IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received March 19, 2019, accepted May 26, 2019, date of publication May 30, 2019, date of current version July 2, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2919956

A Feature Selection Method Based on Hybrid Improved Binary Quantum Particle Swarm Optimization

QING WU, ZHEPING MA, JIN FAN, GANG XU, AND YUANFENG SHEN
Department of Computer Science and Technology, Hangzhou Dianzi University, Hangzhou 310018, China

Corresponding author: Jin Fan (fanjin@hdu.edu.cn)

This work was supported by the Science and Technology Program of Zhejiang Province under Grant 2018C04001.

ABSTRACT As the volume of data available for analysis grows, feature selection is becoming a vital part of ensuring accurate classification results. In classification problems, selecting a small number of features reduces computational complexity, but selecting the right features is important to maintain a high level of accuracy. In this paper, we present a feature selection method based on hybrid improved quantum-behavior particle swarm optimization, called HI-BQPSO. The HI-BQPSO combines a filtering method with an improved quantum-behavior particle swarm optimization algorithm to greatly reduce the dimensionality of the data so as to overcome some of the shortcomings of BQPSO. Tests were conducted on nine gene expression datasets and 36 UCI datasets to evaluate and compare the classification accuracy of the HI-BQPSO's selected feature subsets against four other algorithms. The results, using a variety of different classifiers, show that the HI-BQPSO significantly reduces the number of features required for classification while maintaining higher levels of accuracy in many cases.

INDEX TERMS Feature selection, binary quantum particle swarm optimization, classification, harmonic average, random heuristic search.

I. INTRODUCTION

To data scientists, big data means big dimensionality and an enormous number of related, unrelated, and redundant attributes and relationships [2]. Obviously, such a huge number of characteristics can affect the performance of data analysis. For example, training models can take longer, algorithms are likely to be less efficient, and the resulting model may be more complex. This phenomenon is known as ''dimension disaster'' or ''the curse of dimensionality''. Dimensionality reduction is a common method of dealing with dimensionality challenges, and feature selection or feature extraction is an obvious approach to reducing dimensionality. The goal of feature selection is to find the best subset of features from all possible choices [3] to improve the quality and efficiency of any subsequent data analysis and help us better understand the characteristics of big data and its underlying structures.

There are many aspects to consider in feature selection but, in this paper, we focus on maximizing classification performance and minimizing the size of the selected feature subsets. Our optimization process needs to weigh these two goals.

The feature selection process generally consists of four parts: feature subset search, evaluation, search stop criteria, and validity verification. Feature selection methods can be roughly divided into three categories based on the evaluation criteria used: filter, wrapper, and embedded. When categorized according to the search strategy, these methods can be divided into global optimizations, heuristic searches, and random searches [4]. Although global optimal searches usually produce the best feature subset, they are not widely used due to their high computational complexity along with some other factors [5]. Heuristic searches are less computationally intensive, but it is easy to become trapped in the local optimal solution [6]. Random searches use a combination of random features to find the optimal solution of the objective function.

Some methods combine heuristics with a random search, also known as random heuristic searches. The simulated annealing algorithm (SA) [7], the genetic algorithm (GA) [8], the ant colony optimization algorithm (AOC) [9], the artificial bee colony algorithm (ABC) [10], and particle

The associate editor coordinating the review of this manuscript and approving it for publication was Muhammad Asif Naeem.

swarm optimization (PSO) [11] are among a few examples of a random heuristic search strategy. Each of these algorithms require some parameters to be set, and the appropriateness of the parameter selection plays a key role in the final result. One reason why the PSO algorithm is so widely used is that it can converge quickly and not many parameters need to be adjusted. However, as with all heuristic searches, PSO can easily become trapped in the local optima.

Hence, in this paper, we propose the hybrid improved binary quantum particle swarm optimization algorithm, or HI-BQPSO for short. HI-BQPSO is a multi-objective feature selection method that reduces the number of features to be considered while maximizing classification performance. Our algorithm overcomes the dimension disaster problem by combining the advantages of filtering and a random heuristic search.

To validate the effectiveness of HI-BQPSO, we conducted a series of experiments on gene expression data from the UCI Machine Learning Repository, along with some comparative evaluations using a variety of different classifiers. The results demonstrate HI-BQPSO's efficiency and advantages and show that our approach generated more accurate classifications on most datasets using significantly less selected features than traditional methods.

In summary, the main contributions of this paper are:

- A hybrid filtering method that: greatly reduces the search space of the heuristic random search algorithm.
- In order to avoid local convergence, the proposed method improves the calculation of local attractors in BQPSO and introduces the idea of crossover and mutation.
- The design of a fitness function using the weighted average principle, which balances the number of features selected and classification accuracy.
- The results of an extensive set of simulation experiments on nine gene expression datasets and 36 UCI datasets, which prove that the HI-BQPSO algorithm produces favorable outcomes and more accurate classifications with most of the datasets using markedly less selected features than baseline methods.

The rest of this paper is organized as follows. Section II summarizes the related work. The HI-BQPSO algorithm is introduced in Section III. Section IV sets out the experimental procedure. The results are analyzed in Section V, followed by the conclusion in Section VI.

II. RELATED WORK

There are many studies related to evaluating criteria in feature selection. Traditional evaluation criteria include Pearson [12], [13], mutual information [14], [15], and Spearman [16]. The Pearson correlation coefficient is a statistic used to reflect the degree of linear correlation between two variables. Mutual information methods measure how well a feature relates to a category and then choose the best features through a simple sort according to the measurement results. The spearman correlation coefficient is a nonparametric indicator that measures the dependence of two variables. It uses a monotonic equation to evaluate the correlation of two statistical variables. More recently, a new metric has emerged, called the maximum information coefficient (*MIC*) [17]. The basic idea is that by drawing the variables into a scatter plot, using a grid to segment the space, and then calculating the probability of the scatter points falling across each grid cell, you can identify a broad correlation between two variables. Subsequently, *MIC* has been introduced into feature selection [18] to measure the correlation strength between each feature and label.

Other methods take a higher-level approach and try to determine the best subset of multiple features within the subset space. Heuristic subset search strategies fall into this category. Anbarasi *et al.* [19] used a genetic search followed by a classifier to predict the diagnosis of a patient. Gheyas and Smith [20] proposed a hybrid search algorithm called SAGA, which combines simulated annealing, a genetic algorithm, a generalized regression neural network, and a greedy search algorithm. In addition, in recent years, more and more swarm intelligence algorithms are being combined with feature selection, such as the ant colony optimization algorithm [21], [22] and the artificial bee colony algorithm [23]. Notably, Shokouhifar and Sabet [24] developed a hybrid feature selection method that combines artificial bee colony optimization techniques with neural networks.

Particle swarm optimization (PSO) is an optimization algorithm based on swarm intelligence theory, first proposed by Eberhart and Kennedy in 1995 [25], [26]. Compared with other evolutionary algorithms (EAs), PSO has several advantages including fewer parameters and ease of use. In addition, the PSO algorithm has memory, and the best position of the history of the particle population can be remembered and passed on to other particles. PSO algorithms are largely used to solve continuous optimization problems. However, they are not suitable for discrete optimization problems, which led to the discrete binary PSO algorithm (BPSO).

Quantum-behaved PSO (QPSO) [27] is another extension to the classic PSO that integrates some concepts in quantum physics to update the position of the particles. This algorithm simultaneously considers both the current local and the global optimal position information of each particle during position updates. When QPSO is combined with binary encoding [28], the resulting BQPSO forms a further PSO extension for solving discrete problems. Lin *et al.* [29] further improved BQPSO by introducing variable parameters and a multi-point crossover operator for calculating local attractor coordinates. Xi *et al.* [30] combined BQPSO with the support vector machine (SVM) classifier to study features for gene selection so as to classify cancers. Behjat *et al.* [31] proposed a BQPSO based on a multi-layer perceptron classifier, which not only reduces data dimensionality but also results in higher accuracy in spam detection systems.

TABLE 1. Notations.

Symbol	Meaning
D	dimensions of search space
M	size of particle population
X_i^t	i-th particle position of t-th iteration
V_i^t	i-th particle velocity of t-th iteration
$pbest_i$	best historical position of i-th particle
$gbest$	global optimal position of all particles
$mbest$	average best position of all particles in the population
q_i	local attractor of each particle
β	coefficient of contraction expansion, the only parameter of QPSO
B	sample size
$d_H(\cdot)$	Hamming distance between two particles
$rand$	random number uniformly distributed over [0,1]
ξ	probability of mutation
τ	number of data types

III. HI-BQPSO

A. PRELIMINARY

The following notations are used throughout the paper.

Compared to other EAs, the PSO algorithm has the greatest advantages in terms of its simplicity to implement, fewer parameters to adjust, and no gradient information. Hence, it is widely used in function optimization, neural network training [32], and so on. A great many experiments show that PSO is able to solve a range of optimization problems that genetic algorithms can solve. Compared to the standard PSO, QPSO has more global search capabilities and fewer control parameters, while BQPSO is an improvement over QPSO for solving discrete optimization problems.

