	9.4600E-02	1.8280E-01	1.9921E-01	9.8909E-02	**8.1322E-02**
Har	1.1301E-01	1.5009E-01	1.3182E-01	2.0592E-01	7.7647E-02	1.7226E-01	2.1918E-01	1.4132E-01	**6.9661E-02**
HAPTDataSet	1.1009E-01	1.3759E-01	1.2455E-01	1.7578E-01	1.1889E-01	1.7176E-01	2.1668E-01	1.3398E-01	**7.4973E-02**
Isolet5	1.8417E-01	2.3369E-01	2.0135E-01	2.5734E-01	1.8860E-01	2.4863E-01	2.8970E-01	2.2792E-01	**1.5238E-01**
Semeion	**1.1411E-01**	1.4057E-01	1.4144E-01	2.0159E-01	1.7432E-01	1.9121E-01	2.0568E-01	2.1823E-01	1.5604E-01
UJIIndoorLoc	5.2299E-02	4.8659E-02	4.9425E-02	1.1908E-01	**5.3640E-04**	1.0889E-01	1.1586E-01	5.3640E-04	**5.3640E-04**
Madelon	2.0165E-01	2.8186E-01	2.0549E-01	2.8453E-01	1.5047E-01	2.6865E-01	3.2905E-01	2.4497E-01	**8.2843E-02**
Mfeat	7.2022E-02	7.2670E-02	6.4225E-02	1.5081E-01	5.3614E-02	1.4233E-01	1.5469E-01	8.4943E-02	**4.3692E-02**
TUANDROMD	7.0679E-02	8.0590E-02	6.5464E-02	1.3672E-01	5.3457E-02	1.2571E-01	1.4385E-01	1.0112E-01	**4.9299E-02**

TABLE XIII. ALGORITHM FITNESS VALUE VARIANCE UNDER UCI DATA SETS

	SparseGA	SA	GA	SSA	WOA	PFA	GWO	SSOA	GOA
Arrhythmia	8.9201E-03	**0.0000E+00**	**0.0000E+00**	**0.0000E+00**	2.9257E-17	5.8514E-17	5.8514E-17	**0.0000E+00**	2.9257E-17
COIL20	5.8675E-04	**0.0000E+00**	**0.0000E+00**	2.9257E-17	3.6571E-18	2.9257E-17	2.9257E-17	7.3142E-18	3.6571E-18
CNAE_9	1.9087E-03	5.8514E-17	2.9257E-17	5.8514E-17	5.8514E-17	5.8514E-17	5.8514E-17	5.8514E-17	**0.0000E+00**
Hill_Valley	2.8957E-03	5.8514E-17	5.8514E-17	**0.0000E+00**	5.8514E-17	**0.0000E+00**	1.1703E-16	5.8514E-17	5.8514E-17
Secom	6.9677E-04	**0.0000E+00**	**0.0000E+00**	**0.0000E+00**	**0.0000E+00**	**0.0000E+00**	2.9257E-17	**0.0000E+00**	**0.0000E+00**
Handwritten	1.5383E-04	**0.0000E+00**	1.4628E-17	2.9257E-17	**0.0000E+00**	**0.0000E+00**	2.9257E-17	**0.0000E+00**	**0.0000E+00**
QSAR_androgen_receptor	1.7199E-03	**0.0000E+00**	1.4628E-17	2.9257E-17	1.4628E-17	**0.0000E+00**	2.9257E-17	1.4628E-17	1.4628E-17
Har	1.4459E-02	**0.0000E+00**	2.9257E-17	5.8514E-17	**0.0000E+00**	2.9257E-17	2.9257E-17	**0.0000E+00**	1.4628E-17
HAPTDataSet	2.4873E-03	2.9257E-17	**0.0000E+00**	**0.0000E+00**	**0.0000E+00**	2.9257E-17	**0.0000E+00**	2.9257E-17	1.4628E-17
Isolet5	5.3883E-03	5.8514E-17	2.9257E-17	5.8514E-17	**0.0000E+00**	5.8514E-17	**0.0000E+00**	2.9257E-17	**0.0000E+00**
Semeion	6.1530E-04	2.9257E-17	**0.0000E+00**	2.9257E-17	**0.0000E+00**	**0.0000E+00**	5.8514E-17	2.9257E-17	**0.0000E+00**
UJIIndoorLoc	1.3933E-03	**0.0000E+00**	**0.0000E+00**	**0.0000E+00**	1.1428E-19	**0.0000E+00**	1.4628E-17	1.1428E-19	1.1428E-19
Madelon	5.6653E-03	5.8514E-17	2.9257E-17	**0.0000E+00**	2.9257E-17	5.8514E-17	5.8514E-17	2.9257E-17	1.4628E-17
Mfeat	1.3156E-03	1.4628E-17	1.4628E-17	**0.0000E+00**	**0.0000E+00**	2.9257E-17	2.9257E-17	**0.0000E+00**	7.3142E-18
TUANDROMD	1.8936E-03	**0.0000E+00**	**0.0000E+00**	2.9257E-17	**0.0000E+00**	2.9257E-17	2.9257E-17	1.4628E-17	7.3142E-18