1) PSO

As mentioned above, Eberhart and Kennedy *et al.* (1995) were the first to propose PSO. They did this by simulating the foraging behavior of birds and, in so doing, found a solution to continuous function optimization problems. Each bird in the group is abstracted into a particle with no mass or volume, where each is considered to be a feasible solution to the optimization problem. The fitness value of each particle is calculated by a fitness function that determines the quality of the particle. The direction and distance of each particle is controlled by its speed and trajectory. Particles adjust their trajectory with reference to the best particles, and the optimal solution is found through successive iterative searches.

In a D-dimensional search space with M particles, the position of the particle at the t-th iteration is $X_i^t = (x_{i1}, x_{i2}, \ldots, x_{iD})$, and the velocity is $V_i^t = (v_{i1}, v_{i2}, \ldots, v_{iD})$, where $i = 1,2,\ldots,M$. In PSO, each particle records its own best position in the search space so far, i.e., $pbest_i = (P_{i1}, P_{i2}, \ldots, P_{iD})$. The algorithm also records the best position of each particle so far in global terms, i.e., $gbest = (P_{g1}, P_{g2}, \ldots, P_{gD})$. At each iteration, the PSO algorithm updates the position and velocity of the particles according to Eqs. (1) and (2):

$$v_{id}^{t+1} = w * v_{id}^t + c_1 * rand * (P_{id} - x_{id}^t) + c_2 * rand * (P_{gd} - x_{id}^t), \quad (1)$$

$$x_{id}^{t+1} = x_{id}^t + v_{id}^{t+1}, \quad (2)$$

where $d = 1, 2, \ldots, D$, D corresponds to the dimension of the search space, w denotes the predefined constant inertia weight, and c_1 and c_2 represent the acceleration constants. *rand* is a random number that is uniformly generated in the interval [0, 1].

2) QPSO

QPSO incorporates the concept of quantum mechanics into particle evolution, giving rise to a PSO algorithm based on quantum mechanics. In quantum space, the search for particles spans the entire feasible solution space, so the global search performance of the QPSO algorithm is far superior to the classical PSO. In QPSO, the particles have no velocity or trajectory. Rather, the definition of the position of the particle is iteratively updated:

$$q_i = \varphi * pbest_i + (1 - \varphi) * gbest, \quad (3)$$

$$mbest = \frac{1}{M}\sum_{i=1}^{M} pbest_i = (\frac{1}{M}\sum_{i=1}^{M} P_{i1}, \frac{1}{M}\sum_{i=1}^{M} P_{i2}, \ldots, \frac{1}{M}\sum_{i=1}^{M} P_{iD}), \quad (4)$$

$$X_i^{t+1} = q_i \pm \beta * \left|mbest - X_i^t\right| * \ln\left(\frac{1}{u}\right), \quad (5)$$

where φ and u are random numbers uniformly distributed over [0, 1], and *mbest* represents the average best position of all particles in the population. $q_i = (q_{i1}, q_{i2}, \ldots, q_{iD})$ is the local attractor of each particle, which is determined by the particle's *pbest* and *gbest*. β is the contraction expansion coefficient. Compared to PSO, fewer parameters need to be adjusted in the QPSO algorithm.

3) BQPSO

In BQPSO, the position of the particle is defined in binary terms. Therefore, the distance and position transformation in BQPSO need to be redefined. Here, a Hamming distance $d_H(\cdot)$ is introduced to express the distance between two particles. $d_H(\cdot)$ represents the number of different bits between two binary strings. The distance between two particles X_1 and X_2 can be represented as

$$d_H(X_1, X_2) = sum(X_1 \oplus X_2), \quad (6)$$

where $\oplus$ represents an exclusive OR, and $sum(\cdot)$ denotes the number of 1s in the binary string after the statistics XOR.

Because of the different particle encoding methods, the way that *mbest* is calculated is also different from the QPSO algorithm. The value of each dimension in *mbest* is determined by calculating the number of 0s or 1s present in the corresponding bits of all *pbest*. If the count of 0s is greater than the count of 1s, the corresponding bit for *mbest* is 0. If the opposite were true, the corresponding bit of *mbest* would be 1. Further, if the 0 and 1 counts are equal, the corresponding bit for *mbest* should have an equal probability of being either 0 or 1.

In the BQPSO algorithm, $q_i = (q_{i1}, q_{i2}, \ldots, q_{iD})$ is obtained with a crossover operation similar to that of a genetic algorithm. That is, two children are generated from single or multiple points of $pbest_i$ and $gbest$, and one child is randomly selected as the new q_i point.

The new position X_i of the particle is derived from the local attractor q_i with a probability of δ_i:

$$\delta_i = \begin{cases} \dfrac{b}{l_d} \\ \\ 1 \quad \dfrac{b}{l_d} > 1, \end{cases} \tag{7}$$

$$b = \beta * d_H(X_{id}, mbest_d) * ln(\frac{1}{u}), \tag{8}$$

where l_d is the length of the d-th dimension of the particle, β represents the coefficient of the BQPSO algorithm, and u lies within [0, 1], b is rounded and applied to Eq. (7), and X_{id} is updated as follows:

$$X_{id} = Transf(q_{id}, \delta_i), \tag{9}$$

where the $Transf(\cdot)$ function is described as follows: generate a random number *rand*, which is uniformly distributed over [0,1]. Then, if $rand > \delta_i$, each bit of q_i is inverted, otherwise the corresponding bit of X_i is equal to q_i.

B. METHODOLOGY

Our proposed HI-BQPSO algorithm consists of two parts. First, we screen the features using a filtering method to obtain an initial feature subset. We call this step coarse-grained feature selection. Since the existing correlation calculation method, the *MIC* has been proven to be superior to many other correlation calculation methods, including mutual information (MI), pearson, spearman, maximal correlation and so on [17]. Therefore, this paper used *MIC* to calculate the correlation between features and class labels. The next step is fine-grained feature selection, after which the initial feature subset is input into the improved BQPSO algorithm for optimization resulting in the final optimized feature subset. A classifier is then used to judge the pros and cons of the feature subset. The framework of the method is shown in Figure 1.

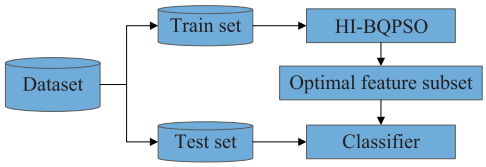


FIGURE 1. Framework of the HI-BQPSO method.

1) COARSE-GRAINED FEATURE SELECTION

The maximum information coefficient (*MIC*) is used to calculate the correlation between the features and class labels. All features are sorted according to *MIC*'s value. A threshold is then set, which removes weakly-correlated features and retains strongly-correlated features. The result is the initial subset of features.

MIC is described as follows. First, a column of features is recorded as a vector, and a column of classes is labeled as a vector. The scalar in a vector of features corresponds to the scalar in a vector of classes to form a sample s. All samples are converted into a scatter plot. Then, according to a given Y row Z column grid, the probability that a scatter point will fall into a particular cell is $P(s \in (y \cup z))$, where the probability that a point will fall into a row is $p(s \in y)$, and $p(s \in z)$ for a column. Once these probabilities are calculated, the mutual information values under the scheme are calculated by

$$I(s)_{Y,Z} = \sum_{y \in Y} \sum_{z \in Z} [P(s \in (y \cup z)) * \log \frac{p(s \in (y \cup z))}{p(s \in y)\, p(s \in z)}]. \tag{10}$$

The mutual information value is then normalized in the interval [0, 1]. To measure the scatter plot with different ranges, the above steps are repeated with multiple different grids. The maximum mutual information values obtained from different grids are then compared, and the largest value is selected as the *MIC*. The resolution of the grid is $Y * Z < B$, where B is 0.6 of the power of the sample size, following [17].

$$m(s)_{Y,Z} = \frac{\max\{I(s)_{Y,Z}\}}{\log(\min\{Y, Z\})}, \tag{11}$$

$$MIC(s) = \max_{Y*Z<B} \{m(s)_{Y,Z}\}. \tag{12}$$

In order to clearly describe the idea of the *MIC*, the basic steps of the process are presented in the following Algorithm 1.

2) FINE-GRAINED FEATURE SELECTION

To enable BQPSO to handle discretization problems with feature selection, features are defined with a binary number encoding. Dimension D of a particle's position is determined by the total number of features in the initialized feature subset, with a value of 0 or 1 for each dimension. 0 means the feature has not been selected, and 1 means it has. For example, $X_i = [1011001]$ means that the feature subset contains seven features. There are four 1s, which means that four features have been selected, and the remaining three have not.

The local attractor q_i in the BQPSO algorithm is generated from the random intersection of the particle's *pbest* and *gbest*. However, this means the particles easily fall into the local best

Algorithm 1 Pseudo-Code for the *MIC* Algorithm

Input: a column of features, a column of classes
Output: *MIC*

1 A scatter plot is drawn from these two column vectors ;
2 Given Y, Z, different grids of Y row and Z column are performed on the scatter plot ;
3 The corresponding mutual information value is calculated by (10) ;
4 Find the largest mutual information value m and normalize it by (11) ;
5 Select more different Y, Z, and repeat steps 2-4 ;
6 Find the largest m value according to (12), which is the *MIC* ;
7 return *MIC* ;

when an optimization problem has multiple optimal extreme points. Therefore, in HI-BQPSO, we calculate q_i using a comprehensive learning strategy [33], which maintains the diversity of the population and improves the convergence performance of the algorithm. An overview of the q_i calculation process follows.

First, a learning probability η_i is introduced to determine whether each dimension in q_i is the *pbest* of the particle itself or of another particle. A random number $rand \in [0, 1]$ is generated during the learning process. If $rand > \eta_i$, the value of the corresponding bit in q_i is its own *pbest*. Otherwise, this value needs to be learned from among the *pbest*s of the other particles. If q_i needs to be learned from other particles, two particles are randomly selected from the population. The fitness values of the two corresponding *pbest*s are then calculated, and the better *pbest* is chosen. q_i is then updated to the chosen *pbest*. If all values of q_i are equal to the value of the particle's own *pbest*, then one dimension is randomly selected from the *pbest*s of other particles and learned.

One potential drawback of the BQPSO algorithm is that the particles tend to converge prematurely, i.e., it falls into the local optimum as the algorithm iterates. To solve this problem, we have incorporated the idea of crossover and mutation from genetic algorithms. In each iteration, once all particles have been updated, a new generation of particles is produced through crossover. The particles are selected and mutated when the fitness value of the particles is greater than the average of the population. The process of this variation is

$$x_{id} = \begin{cases} 1 - x_{id} & rand < \xi \\ x_{id} & otherwise, \end{cases} \tag{13}$$

where ξ is the probability of mutation, and *rand* indicates a random number, which is evenly distributed over [0, 1]. Crossover and mutation operations are effective at increasing the diversity of particle swarms. In the later iterations of the algorithm, they essentially prevent particles from falling into local extreme points and, therefore, premature convergence.

It is important to design a reasonable fitness function in HI-BQPSO because the algorithm evaluates the merits of a particle by calculating its fitness, and an excellent fitness function helps to improve the algorithm's convergence performance. Within this design, we must not only consider the accuracy of the classification, but also the number of selected features. The purpose of feature selection is to select a subset of the most representative features from the original dataset. This feature subset typically contains fewer features but achieves higher classification accuracy. Based on this principle, we designed the following fitness function:

$$fitness = \frac{\left(1 + \theta^2\right) * Acc * norm\left(F_{num}\right)}{\theta^2 * Acc + norm\left(F_{num}\right)}, \tag{14}$$

$$Acc = \frac{\sum_{j=\tau}^{\tau} acc_j}{\tau}, \tag{15}$$

$$acc_j = \frac{\delta_j}{\epsilon_j}, \tag{16}$$

$$norm\left(F_{num}\right) = 1 - \frac{F_{num}}{D}, \tag{17}$$

where τ is the number of classes of the dataset, acc_j is the classification accuracy of class j. The data is classified by the selected features using the support vector machine (SVM) classifier. δ_j means the number of samples that are correctly classified in class j. ϵ_j is equivalent to the total number of samples in class j. Acc represents the average classification accuracy over all categories. This is mainly due to the impact of data imbalance on classification accuracy. F_{num} is the number of features selected for each particle, and D is the dimension of the solution space, that is, the total number of features after initialization. $norm(\cdot)$ is equivalent to normalizing the number of features to be the same as the range of accuracy.

The design principle behind this fitness function is to use a weighted harmonic mean of the classification accuracy rate and the number of features, and adjust the proportions between the accuracy rate and the number of features by a weight of θ. The weight θ becomes smaller as the importance of accuracy increases. At this point, HI-BQPSO has achieved its purpose, that is, higher classification accuracy with fewer features.

With the optimized feature subset *gbest* obtained, all that is left is to evaluate the feature subset with a classifier using the test set. A flowchart of the algorithm is shown in Figure 2.

IV. EXPERIMENT

A. DATASET DESCRIPTION

We used the classifier to verify the performance of feature selection in nine gene expression datasets and 36 benchmark datasets from the UCI Machine Learning Repository [34], as shown in Tables 2 and 3. Two gene expression datasets Colon [35] and Leukaemia [36] were retrieved from the R/Bioconductor packages colonCA and golubEsets, respectively. Adenoma [37], ALL [38], CNS [39], DLBCL [40], Lymphoma [41] and Prostate [42] were downloaded from the Broad Institute Genome Data Analysis Center, which

TABLE 2. Gene expression data.

Dataset	Instances	Features	Summary
Adeno (Adenoma)	36	7457	colon adenocarcinoma (18) and normal (18)
ALL	128	12625	B-cell (95) and T-cell (33)
CNS	60	7129	medulloblastoma survivors (39) and treatment failures (21)
Colon	62	2000	tumour (40) and normal (22)
DLBCL	77	7129	DLBCL patients (58) and follicular lymphoma (19)
Leuk (Leukaemia)	72	7129	ALL (47) and AML (25)
Lym (Lymphoma)	45	4026	germinalcentre (22) and activated B-like DLBCL (23)
Mye (Myeloma)	173	12625	presence (137) and absence (36) of focallesions of bone
Pros (Prostate)	102	12625	prostate (52) and non-prostate (50)

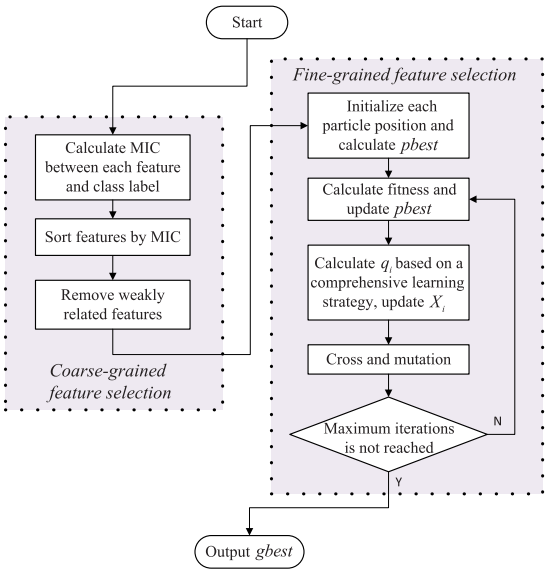


FIGURE 2. Algorithm flowchart.

Algorithm 2 Pseudo-Code for the HI-BQPSO Algorithm

Input: dataset, population size *popsize*
Output: global best position *gbest*

calculate the correlation between each feature and class label based on *MIC* ;
feature ranking and filtering ;
generate an initial feature subset ;
Initialize D, X_i and $pbest_i$;
while *Maximum iterations is not reached* **do**
  for $i = 1$ *to popsize* **do**
    if $f(X_i) < f(pbest_i)$ **then**
      $pbest_i \leftarrow X_i$;
      $f(pbest_i) \leftarrow f(X_i)$;
    end
  end
  update *gbest* ;
  update *mbest* ;
  for $i = 1$ *to popsize* **do**
    compute q_i by comprehensive learning strategy ;
    compute δ_i by (7) ;
    update X_i by (9) ;
  end
  crossover and mutation operations for particles ;
end
return *gbest* ;

is available at http://www.broadinstitute.org/cgi-bin/cancer/datasets.cgi. A further dataset, Myeloma (accession: GDS531) [43], was downloaded from the NCBI Gene Expression Omnibus (GEO) database.

B. EXPERIMENTS PREPARATION

This section focusses on the parameter θ in the fitness function.

1) PARAMETER SETTINGS

To verify the performance of the algorithm, we conducted MATLAB simulation experiments. The initial feature subset size for the gene expression datasets was set to 100; the maximum iterations T_{max} to 100; and the population number was set to

$$popsize = round(12 + \sqrt{2 * D}), \tag{18}$$

where D is the dimension of particles, i.e., the number of features, and the $round(\cdot)$ function is the rounding operation [1]. The learning probability η_i was set to 0.5. The coefficient β of the improved BQPSO algorithm was updated according to the following formula:

$$\beta = (0.8 - 0.6) * \frac{T_{max} - t}{T_{max}} + 0.6, \tag{19}$$

where t is the current number of iterations [29]. The results from the gene expression datasets were obtained from 50 independent runs, with 80% as the training set and 20% as the test set [44]. The results from the UCI datasets were obtained from 100 independent runs, 90% of which were used as the training set and 10% as the test set [45]. In each independent run, the training and test sets were re-classified by category. All algorithms were performed on the same training datasets and evaluated on the same test data.

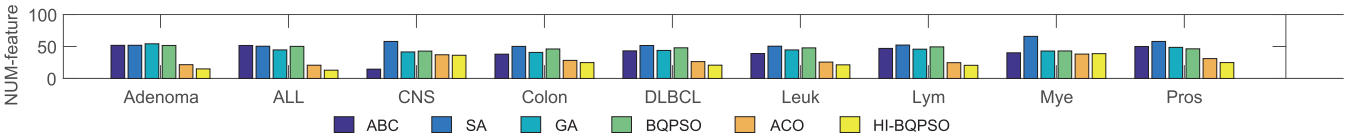


FIGURE 3. A histogram comparing the number of features from HI-BQPSO with the other algorithms on the gene expression data.