TABLE XIV. THE AVERAGE NUMBER OF FEATURE SELECTIONS BY THE ALGORITHM UNDER THE UCI DATA SETS

	SparseGA	SA	GA	SSA	WOA	PFA	GWO	SSOA	GOA
Arrhythmia	1.3550E+02	1.5500E+02	1.3400E+02	1.3200E+02	2.2000E+01	1.3000E+02	1.6200E+02	1.4000E+01	**4.0000E+00**
COIL20	4.8450E+02	5.1800E+02	5.2700E+02	5.1500E+02	**3.0000E+01**	4.4300E+02	5.6500E+02	4.4000E+01	**3.0000E+01**
CNAE_9	4.4060E+02	4.3500E+02	4.2800E+02	4.4200E+02	6.0200E+02	4.3700E+02	6.5800E+02	4.4300E+02	**2.3400E+02**
Hill_Valley	4.0200E+01	5.1000E+01	4.9000E+01	4.0000E+01	9.0000E+00	3.0000E+01	4.8000E+01	**2.0000E+00**	3.0000E+00
Secom	2.9930E+02	2.9100E+02	2.9100E+02	2.4600E+02	**1.0000E+00**	2.2400E+02	2.5900E+02	**1.0000E+00**	**1.0000E+00**
Handwritten	1.2840E+02	1.3600E+02	1.2800E+02	1.2800E+02	8.2000E+01	1.2500E+02	1.5200E+02	1.1600E+02	**6.8000E+01**
QSAR_androgen_receptor	5.4310E+02	5.1300E+02	5.0200E+02	5.0400E+02	1.3500E+02	4.8100E+02	5.4100E+02	**5.7000E+01**	6.3000E+01
Har	2.8160E+02	2.8100E+02	2.7200E+02	2.7900E+02	2.2000E+01	2.6500E+02	3.5900E+02	1.6000E+01	**6.0000E+00**
HAPTDataSet	2.9040E+02	2.8100E+02	2.7800E+02	2.5200E+02	1.1800E+02	2.6400E+02	3.5400E+02	**2.8000E+01**	3.0000E+01
Isolet5	3.1520E+02	3.1700E+02	2.9600E+02	3.1200E+02	8.4000E+01	3.0300E+02	4.1900E+02	**3.8000E+01**	4.5000E+01
Semeion	1.1340E+02	1.2800E+02	1.2300E+02	1.1600E+02	8.9000E+01	1.2100E+02	1.5700E+02	1.2500E+02	**6.4000E+01**
UJIIndoorLoc	2.7070E+02	2.5400E+02	2.5800E+02	2.2200E+02	**1.0000E+00**	2.0300E+02	2.1600E+02	**1.0000E+00**	**1.0000E+00**
Madelon	2.3790E+02	2.6700E+02	2.1400E+02	2.4600E+02	6.1000E+01	2.3000E+02	3.7000E+02	2.7000E+01	**7.0000E+00**
Mfeat	3.3330E+02	3.1100E+02	3.0000E+02	3.1200E+02	4.5000E+01	2.8400E+02	3.2100E+02	3.0000E+01	**2.2000E+01**
TUANDROMD	1.1820E+02	1.3100E+02	1.1400E+02	1.0100E+02	2.1000E+01	9.5000E+01	1.1200E+02	1.2000E+01	**7.0000E+00**