TABLE 3. UCI data.

Dataset	Instances	Features	Classes
acute-inflammation (Acu-I)	120	6	2
breast-tissue (Bre-T)	106	9	6
echocardiogram (Echo)	131	10	2
fertility	100	9	2
haberman-survival (Hab-S)	306	3	2
ilpd-indian-liver (IIL)	583	9	2
iris	150	4	3
lenses	24	4	3
monks-1	556	6	2
monks-2	601	6	2
pittsburg-bridges-MATERIAL (PB-M)	106	7	3
pittsburg-bridges-REL-L (PB-R-L)	103	7	3
pittsburg-bridges-TYPE (PB-T)	105	7	6
congressional-voting (Con-V)	435	16	2
glass	214	9	6
heart-cleveland (Hea-C)	303	13	5
heart-switzerland (Hea-S)	123	12	5
statlog-australian-credit (SAC)	690	14	2
statlog-heart (Sta-H)	270	13	2
zoo	101	16	7
horse-colic (Hor-C)	368	25	2
oocytes_trisopterus_nucleus_2f (OTN-2f)	912	25	2
statlog-german-credit (SGC)	1000	24	2
breast-cancer-wisc-prog (BCWP)	198	33	2
breast-cancer-wisc-diag (BCWD)	569	30	2
flags	194	28	8
lung-cancer (Lun-C)	32	56	3
spectf	267	44	2
oocytes_merluccius_nucleus_4d (OMN-4d)	1022	41	2
ozone	2536	72	2
plant-shape	1600	64	100
conn-bench-sonar-mines-rocks (CBSMR)	208	60	2
molec-biol-promoter (MBP)	106	57	2
molec-biol-splice (MBS)	3190	60	3
low-res-spect (LRS)	531	100	9
musk-1	476	166	2

The optimal feature subset was generated once the algorithm completed, then the accuracy of the classification was evaluated on the test dataset. The average accuracy is reported for comparison.

2) EXPERIMENTAL RESULTS ON PARAMETER θ

As previously mentioned, the weight θ in the fitness function is used to measure the accuracy and the quantity of features and needs to be tuned to find a suitable value. The results shown in Table 13 reflect different values of this control parameter at $\theta = [0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1, 2, 4, 6, 8]$. Bold denotes the highest accuracy for the corresponding dataset. Through sufficient experimental data testing, we set θ to 0.3 for the subsequent experiments. A more detailed discussion on tunning this parameter is included in Section V-C.

C. EVALUATION CRITERION

We used several performance metrics to evaluate the selected algorithms. These were average accuracy (ACC), the number of features selected (NUM-feature), F1-scores (F1), Friedman test and two-tailed t-test. The methods for calculating precision (P), recall (R), and F1 from the confusion matrix are shown below.

Label \ Predict	Positive	Negative
Positive	TP	FN
Negative	FP	TN

$$P = \frac{TP}{TP + FP}, \qquad R = \frac{TP}{TP + FN}, \tag{20}$$

$$F1 = \frac{2 * P * R}{P + R}, \tag{21}$$

where the following rules apply:
TP (True Positive): positive samples were judged as positive;
TN (True Negative): negative samples were judged as negative;
FP (False Positive): negative samples were judged as positive;
FN (False Negative): positive samples were judged as negative.

The gene expression datasets are binary classification data and were evaluated using ACC, F1 and Friedman test. The UCI datasets include multivariate classification data and were evaluated using average classification accuracy and two-tailed t-test, as will be explained in detail below.

V. RESULTS AND DISCUSSION

A. GENE EXPRESSION DATA LEARNING TASKS

We first test the performance of the proposed method on nine gene expression data sets. In order to demonstrate the performance of the proposed method, compared to ABC, SA, GA, BQPSO and ACO, our experiments will first report the performance of six methods with respect to classification accuracy (ACC) measured by the support vector

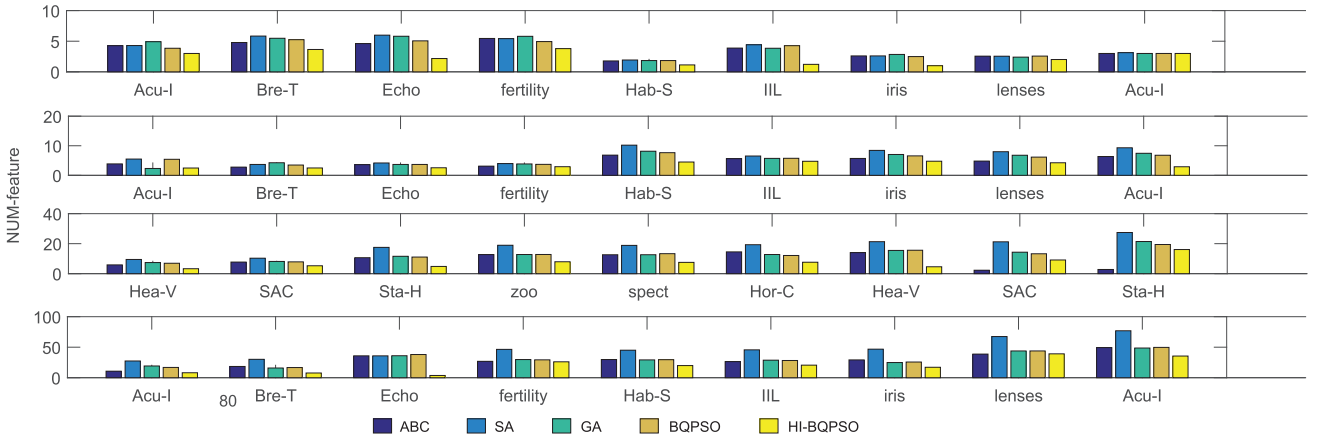


FIGURE 4. A histogram comparing the number of features from HI-BQPSO with the other algorithms on the UCI data.

TABLE 4. HI-BQPSO/baseline comparison with the SVM classifier and the gene expression data.

Dataset	ACC (%)						F1					
	ABC	SA	GA	BQPSO	ACO	HI-BQPSO	ABC	SA	GA	BQPSO	ACO	HI-BQPSO
Adenoma	95.17	99.50	99.50	98.00	99.33	**99.50**	0.9492	0.9945	0.9945	0.9778	0.9945	**0.9945**
ALL	97.19	100.00	99.53	99.37	99.84	**100.00**	0.9814	1.0000	0.9969	0.9958	0.9989	**1.0000**
CNS	65.59	70.63	71.94	72.58	73.17	**73.64**	0.3087	0.2903	0.3636	0.3594	0.4118	**0.4234**
Colon	77.74	86.74	88.03	83.59	86.74	**88.33**	0.8442	0.8972	0.9059	0.8759	0.8962	**0.9082**
DLBCL	83.12	93.33	95.52	92.22	95.54	**95.79**	0.5113	0.8602	0.9119	0.8333	0.9154	**0.9167**
Leuk	84.13	96.11	96.65	93.10	96.08	**97.81**	0.7107	0.9435	0.9512	0.8936	0.9444	**0.9683**
Lym	84.88	94.70	95.76	93.40	**96.66**	94.50	0.8426	0.9444	0.9581	0.9315	**0.9677**	0.9455
Mye	79.06	82.21	**83.72**	82.45	82.90	83.71	0.8819	0.8959	**0.9031**	0.8969	0.8988	0.9020
Pros	90.00	91.38	91.37	90.41	90.80	**91.60**	0.8957	0.9113	0.9116	0.9006	0.9043	**0.9135**
Average rank	6	3	2.44	4.67	3	1.44	5.89	3.33	2.22	4.78	2.78	1.56

machine (SVM) classifier, corresponding F1-score, and the number of features. Finally, we compare related algorithms via Friedman test with $\alpha = 0.05$, 95% confidence. Based on the statistical theory, the difference is statistically significant only if the probability of significant difference is at least the critical value, a more detailed discussion is included in Section V-A2. Our experimental results indicate that the proposed method has very significant gain compared to other methods on the above three evaluation criteria.

1) PERFORMANCE

The results for efficiency are shown in Table 4. The optimal feature subsets for all these algorithms were compared in terms of ACC and the F1. We used nine gene expression datasets, all of which comprised dichotomous and unbalanced data. Therefore, F1 is a suitable metric because it considers both the accuracy and the recall rate of the classification model.

With the SVM classifier, the HI-BQPSO algorithm outperformed the other algorithms on seven of the nine gene expression datasets in terms of both ACC and F1. Further, the results demonstrate that HI-BQPSO was not affected by data imbalances. With the remaining two datasets, HI-BQPSO ranked third on Lym and second on Mye.

TABLE 5. HI-BQPSO/baseline comparison with the gene expression data (number of features).

Dataset	NUM-feature					
	ABC	SA	GA	BQPSO	ACO	HI-BQPSO
Adenoma	51.8	51.88	54.24	51.6	21.56	**14.96**
ALL	51.6	50.4	44.68	50.2	20.72	**12.92**
CNS	**14.56**	57.92	41.48	42.8	37.08	36.28
Colon	38	50.16	40.76	46.12	28.28	**24.8**
DLBCL	43.2	51.56	43.84	47.96	26.36	**20.8**
Leuk	39	50.48	44.64	47.84	25.6	**21.32**
Lym	47	52.24	45.8	49.4	24.68	**20.56**
Mye	40.16	65.84	42.92	43.04	**38.12**	38.8
Pros	50	57.92	48.64	46.36	31.04	**24.84**
Average rank	3.56	5.78	4	4.44	2	1.22

In addition to comparing classification performance, we also compared the size of the optimal feature subset for each algorithm. The results of these experiments are shown in Table 5, which demonstrate that HI-BQPSO performed

TABLE 6. The Friedman test results of three evaluation indicators on the gene expression data.

index	X_f^2	F_F	$F(5,40)$	Significant
ACC	26.59	11.55	2.45	Positive
F1-score	26.21	11.16	2.45	Positive
NUM-feature	35.41	29.55	2.45	Positive

significantly better on eight of the nine gene expression datasets with half and even a third fewer features than the other algorithms. ABC on the CNS dataset was the only comparator to outperform HI-BQPSO. The advantages of selecting fewer features are apparent in Figure 3. The smaller the number of features obtained by feature selection, the smaller the amount of computation for subsequent data processing and the smaller the storage capacity.

2) DISCUSSION

In order to perform a comprehensive comparison of the proposed method and the other algorithms, Friedman test method [46], [47] is used to measure the statistical significance of each evaluation index. Statistics offers more powerful procedures to test the significance of differences between multiple methods. The Friedman test is a non-parametric test for measuring statistical differences between different methods by ranking each algorithm. For each subset of functions, multiple methods are ranked according to different evaluation indicators. In case some methods have the same performance value, the same rank is assigned to them. The Friedman estimator is defined as:

$$F_F = \frac{(N_B - 1)\, X_f^2}{N_B\,(N_M - 1) - X_f^2}, \tag{22}$$

where N_B and N_M are the number of datasets and the number of methods, respectively, and X_f^2 defined is as follows:

$$X_f^2 = \frac{12N_B}{N_M\,(N_M + 1)} \left(\sum_{j=1}^{N_M} R_j - \frac{N_M\,(N_M + 1)^2}{4} \right), \tag{23}$$

where R_j is the ranking of each method. F_F follows a Fisher distribution with $N_M - 1$ and $(N_M - 1)(N_B - 1)$ degrees of freedom. In the experiments, the critical value of the Fisher distribution is set to $\alpha = 0.05$, 95% confidence.

In this paper, the Friedman test is applied to different evaluation indicators of multiple methods, and the results are shown in Table 6. Six of the methods, $N_M = 6$, 9 datasets, $N_B = 9$, degrees of freedom of 5 and 40 (i.e. $N_M - 1 = 5$, $(N_M - 1)(N_B - 1) = 40$), get Fisher distribution F(5,40) = 2.45 critical value. The results show that for the three indicators of accuracy, F1-score and feature subset size, the value of F_F is greater than the critical value of 2.45. This means that all evaluation indicators reject the null hypothesis. Therefore, it can be concluded that the results of the above three indicators are statistically significant.

TABLE 7. HI-BQPSO/baseline comparison with the SVM classifier and the UCI data (ACC %).

Dataset	ABC	SA	GA	BQPSO	HI-BQPSO
Acu-I	99.92	**100.00**	**100.00**	**100.00**	**100.00**
Bre-T	52.43	51.96	53.70	52.30	**54.83**
Echo	83.22	84.15	**84.95**	84.25	82.85
fertility	**87.49**	87.03	87.24	86.42	85.96
Hab-S	72.69	72.92	72.88	**72.98**	72.81
IIL	71.27	**71.31**	71.29	**71.31**	71.15
iris	92.40	93.20	**93.60**	93.13	92.60
lenses	77.83	76.75	79.67	76.67	**85.83**
monks-1	**95.60**	95.40	**95.60**	**95.60**	**95.60**
monks-2	66.76	**69.27**	65.58	68.79	65.78
PB-M	**84.28**	82.56	83.00	84.05	82.74
PB-R-L	65.77	64.75	67.42	**67.62**	66.04
PB-T	51.95	51.82	53.28	54.72	**55.42**
Con-V	**61.72**	61.54	60.90	61.47	61.47
glass	53.01	54.66	**55.29**	53.90	54.02
Hea-C	55.75	56.19	**57.85**	57.24	56.95
Hea-S	37.60	37.90	**39.32**	37.50	37.31
SAC	66.78	**67.34**	67.10	66.51	65.96
Sta-H	80.74	79.48	79.30	80.52	**81.78**
zoo	93.78	93.02	94.68	93.95	**95.47**
Hor-C	81.75	81.07	81.58	82.08	**84.88**
OTN-2f	71.15	70.65	71.14	71.06	**73.15**
SGC	74.41	74.89	74.62	**74.90**	74.80
BCWP	77.71	77.39	78.47	**79.36**	79.16
BCWD	92.09	91.98	92.33	92.33	**92.60**
flags	**57.32**	48.99	47.94	46.20	47.20
Lun-C	**45.75**	39.92	37.58	40.42	39.08
spectf	84.01	83.82	84.31	84.26	**84.39**
OMN-4d	75.81	76.16	76.96	76.83	**77.17**
ozone	**97.12**	**97.12**	**97.12**	**97.12**	**97.12**
plant-shape	17.74	17.77	18.94	19.37	**19.50**
CBSMR	79.39	79.55	80.55	79.59	**81.09**
MBP	81.35	**83.79**	82.85	81.59	82.75
MBS	79.42	78.71	82.09	82.19	**82.79**
LRS	78.94	78.61	78.85	**79.14**	78.09
musk-1	81.56	81.74	81.91	81.20	**82.08**

B. UCI BENCHMARK LEARNING TASKS

In this part of experiment, we report the performance of the proposed method for UCI benchmark learning tasks. The UCI dataset is a standard test dataset for common machine learning and data mining. In addition, we compare the proposed algorithm with ABC, SA, GA and BQPSO. Finally, we compare the related algorithms by a two-tailed t-test.

1) PERFORMANCE

The results of the experiments conducted on the 36 UCI datasets are listed in Table 7. The classification results are reported in terms of ACC with an SVM classifier. The last

TABLE 8. HI-BQPSO/baseline comparison with the UCI data (number of features).

Dataset	ABC	SA	GA	BQPSO	HI-BQPSO
Acu-I	4.27	4.29	4.92	3.85	**3**
Bre-T	4.79	5.84	5.48	5.25	**3.65**
Echo	4.61	5.99	5.81	5.06	**2.17**
fertility	5.45	5.42	5.8	4.94	**3.79**
Hab-S	1.78	1.93	1.84	1.84	**1.13**
IIL	3.88	4.42	3.85	4.27	**1.22**
iris	2.6	2.6	2.84	2.49	**1**
lenses	2.56	2.55	2.4	2.57	**2.01**
monks-1	3	3.13	3	3	**3**
monks-2	3.87	5.5	2.32	5.42	2.47
PB-M	2.78	3.68	4.26	3.48	**2.49**
PB-R-L	3.61	4.18	3.68	3.68	**2.56**
PB-T	3.11	3.97	3.85	3.71	**2.92**
Con-V	6.83	10.2	8.15	7.66	**4.5**
glass	5.64	6.53	5.73	5.76	**4.75**
Hea-C	5.71	8.43	7.05	6.58	**4.77**
Hea-S	4.82	7.99	6.79	6.16	**4.25**
SAC	6.36	9.32	7.45	6.79	**2.9**
Sta-H	5.86	9.47	7.4	6.94	**3.3**
zoo	7.73	10.31	8.15	7.87	**5.27**
Hor-C	10.66	17.53	11.58	11.07	**4.82**
OTN-2f	12.71	18.94	12.77	12.79	**7.94**
SGC	12.62	18.85	12.63	13.33	**7.53**
BCWP	14.53	19.31	12.77	12.12	**7.62**
BCWD	14.03	21.33	15.49	15.62	**4.6**
flags	2.29	21.24	14.31	13.27	9.12
Lun-C	2.79	27.44	21.43	19.44	16.08
spectf	10.69	27.42	19.23	16.85	**8.14**
OMN-4d	18.44	30.14	15.9	16.67	**7.73**
ozone	35.88	35.84	36.04	38	**3.67**
plant-shape	26.85	46.45	29.75	29.3	**26.08**
CBSMR	29.78	45.08	29.2	29.53	**19.88**
MBP	26.38	45.68	28.73	28.22	**20.58**
MBS	29.19	46.78	24.85	25.7	**17.11**
LRS	38.71	67.43	43.86	43.88	39.14
musk-1	49.33	76.96	48.64	49.69	**35.51**

TABLE 9. HI-BQPSO/baseline comparison with the MNB classifier and the UCI data (ACC %).