TABLE XV. NON-PARAMETRIC WILCOXON TEST OF ALGORITHM CLASSIFICATION ACCURACY UNDER UCI DATA SETS

	SA	GA	SSA	WOA	PFA	GWO	SSOA	GOA
Arrhythmia	2.4000E-05	2.4000E-05	2.4000E-05	2.4000E-05	NaN	2.4000E-05	2.4000E-05	2.4000E-05
COIL20	2.4000E-05	NaN	2.4000E-05	2.4000E-05	2.4000E-05	2.4000E-05	1.6000E-05	7.5600E-04
CNAE_9	2.4000E-05	2.4000E-05	2.4000E-05	1.6000E-05	2.4000E-05	2.4000E-05	2.4000E-05	2.4000E-05
Hill_Valley	1.6000E-05	2.4000E-05	2.4000E-05	2.4000E-05	2.4000E-05	1.6000E-05	2.4000E-05	2.4000E-05
Secom	2.4000E-05	2.4000E-05	1.6000E-05	2.4000E-05	2.4000E-05	2.4000E-05	2.4000E-05	1.6000E-05
Handwritten	2.4000E-05	2.4000E-05	2.4000E-05	2.4000E-05	2.4000E-05	2.4000E-05	2.4000E-05	9.7000E-05
QSAR_androgen _receptor	3.6812E-01	2.4000E-05	2.4000E-05	2.4000E-05	2.4000E-05	2.4000E-05	2.4000E-05	2.4000E-05
Har	2.4000E-05	2.4000E-05	2.4000E-05	2.4000E-05	2.4000E-05	2.4000E-05	2.4000E-05	2.4000E-05
HAPTDataSet	2.4000E-05	2.4000E-05	2.4000E-05	2.4000E-05	2.4000E-05	2.4000E-05	2.4000E-05	2.4000E-05
Isolet5	2.4000E-05	2.4000E-05	2.4000E-05	2.4000E-05	2.4000E-05	2.4000E-05	2.4000E-05	2.4000E-05
Semeion	2.4000E-05	NaN	2.4000E-05	1.6000E-05	2.4000E-05	2.4000E-05	NaN	2.4000E-05
UJIIndoorLoc	1.6000E-05	2.4000E-05	2.4000E-05	1.6000E-05	2.4000E-05	1.6000E-05	2.4000E-05	2.4000E-05
Madelon	1.6000E-05	7.5600E-04	1.6000E-05	2.4000E-05	2.4000E-05	1.6000E-05	2.4000E-05	1.6000E-05
Mfeat	2.4000E-05	2.4000E-05	1.6000E-05	2.4000E-05	1.6000E-05	2.4000E-05	2.4000E-05	1.6000E-05
TUANDROMD	2.4000E-05	1.6000E-05	2.4000E-05	2.4000E-05	1.6000E-05	2.4000E-05	1.6000E-05	2.4000E-05