Dataset	ABC	SA	GA	BQPSO	HI-BQPSO
Acu-I	**82.52**	**82.52**	**82.52**	**82.52**	**82.52**
Bre-T	64.44	65.42	**67.04**	65.90	65.33
Echo	**85.50**	**85.50**	**85.50**	**85.50**	**85.50**
fertility	86.02	86.43	85.86	86.37	**87.00**
Hab-S	73.46	74.01	73.82	74.01	**74.14**
IIL	54.49	53.74	57.99	54.97	**63.99**
iris	95.07	**95.60**	95.40	95.27	95.33
lenses	Nan	Nan	Nan	Nan	Nan
monks-1	**66.24**	66.20	**66.24**	**66.24**	**66.24**
monks-2	64.35	63.35	65.34	63.10	**65.71**
PB-M	82.24	81.54	82.72	81.51	**83.22**
PB-R-L	68.06	68.06	68.29	67.84	**70.06**
PB-T	Nan	Nan	Nan	Nan	Nan
Con-V	56.28	53.31	54.55	53.15	**58.09**
glass	Nan	Nan	Nan	Nan	Nan
Hea-C	52.77	52.95	53.73	54.10	**54.58**
Hea-S	Nan	Nan	Nan	Nan	Nan
SAC	62.82	57.88	61.02	62.95	**67.51**
Sta-H	81.22	82.11	81.11	**82.30**	78.93
zoo	Nan	Nan	Nan	Nan	Nan
Hor-C	77.94	74.57	77.49	76.88	**83.38**
OTN-2f	52.91	52.83	52.90	52.98	**53.46**
SGC	72.65	72.18	72.40	72.35	**72.79**
BCWP	71.75	70.55	71.63	72.57	**73.95**
BCWD	93.68	93.31	93.36	93.30	**95.06**
flags	Nan	Nan	Nan	Nan	Nan
Lun-C	Nan	Nan	Nan	Nan	Nan
spectf	76.98	**77.76**	76.75	77.35	77.12
OMN-4d	62.99	60.71	63.00	62.57	**67.16**
ozone	72.30	72.36	72.24	72.86	**97.00**
plant-shape	50.86	**52.62**	51.38	51.48	51.19
CBSMR	68.09	**68.99**	67.97	67.63	68.17
MBP	83.46	**87.82**	86.78	86.50	86.22
MBS	85.76	89.56	**90.30**	89.96	89.75
LRS	Nan	Nan	Nan	Nan	Nan
musk-1	69.39	**69.56**	69.33	68.61	68.91

column shows the ACC for HI-BQPSO. Bold indicates the highest score. For 17 of the 36 UCI datasets, HI-BQPSO demonstrated the best performance.

In addition, Table 11 illustrate the compared results of two-tailed t-test, where each entry $w/t/l$ means a one-on-one comparison between each of the four algorithms [48], [49]. The value sets represent win (w), tie (t), and loss (l), respectively, and the values themselves represent the number of datasets. Win means the number of datasets where the column algorithm performed better than the row algorithm. Tie indicates the two algorithms performed equally well. Loss means the column algorithm did not perform as well as the row algorithm.

The size of the optimal feature subset produced by the algorithms is also an important factor to consider. The results of this criteria are shown in Table 8. HI-BQPSO produced subsets with significantly fewer features with 32 of the 36 datasets and ranked second on the remaining four. The advantage of a smaller number of features is more clearly shown in the histogram in Figure 4.

TABLE 10. HI-BQPSO/baseline comparison with the KNN classifier and the UCI data (ACC %).

Dataset	ABC	SA	GA	BQPSO	HI-BQPSO
Acu-I	98.99	98.33	97.42	98.17	**99.50**
Bre-T	58.52	60.40	**60.46**	59.57	57.30
Echo	85.05	83.93	84.76	84.60	**85.50**
fertility	**88.11**	**88.11**	**88.11**	**88.11**	**88.11**
Hab-S	73.56	73.81	72.97	73.45	**73.92**
IIL	68.86	68.61	68.91	69.67	**70.21**
iris	95.27	95.07	95.00	94.93	**95.40**
lenses	**66.67**	**66.67**	**66.67**	**66.67**	**66.67**
monks-1	**95.67**	95.40	**95.67**	**95.67**	**95.67**
monks-2	66.25	**68.03**	65.44	67.41	65.96
PB-M	86.12	86.00	**86.86**	86.78	85.04
PB-R-L	66.18	65.70	66.48	66.74	**67.41**
PB-T	55.22	54.79	55.20	54.64	**55.51**
Con-V	**61.61**	61.25	60.92	61.18	61.54
glass	**63.81**	62.27	62.28	62.32	62.51
Hea-C	56.30	57.72	**58.24**	58.21	57.66
Hea-S	37.99	36.53	37.29	35.40	**38.62**
SAC	65.38	64.92	**65.57**	65.10	62.58
Sta-H	80.48	81.74	80.70	81.70	**81.93**
zoo	81.23	**82.22**	81.40	82.07	78.09
Hor-C	83.24	81.26	83.30	83.76	**84.96**
OTN-2f	73.01	**73.81**	73.76	73.63	73.64
SGC	73.50	73.26	73.97	**74.04**	73.28
BCWP	77.50	77.42	77.72	**78.26**	78.11
BCWD	95.92	95.96	**96.12**	95.92	95.91
flags	**50.15**	45.74	45.86	47.10	46.70
Lun-C	33.75	42.67	44.42	41.17	**44.75**
spectf	84.93	85.00	84.90	85.01	**85.35**
OMN-4d	75.27	74.88	75.70	75.52	**76.46**
ozone	97.10	97.07	97.09	97.09	**97.12**
plant-shape	48.98	**49.92**	49.49	49.33	49.13
CBSMR	**75.23**	73.23	74.54	74.33	74.40
MBP	78.10	80.22	**80.95**	79.08	80.41
MBS	73.69	71.80	76.86	76.50	**78.50**
LRS	85.65	85.66	86.17	85.76	**85.90**
musk-1	82.23	82.31	82.51	**82.84**	82.14

2) DISCUSSION

We also compared the optimal feature subsets for all algorithms using different classifiers on the UCI datasets. However, unlike gene expression datasets, the UCI datasets contain both binary and multi-class data, so we chose a multiclass naive Bayes model (MNB) classifier and a K-nearest neighbor (KNN) classifier.

Table 9 shows the results as evaluated by the MNB classifier from MATLAB. HI-BQPSO shows the highest ACC with 17 of the 36 UCI datasets and in some cases significantly better. For example, on the ozone dataset, HI-BQPSO's ACC was 24% higher than the next best algorithm with 10 times fewer features. On the Acu-I and Echo datasets, HI-BQPSO's ACC was comparable to the other algorithms. Similarly, the results in Table 10 using the KNN classifier, show HI-BQPSO had the highest ACC for 19 of the datasets – far more than the comparators. Notably, MATLAB's MNB classifier returned eight invalid datasets among the 36. Error reports show that these datasets have zero variance, namely, the classifier could not classify the samples given the features in the subset.

TABLE 11. Two-tailed t-test for classification accuracy (ACC) of different classifiers.

Classifier	Algorithms	HI-BQPSO	BQPSO	GA	SA
SVM	ABC	**23/2/11**	22/212	24/2/10	16/1/19
	SA	**19/2/15**	22/2/11	24/2/10	
	GA	**19/3/14**	15/4/17		
	BQPSO	**17/4/15**			
MNB	ABC	**23/11/2**	15/11/10	13/11/12	13/11/12
	SA	**18/10/8**	13/11/12	16/10/10	
	GA	**18/11/7**	10/11/15		
	BQPSO	**19/11/6**			
KNN	ABC	**31/2/12**	20/4/12	20/3/13	14/2/20
	SA	**24/2/10**	19/2/15	24/2/10	
	GA	**20/3/13**	15/4/17		
	BQPSO	**21/3/12**			

Table 11 shows the results of the one-on-one comparisons of the five algorithms. According to the t-test results in Tables 11, some detailed explanations can be discussed as follows: In SVM, HI-BQPSO is superior to the other four comparison methods, (23 wins and 11 losses) compared with ABC, (19 wins and 15 losses) compared with SA, (19 wins and 14 losses) compared with GA, (17 wins and 15 losses) compared with BQPSO. In the other two classifiers, HI-BQPSO always keep more winning than losing compared with the other four algorithms. Especially in the MNB classifier, the winning ratio of HI-BQPSO is significantly higher than that of the other four comparison methods. Specifically, (23 wins and 2 losses) compared with ABC, (18 wins and 8 losses) compared with SA, (18 wins and 7 losses) compared with GA, (19 wins and 6 losses) compared with BQPSO.

In summary, HI-BQPSO had the highest ACC on six datasets with three classifiers, 11 datasets with two classifiers and 16 datasets with one classifier.

C. DISCUSSION OF PARAMETER θ

Our novel fitness function can be defined as the harmonic average value of the classification accuracy and the number of features. However, in some cases, higher classification accuracy may be more important to the task at hand. These situations are simply addressed by adjusting the value of θ to less than 1. Conversely, if reducing the number of features is more important, the value of θ should be adjusted to greater

TABLE 12. Comparison of different parameter values for θ in HI-BQPSO with the SVM classifier (ACC %).

θ	ACC-SVM (%)													
	0.1	0.2	**0.3**	0.4	0.5	0.6	0.7	0.8	0.9	**1**	2	4	6	8
Adenoma	**99.50**	98.50	**99.50**	99.00	98.83	97.83	99.00	98.33	97.17	99.00	97.17	96.17	99.00	99.00
ALL	99.52	99.85	**100.00**	**100.00**	99.85	99.54	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**	99.84	99.84	99.84
CNS	72.58	**76.56**	73.64	74.92	74.36	75.28	75.25	72.99	75.56	70.97	71.61	67.29	68.67	68.26
Colon	**88.36**	87.05	88.33	86.77	87.72	87.67	86.77	87.03	86.97	87.41	88.00	88.00	87.33	87.44
DLBCL	95.81	96.33	95.79	96.35	95.80	95.27	95.33	94.20	94.99	**96.80**	92.26	94.51	93.99	93.33
Leuk	97.47	95.81	**97.81**	94.97	96.67	97.20	96.40	96.11	96.06	96.38	96.93	95.28	97.18	96.65
Lym	96.80	93.31	94.50	94.96	92.46	94.30	94.21	94.06	**97.20**	**97.20**	94.70	92.97	94.67	92.21
Mye	**84.74**	84.51	83.71	84.89	84.38	82.99	85.22	84.59	82.71	82.99	80.84	82.19	80.37	81.68
Pros	91.38	90.58	91.60	90.80	91.16	90.60	88.80	**91.77**	89.99	89.41	89.59	88.61	90.03	88.46

TABLE 13. Comparison of parameter θ set to 0.3, 0.1, and 1 in HI-BQPSO with the SVM classifier (ACC %).

θ	ACC-SVM (%)		
	0.1	**0.3**	1
Adenoma	**99.50**	**99.50**	99.00
ALL	99.52	**100.00**	**100.00**
CNS	72.58	**73.64**	70.97
Colon	**88.36**	88.33	87.41
DLBCL	95.81	95.79	**96.80**
Leuk	97.47	**97.81**	96.38
Lym	96.80	94.50	**97.20**
Mye	**84.74**	83.71	82.99
Pros	91.38	**91.60**	89.41

than 1. Our purpose with the above set of experiments was to reduce the number of features somewhat while ensuring that a high level of accuracy was maintained. Therefore, the value we chose for θ was mostly based on classification accuracy.

However, to assess the impact of θ, we conducted a set of experiments with 14 different values of θ, as shown in Table 12. Bold indicates the maximum SVM classification accuracy for each dataset. Among these, the three values 0.1, 0.3, and 1 count the most datasets, which represents the best accuracy across all the datasets. To choose between the three values, we analyzed the results for just these three values, as shown in Table 13. θ with a value of 0.3 produced the highest accuracy on five of the datasets, while 0.1 and 1 only returned the highest accuracy on three datasets. Thus, we set θ to 0.3 for the experiments with this suite of datasets.

VI. CONCLUSIONS

In this paper, we proposed a feature selection method called hybrid improved binary quantum particle swarm optimization (HI-BQPSO). The first step of the method is to filter out some of the features to reduce the dimensionality of high-order datasets. And then improve the BQPSO algorithm to optimizes the remaining feature subsets to further reduce the number of features. The complete learning strategy and the principles of cross-variation we used to improve the BQPSO algorithm are set out in detail, along with our design for a novel fitness function, which includes three parameters: the classification accuracy of the feature subset, the distribution of data samples, and the number of features. To evaluate the effectiveness and robustness of HI-BQPSO compared to other swarm intelligence algorithms, we conducted experiments with nine gene expression datasets and 36 UCI datasets using several different classifiers. The results show that HI-BQPSO has good overall performance, strong searchability, and was able to maintain high efficiency with a range of different classifiers.

ACKNOWLEDGMENT

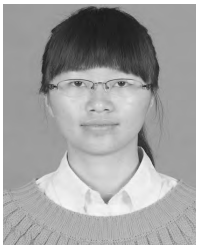
Previous work has accepted as a regular paper and presented at the International Joint Conference on Neural Networks (IJCNN 2018) [1].

REFERENCES

[1] Q. Wu, Y. Shen, Z. Ma, J. Fan, and R. Ge, "iBQPSO: An improved BQPSO algorithm for feature selection," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2018, pp. 1–8.
[2] A. Das and S. Das, "Feature weighting and selection with a Pareto-optimal trade-off between relevancy and redundancy," *Pattern Recognit. Lett.*, vol. 88, pp. 12–19, Mar. 2017.
[3] H. Wang and B. Niu, "A novel bacterial algorithm with randomness control for feature selection in classification," *Neurocomputing*, vol. 228, pp. 176–186, Mar. 2017.
[4] M. Al-Rajab, J. Lu, and X. Qiang, "Examining applying high performance genetic data feature selection and classification algorithms for colon cancer diagnosis," *Comput. Methods Programs Biomed.*, vol. 146, pp. 11–24, Jul. 2017.
[5] H. Dong, T. Li, R. Ding, and J. Sun, "A novel hybrid genetic algorithm with granular information for feature selection and optimization," *Appl. Soft. Comput.*, vol. 65, pp. 33–46, Apr. 2018.
[6] R. Sheikhpour, M. A. Sarram, S. Gharaghani, and M. A. Z. Chahooki, "A survey on semi-supervised feature selection methods," *Pattern Recognit.*, vol. 64, pp. 141–158, Apr. 2017.
[7] J. C. W. Debuse and V. J. Rayward-Smith, "Feature subset selection within a simulated annealing data mining algorithm," *J. Intell. Inf. Syst.*, vol. 9, no. 1, pp. 57–81, 1997.
[8] E. B. Huerta, B. Duval, and J.-K. Hao, "A hybrid GA/SVM approach for gene selection and classification of microarray data," in *Proc. Workshops Appl. Evol. Comput.* Berlin, Germany: Springer, 2006, pp. 34–44.