TABLE XVI. TIME COMPLEXITY OF ALGORITHM CLASSIFICATION ACCURACY UNDER UCI DATA SETS

	SparseGA	SA	GA	SSA	WOA	PFA	GWO	SSOA	GOA
Arrhythmia	1.4874E+01	6.7953E+01	1.3696E+01	1.1356E+01	8.2003E+00	1.1549E+01	1.2263E+01	**3.1085E+00**	7.9059E+00
COIL20	1.0309E+02	5.1146E+02	1.0562E+02	1.0040E+02	1.6848E+01	9.8270E+01	1.1227E+02	2.0854E+01	**1.2695E+01**
CNAE_9	6.0814E+01	2.7426E+02	5.5320E+01	5.4335E+01	4.9637E+01	5.3100E+01	7.9407E+01	**2.0442E+01**	2.8612E+01
Hill_Valley	1.4896E+01	7.6895E+01	1.5505E+01	1.2611E+01	1.0414E+01	1.2905E+01	1.3279E+01	**7.7095E-01**	7.2574E+00
Secom	7.2525E+01	3.4364E+02	6.8774E+01	6.1083E+01	9.7855E+00	6.6682E+01	6.9879E+01	**4.8088E+00**	8.6729E+00
Handwritten	3.4821E+01	1.6460E+02	3.3146E+01	3.2356E+01	2.2252E+01	3.1804E+01	3.7973E+01	**1.2734E+01**	1.5388E+01
QSAR_androgen _receptor	1.4283E+02	6.5518E+02	1.3008E+02	1.2987E+02	3.9681E+01	1.2928E+02	1.4294E+02	2.1850E+01	**1.9122E+01**
Har	1.4241E+01	6.0512E+01	1.2153E+01	1.1135E+01	8.7207E+00	1.1281E+01	1.2412E+01	7.8088E+00	**7.7429E+00**
HAPTDataSet	1.6166E+01	6.8231E+01	1.3676E+01	1.2396E+01	1.0273E+01	1.2719E+01	1.4177E+01	9.0587E+00	**8.2936E+00**
Isolet5	7.6773E+01	3.6100E+02	6.9734E+01	7.0264E+01	2.5011E+01	7.0460E+01	9.5005E+01	**1.3912E+01**	1.7905E+01
UJIIndoorLoc	1.5249E+01	6.4717E+01	1.3132E+01	1.0752E+01	6.2903E+00	1.1022E+01	1.1398E+01	**3.2871E+00**	5.4309E+00
Madelon	1.4338E+02	6.8796E+02	1.1910E+02	1.3496E+02	3.6676E+01	1.3379E+02	2.0383E+02	**1.3832E+01**	1.6297E+01
Mfeat	1.1994E+02	5.6899E+02	1.0603E+02	1.0984E+02	2.2978E+01	1.0974E+02	1.1789E+02	**1.3346E+01**	1.5905E+01
TUANDROMD	1.6467E+02	8.4344E+02	1.5639E+02	1.4274E+02	3.5425E+01	1.5689E+02	1.5319E+02	**9.2666E+00**	2.4483E+01

TABLE XVII. AVERAGE CLASSIFICATION PERFORMANCE INDEX OF CONFUSION MATRIX UNDER HEPATOTOXICITY DATA SETS

	Algorithms	Accuracy	Sensitivity	Specificity	precision	F1_score	G_mean
Hepatotoxicity	SparseGA	7.1514E-01	5.7488E-01	**8.3051E-01**	**6.9512E-01**	6.1456E-01	**6.9097E-01**
	SA	6.7530E-01	5.4589E-01	7.6610E-01	6.2088E-01	5.8098E-01	6.4669E-01
	GA	6.8526E-01	5.5072E-01	7.7966E-01	6.3687E-01	5.9067E-01	6.5527E-01
	SSA	6.6394E-01	5.3140E-01	8.1017E-01	6.6265E-01	5.8981E-01	6.5614E-01
	WOA	7.1273E-01	5.6039E-01	7.1186E-01	5.7711E-01	5.6863E-01	6.3160E-01
	PFA	6.6414E-01	5.6522E-01	8.0339E-01	6.6857E-01	6.1257E-01	6.7386E-01
	GWO	6.5249E-01	5.7005E-01	8.0000E-01	6.6667E-01	**6.1458E-01**	6.7531E-01
	SSOA	6.9901E-01	4.7826E-01	7.4237E-01	5.6571E-01	5.1832E-01	5.9586E-01
	GOA	**7.3857E-01**	**5.8454E-01**	7.4576E-01	6.1735E-01	6.0050E-01	6.6025E-01

TABLE XVIII. AVERAGE CLASSIFICATION PERFORMANCE INDEX OF THE ALGORITHM UNDER THE HEPATOTOXICITY DATA SETS

	Algorithms	Mean_fitness	Std_fitness	Feature number	Wilcoxon	Time
Hepatotoxicity	SparseGA	3.3114E-01	5.6817E-03	1.1590E+02		5.7979E+01
	SA	3.4320E-01	**0.0000E+00**	1.0500E+02	2.4000E-05	2.6086E+02
	GA	3.3909E-01	**0.0000E+00**	1.1500E+02	2.4000E-05	5.8041E+01
	SSA	3.4857E-01	**0.0000E+00**	9.5000E+01	2.4000E-05	4.8207E+01
	WOA	2.6194E-01	**0.0000E+00**	7.0000E+00	7.5600E-04	1.5056E+01
	PFA	3.5227E-01	**0.0000E+00**	1.0300E+02	2.4000E-05	4.9699E+01
	GWO	3.6858E-01	5.8514E-17	1.1500E+02	2.4000E-05	5.8335E+01
	SSOA	2.7478E-01	5.8514E-17	8.0000E+00	7.5600E-04	**5.7609E+00**
	GOA	**2.3771E-01**	**0.0000E+00**	**5.0000E+00**	2.4000E-05	1.1817E+01