[9] P. Moradi and M. Rostami, "Integration of graph clustering with ant colony optimization for feature selection," *Knowl.-Based Syst.*, vol. 84, pp. 144–161, Aug. 2015.
[10] D. Karaboga, B. Gorkemli, C. Ozturk, and N. Karaboga, "A comprehensive survey: Artificial bee colony (ABC) algorithm and applications," *Artif. Intell. Rev.*, vol. 42, no. 1, pp. 21–57, 2014.
[11] Y. Liu, G. Wang, H. Chen, H. Dong, X. Zhu, and S. Wang, "An improved particle swarm optimization for feature selection," *J. Bionic Eng.*, vol. 8, no. 2, pp. 191–200, Jun. 2011.
[12] P. Sedgwick, "Pearson's correlation coefficient," *BMJ*, vol. 345, p. e4483, 2012.
[13] M.-T. Puth and M. Neuhäuser, and G. D. Ruxton, "Effective use of Pearson's product-moment correlation coefficient," *Animal Behav.*, vol. 93, pp. 183–189, Jul. 2014.
[14] N. X. Vinh, S. Zhou, J. Chan, and J. Bailey, "Can high-order dependencies improve mutual information based feature selection?" *Pattern Recognit.*, vol. 53, pp. 46–58, May 2016.
[15] W. Gad and S. Rady, "Email filtering based on supervised learning and mutual information feature selection," in *Proc. 10th Int. Conf. Comput. Eng. Syst. (ICCES)*, Dec. 2015, pp. 147–152.
[16] P. Sedgwick, "Spearman's rank correlation coefficient," *Proc. Brit. Med. J.*, vol. 349, p. 7327, Nov. 2014.
[17] D. N. Reshef, Y. A. Reshef, H. K. Finucane, S. R. Grossman, G. McVean, P. J. Turnbaugh, E. S. Lander, M. Mitzenmacher, and P. C. Sabeti, "Detecting novel associations in large data sets," *Science*, vol. 334, no. 6062, pp. 1518–1524, 2011.
[18] C. Lin, T. Miller, D. Dligach, R. M. Plenge, E. W. Karlson, and G. Savova, "Maximal information coefficient for feature selection for clinical document classification," in *Proc. ICML Workshop Mach. Learn. Clin. Data*, Edingburgh, U.K., 2012, pp. 963–970.
[19] M. Anbarasi, E. Anupriya, and N. Iyengar, "Enhanced prediction of heart disease with feature subset selection using genetic algorithm," *Int. J. Eng. Sci. Technol.*, vol. 2, no. 10, pp. 5370–5376, 2010.
[20] I. A. Gheyas and L. S. Smith, "Feature subset selection in large dimensionality domains," *Pattern Recognit.*, vol. 43, no. 1, pp. 5–13, 2010.
[21] S. Tabakhi and P. Moradi, "Relevance-redundancy feature selection based on ant colony optimization," *Pattern Recognit.*, vol. 48, no. 9, pp. 2798–2811, 2015.
[22] S. Tabakhi, P. Moradi, and F. Akhlaghian, "An unsupervised feature selection algorithm based on ant colony optimization," *Eng. Appl. Artif. Intell.*, vol. 32, pp. 112–123, Jun. 2014.
[23] M. Schiezaro and H. Pedrini, "Data feature selection based on Artificial Bee Colony algorithm," *EURASIP J. Image Video Process.*, vol. 2013, no. 1, p. 47, 2013.
[24] M. Shokouhifar and S. Sabet, "A hybrid approach for effective feature selection using neural networks and artificial bee colony optimization," in *Proc. 3rd Int. Conf. Mach. Vis. (ICMV)*, 2010, pp. 502–506.
[25] J. Kennedy, "Particle swarm optimization," in *Encyclopedia of Machine Learning and Data Mining*. Boston, MA, USA: Springer, 2011, pp. 760–766.
[26] R. Poli, J. Kennedy, and T. Blackwell, "Particle swarm optimization," *Swarm Intell.*, vol. 1, no. 1, pp. 33–57, Jun. 2007.
[27] S. N. Omkar, R. Khandelwal, T. V. S. Ananth, G. N. Naik, and S. Gopalakrishnan, "Quantum behaved particle swarm optimization (QPSO) for multi-objective design optimization of composite structures," *Expert Syst. Appl.*, vol. 36, no. 8, pp. 11312–11322, 2009.
[28] J. Sun, W. Xu, W. Fang, and Z. Chai, "Quantum-behaved particle swarm optimization with binary encoding," in *Proc. Int. Conf. Adapt. Natural Comput. Algorithms*. Berlin, Germany: Springer, 2007, pp. 376–385.
[29] H. Lin, X. Maolong, and S. Jun, "An improved quantum-behaved particle swarm optimization with binary encoding," in *Proc. Int. Conf. Intell. Syst. Design Eng. Appl.*, vol. 1, Oct. 2010, pp. 243–249.
[30] M. Xi, J. Sun, L. Liu, F. Fan, and X. Wu, "Cancer feature selection and classification using a binary quantum-behaved particle swarm optimization and support vector machine," *Comput. Math. Methods Med.*, vol. 2016, 2016, Art. no. 3572705.
[31] A. R. Behjat, A. Mustapha, N.-P. Hossein, N. M. Sulaiman, N. Mustapha, "Feature subset selection using binary quantum particle swarm optimization for spam detection system," *Adv. Sci. Lett.*, vol. 20, no. 1, pp. 188–192, 2014.
[32] D. Chen, S. Li, and Q. Wu, "Rejecting chaotic disturbances using a super-exponential-zeroing neurodynamic approach for synchronization of chaotic sensor systems," *Sensors*, vol. 19, no. 1, P. 74, 2018.
[33] L. Wei, R. Fan, and X. Li, "A novel multi-objective decomposition particle swarm optimization based on comprehensive learning strategy," in *Proc. 36th Chin. Control Conf. (CCC)*, Jul. 2017, pp. 2761–2766.
[34] K. Bache and M. Lichman, "UCI machine learning repository," School Inf. Comput. Sci., Univ. California Irvine, Irvine, CA, USA, Tech, Rep., 2013.
[35] U. Alon, N. Barkai, D. A. Notterman, K. Gish, S. Ybarra, D. Mack, and A. J. Levine, "Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays," *Proc. Nat. Acad. Sci. USA*, vol. 96, no. 12, pp. 6745–6750, 1999.
[36] T. R. Golub, D. K. Slonim, P. Tamayo, C. Huard, M. Gaasenbeek, J. P. Mesirov, H. Coller, M. L. Loh, J. R. Downing, M. A. Caligiuri, C. D. Bloomfield, and E. S. Lander, "Molecular classification of cancer: Class discovery and class prediction by gene expression monitoring," *Science*, vol. 286, no. 5439, pp. 531–537, 1999.
[37] D. A. Notterman, U. Alon, A. J. Sierk, and A. J. Levine, "Transcriptional gene expression profiles of colorectal adenoma, adenocarcinoma, and normal tissue examined by oligonucleotide arrays," *Cancer Res.*, vol. 61, no. 7, pp. 3124–3130, 2001.
[38] S. Chiaretti, X. Li, R. Gentleman, A. Vitale, M. Vignetti, F. Mandelli, J. Ritz, and R. Foa, "Gene expression profile of adult T-cell acute lymphocytic leukemia identifies distinct subsets of patients with different response to therapy and survival," *Blood*, vol. 103, no. 7, pp. 2771–2778, 2004.
[39] S. L. Pomeroy *et al.*, "Prediction of central nervous system embryonal tumour outcome based on gene expression," *Nature*, vol. 415, no. 6870, pp. 436–442, 2002.
[40] M. A. Shipp, K. N. Ross, P. Tamayo, A. P. Weng, J. L. Kutok, R. C. T. Aguiar, M. Gaasenbeek, M. Angelo, M. Reich, G. S. Pinkus, T. S. Ray, M. A. Koval, K. W. Last, A. Norton, T. A. Lister, J. Mesirov, D. S. Neuberg, E. S. Lander, J. C. Aster, and T. R. Golub, "Diffuse large B-cell lymphoma outcome prediction by gene-expression profiling and supervised machine learning," *Nature Med.*, vol. 8, no. 1, pp. 68–74, 2002.
[41] A. A. Alizadeh *et al.*, "Distinct types of diffuse large B-cell lymphoma identified by gene expression profiling," *Nature*, vol. 403, no. 6769, pp. 503–511, 2000.
[42] D. Singh, P. G. Febbo, K. Ross, D. G. Jackson, J. Manola, C. Ladd, P. Tamayo, A. A. Renshaw, A. V. D'Amico, J. P. Richie, E. S. Lander, M. Loda, P. W. Kantoff, T. R. Golub, and W. R. Sellers, "Gene expression correlates of clinical prostate cancer behavior," *Cancer cell*, vol. 1, no. 2, pp. 203–209, 2002.
[43] E. Tian, F. Zhan, R. Walker, E. Rasmussen, Y. Ma, B. Barlogie, and J. D. Shaughnessy, Jr., "The role of the Wnt-signaling antagonist DKK1 in the development of osteolytic lesions in multiple myeloma," *England J. Med.*, vol. 349, no. 26, pp. 2483–2494, 2003.
[44] T. Ching, X. Zhu, and L. X. Garmire, "Cox-nnet: An artificial neural network method for prognosis prediction of high-throughput omics data," *PLoS Comput. Biol.*, vol. 14, no. 4, 2018, Art. no. e1006076.
[45] Z. Lu and T. K. Leen, "Semi-supervised learning with penalized probabilistic clustering," in *Proc. Adv. Neural Inf. Process. Syst.*, 2005, pp. 849–856.
[46] M. Labani, P. Moradi, F. Ahmadizar, and M. Jalili, "A novel multivariate filter method for feature selection in text classification problems," *Eng. Appl. Artif. Intell.*, vol. 70, pp. 25–37, Apr. 2018.
[47] P. Moradi and M. Gholampour, "A hybrid particle swarm optimization for feature subset selection by integrating a novel local search strategy," *Appl. Soft. Comput.*, vol. 43, pp. 117–130, Jun. 2016.
[48] L. Jiang and H. Zhang, "Learning instance greedily cloning naive Bayes for ranking," in *Proc. 5th IEEE Int. Conf. Data Mining*, Nov. 2005, p. 8.
[49] J. Wu, S. Pan, X. Zhu, Z. Cai, P. Zhang, and C. Zhang, "Self-adaptive attribute weighting for Naive Bayes classification," *Expert Syst. Appl.*, vol. 42, no. 3, pp. 1487–1502, 2015.

QING WU received the Ph.D. degree in computer science from Zhejiang University, in 2006. He is currently a Professor with the School of Computer Science and Technology, Hangzhou Dianzi University, Hangzhou, China. His current research interests include machine learning, data mining, adaptive software, and ubiquitous computing.



ZHEPING MA received the B.S. degree in computer science and technology from Qingdao University, Qingdao, China, in 2015. She is currently pursuing the M.S. degree in computer technology with Hangzhou Dianzi University, Hangzhou, China. Her research interests include pattern classification, intelligent optimization algorithms, and machine learning.

JIN FAN received the B.Sc. degree from Xi'an Jiaotong University, China, the M.Sc. and Ph.D. degrees from Loughborough University, U.K. She is currently an Associate Professor with the Department of Computer Science and Technology, Hangzhou Dianzi University, China. She has published over 20 technical papers in international journals and conferences. Her research interests include data analytics, wireless sensor networks, and mobile computing and related aspects.

GANG XU received the B.Sc. degree in computational mathematics from Shandong University, in 2003, and the Ph.D. degree in applied mathematics from Zhejiang University, in 2008. He was a Postdoctoral Researcher with INRIA Sophia-Antipolis, from 2008 to 2010. He is currently a Professor with the School of Computer Science and Technology, Hangzhou Dianzi University. His research interests include image processing and computer graphics. He received the Young Scholar Award on Geometric Design and Computing of China, in 2016.

YUANFENG SHEN received the M.S. degree in computer technology from Hangzhou Dianzi University, Hangzhou, China, in 2018. His research interests include feature selection, intelligent optimization algorithms, and machine learning.