V. Conclusion

SparseGA is proposed for large-scale sparse high-dimensional data set optimization problems. The algorithm achieved a significant increase in classification accuracy when applied to hepatotoxicity classification in the medical field, improving the classification accuracy from 68.53% to 71.51%. The algorithm divides decision variables into real-valued vectors and 0-1 binary vectors to ensure sparsity, and utilizes a greedy population initialization strategy and greedy genetic operators to expedite convergence. Future research could explore combining SparseGA with other evolutionary algorithms to address multi-objective optimization problems, and testing it on high-dimensional sparse datasets in different domains to enhance data classification accuracy. These research findings offer new insights for predicting compound hepatotoxicity in drug development stages, and demonstrate the potential of SparseGA in optimization algorithms and data classification accuracy.

References

[1] E. Emary, Hossam, M. Zawbaa, and A. E. Hassanien, "Binary grey wolf optimization approaches for feature selection," *Neurocomputing*, vol. 172, pp. 371-381, 2016.

[2] M. Tubishat, M. A. M. Abushariah, N. Idris, and I. Aljarah, "Improved whale optimization algorithm for feature selection in Arabic sentiment analysis," *Applied Intelligence*, vol. 49, no. 5, pp. 1688-1707, 2019.

[3] L. Abualigah, M. Shehab, M. Alshinwan, and H. Alabool, "Salp swarm algorithm: a comprehensive survey," *Neural Computing and Applications*, vol. 32, no. 15, pp. 11195-11215, 2020.

[4] H. Yapici, and N. Cetinkaya, "A new meta-heuristic optimizer: Pathfinder algorithm," *Applied Soft Computing*, vol. 78, pp. 545-568, 2019.

[5] E. Aličković, and A. Subasi, "Breast cancer diagnosis using GA feature selection and Rotation Forest," *Neural Computing and Applications*, vol. 28, no. 4, pp. 753-763, 2017.

[6] C. L. Huang, and C. J. Wang, "A GA-based feature selection and parameters optimization for support vector machines," *Expert Systems with Applications*, vol. 31, no. 2, pp. 231-240, 2006.

[7] C. De Stefano, F. Fontanella, C. Marrocco, and A. Scotto di Freca, "A GA-based feature selection approach with an application to handwritten character recognition," *Pattern Recognition Letters*, vol. 35, pp. 130-141, 2014.

[8] X. P Li, Y D. Wang, and R. Ruiz, "A survey on sparse learning models for feature selection," *IEEE Transactions on Cybernetics*, vol. 52, no. 3, pp. 1642-1660, 2020.

[9] C. P. Hou, F. P. Nie, X. L. Li, D. Y. Yi, and Y. Wu, "Joint embedding learning and sparse regression: A framework for unsupervised feature selection," *IEEE Transactions on Cybernetics*, vol. 44, no. 6, pp. 793-804, 2013.

[10] A. Y. Yang, J. Wright, Y. Ma, and S. S. Sastry, "Feature selection in face recognition: A sparse representation perspective," *IEEE Transactions Pattern Analysis and Machine Intelligence*, vol. 2, pp. 1-34, 2007.

[11] N. Maleki, Y. Zeinali, and S. T. A. Niaki, "A k-NN method for lung cancer prognosis with the use of a genetic algorithm for feature selection," *Expert Systems with Applications*, vol. 164, pp. 113981, 2021.

[12] S. Aalaei, H. Shahraki, A. Rowhanimanesh, and S. Eslami, "Feature selection using genetic algorithm for breast cancer diagnosis: experiment on three different datasets," *Iranian Journal of Basic Medical Sciences*, vol. 19, no. 5, pp. 476, 2016.

[13] L. K. Singh, M. Khanna, H. Garg, and R. Singh, "Efficient feature selection based novel clinical decision support system for glaucoma prediction from retinal fundus images," *Medical Engineering & Physics*, vol. 123, pp. 104077, 2024.

[14] L. K. Singh, M. Khanna, H. Garg, and R. Singh, "Emperor penguin optimization algorithm-and bacterial foraging optimization algorithm-based novel feature selection approach for glaucoma classification from fundus images," *Soft Computing*, vol. 28, no. 3, pp. 2431-2467, 2024.

[15] S. M. Paul, D. S. Mytelka, C. T. Dunwiddie, C. C. Persinger, B. H. Munos, S. R. Lindborg, and A. L. Schacht, "How to improve R&D productivity: the pharmaceutical industry's grand challenge," *Nature Reviews Drug Discovery*, vol. 9, no. 3, pp. 203-214, 2010.

[16] R. A. Wilke, D. W. Lin, D. M. Roden, P. B. Watkins, D. Flockhart, I. Zineh, K. M. Giacomini, and R. M. Krauss, "Identifying genetic risk factors for serious adverse drug reactions: current progress and challenges," *Nature Reviews Drug Discovery*, vol. 6, no. 11, pp. 904-916, 2007.

[17] J. Liu, W. J Guo, S. Sakkiah, Z. W. Ji, G. Yavas, W. Zou, M. J. Chen, W. D. Tong, T. A. Patterson, and H. X. Hong, "Machine learning models for predicting liver toxicity," *Silico Methods for Predicting Drug Toxicity*, vol. 2425, pp. 393-415, 2022.

[18] M. J. Chen, H. X. Hong, H. Fang, R. Kelly, G. X. Zhou, J. Borlak, and W. D. Tong, "Quantitative structure-activity relationship models for predicting drug-induced liver injury based on FDA-approved drug labeling annotation and using a large collection of drugs," *Toxicological Sciences*, vol. 136, no. 1, pp. 242-249, 2013.

[19] D. P. Williams, S. E. Lazic, A. J. Foster, E. Semenova, and P. Morgan, "Predicting drug-induced liver injury with Bayesian machine learning," *Chemical Research in Toxicology*, vol. 33, no. 1, pp. 239-248, 2019.

[20] X. W. Zhu, Y. J. Xin, and Q. H. Chen, "Chemical and in vitro biological information to predict mouse liver toxicity using recursive random forests," *SAR and QSAR in Environmental Research*, vol. 27, no. 7, pp. 559-572, 2016.

[21] Y. J. Zhang, Y. Tian, and X. Y. Zhang, "Improved SparseEA for sparse large-scale multi-objective optimization problems," *Complex & Intelligent Systems*, pp. 1-16, 2021.

[22] Y. C. Lo, S. E. Rensi, W. Torng, and R. B. Altman, "Machine learning in chemoinformatics and drug discovery," *Drug Discovery Today*, vol. 23, no. 8, pp. 1538-1546, 2018.

[23] M. Bourhia, R. Ullah, A. S. Alqahtani, and S. Ibenmoussa, "Evidence of drug-induced hepatotoxicity in the Maghrebian population," *Drug and Chemical Toxicology*, vol. 45, no. 3, pp. 985-989, 2022.

[24] S. Russmann, A. G. Kullak-Ublick, and I. Grattagliano, "Current concepts of mechanisms in drug-induced hepatotoxicity," *Current Medicinal Chemistry*, vol. 16, no. 23, pp. 3041-3053, 2009.

[25] A. Regev, "Drug-induced liver injury and drug development: industry perspective," *Seminars in Liver Disease*, *Thieme Medical Publishers*, vol. 34, no. 2, pp. 227-239, 2014.

[26] D. Mulliner, F. Schmidt, M. Stolte, H. Spirkl, A. Czich, and A. Amberg, "Computational models for human and animal hepatotoxicity with a global application scope," *Chemical Research in Toxicology*, vol. 29, no. 5, pp. 757-767, 2016.

[27] I. J. Onakpoya, C. J. Heneghan, and J. K. Aronson, "Post-marketing withdrawal of 462 medicinal products because of adverse drug reactions: a systematic review of the world literature," *BMC Medicine*, vol. 14, no. 1, pp. 1-11, 2016.