Feature Selection and Classification of Microarray Datasets Based on an Improved Binary Harris Hawks Optimization Algorithm

Guoxia LI[1], Wen SHI[2], Jingyu ZHANG[3], Zhixia GU[4], Jixiang XU[5], Yueyue LI[6], Yaxing SUN[7]
School of Information Engineering, Tianjin University of Commerce, Tianjin 300134, China[1,2,4,5,6,7]
Tianjin NanKai Hospital, Tianjin 300100, China[3]

***Abstract*—High-dimensional microarray datasets are prone to the "curse of dimensionality" due to feature redundancy, which impairs the performance of machine learning models, and feature selection is the key to addressing this issue. This study proposes an Improved Binary Harris Hawks Optimization algorithm (IBHHO) for feature selection in high-dimensional microarray data. Core innovations comprise: i) a hybrid filter-wrapper framework integrating a filter method (ReliefF), a wrapper method (HHO) and a classifier (SVM) to simultaneously optimize ReliefF parameters, SVM hyperparameters, and feature subsets; ii) a differentiated exploration–exploitation strategy leveraging HHO's two-stage behavior (global parameter optimization during exploration; feature refinement and local parameter tuning during exploitation); and iii) an elite feature guidance strategy that reduces redundant exploration and accelerates convergence via fixed key-feature anchor points. Experiments conducted on eight public microarray datasets demonstrate that IBHHO reduces feature counts while improving classification accuracy, achieving comprehensive performance superior to benchmark algorithms. Consequently, IBHHO offers an efficient feature selection framework for high-dimensional biomedical data analysis.**

Keywords*—*Microarray dataset; feature selection; parameter optimization; ReliefF

I. Introduction

Technological advancements in the contemporary era have driven revolutionary transformations in data acquisition methodologies, while also engendering the proliferation of redundant or misleading data—this may inadvertently compromise the performance of learning models. Furthermore, coupled with the "curse of dimensionality" [1], the substantial volume of features in datasets frequently induces overfitting in machine learning frameworks.

A domain of particular interest is the analysis of high-dimensional microarray datasets for gene expression profiling, which facilitates the identification of cancer-related genes. Microarray data (obtained from microarray experiments) is typically presented as a two-dimensional table, with rows representing gene expression levels and columns corresponding to samples [2]. Notably, these datasets are characterized by a small number of samples juxtaposed against hundreds of thousands of genes. While microarray datasets have been widely used for cancer classification [3], a large number of genes are irrelevant to disease classification—posing a challenge for understanding target diseases and underscoring the crucial role of feature selection in reducing data dimensionality [4].

The objective of feature selection for microarray data is to identify a set of genes that best discriminates between biological sample types, eliminating irrelevant and redundant information to enhance the efficiency and accuracy of cancer classification algorithms. According to feature evaluation patterns, feature selection algorithms are broadly categorized into three types: filter methods, wrapper methods, and hybrid methods [5].

Filter methods employ fast standard metrics to evaluate feature importance without classifier feedback, offering low computational complexity but being agnostic to classifier types. In contrast, wrapper methods leverage classifier feedback for feature subset selection—their selection process and performance are biased toward the specific classifier, and while they achieve superior classification accuracy, they incur higher computational overhead. The hybrid method integrates wrapper and filter approaches: in the initial stage, a filter method quickly screens out irrelevant features to reduce dimensionality; subsequently, a wrapper method explores the reduced feature space for a more concise subset [6]. It balances computational efficiency and classification accuracy—slightly slower than pure filters but with better classification performance, and significantly less complex than standalone wrappers while achieving comparable or superior results.

Recently, hybrid feature selection strategies have gained traction for high-dimensional datasets (e.g., [7], [8]), typically using a filter to preselect top n features before a wrapper searches for the optimal subset. However, this traditional two-stage strategy has obvious limitations: the filtering stage relies solely on statistical correlation between features and the target, ignoring the classifier's learning needs—potentially discarding features crucial for classification but with insignificant statistical indicators (which cannot be reintroduced in the wrapping stage). Additionally, the manually preset n creates a trade-off: too small restricts the wrapper's search range, while too large increases computational load—this defect is particularly prominent in microarray data. Therefore, this study proposes integrating the filter's initial feature screening into the wrapper stage via a dynamic optimization mechanism. Filter feature evaluation receives classifier feedback to avoid misjudgment; unselected features are retained in a candidate pool for reevaluation during wrapper iteration, enabling dynamic recall of useful features. Meanwhile, filter subset size parameters adapt to different datasets.

Support Vector Machine (SVM) [9] is a widely used

[*] Corresponding authors.

classification technique with proven effectiveness [10]. Beyond feature selection, SVM parameter configuration significantly impacts accuracy, and the core challenge in enhancing SVM's performance lies in simultaneous optimization of feature subsets (discrete 0/1 variables) and SVM parameters (continuous values)—a mixed-variable optimization problem [11].

Metaheuristic algorithms (e.g., Harris Hawks Optimization, HHO [12]) are well-suited for wrapper-based feature selection due to computational efficiency and global search capabilities. HHO mimics Harris hawks' cooperative hunting (exploration, transition, exploitation): the exploration phase avoids local optima via global solution space search, while the exploitation phase refines solution quality through precise local adjustments. Accordingly, this study integrates filter parameter optimization into HHO's exploration phase (aligning feature rankings with classification objectives) and feature subset refinement into the exploitation phase (eliminating redundancy, retaining critical features, and adapting classifier parameters). This phased integration realizes dynamic coordination between parameter tuning and feature screening. Traditional HHO-based feature selection (binary encoding of candidate subsets) lacks clear directional anchors, leading to excessive ineffective searches in redundant feature regions—slowing convergence and risking missed optimal solutions. To address this, we propose an "elite feature guidance mechanism": key features in the top-performing subset of each iteration are marked as "elite" (fixed as selected in subsequent optimizations). These features act as search anchors, reducing ineffective exploration, accelerating convergence to the global optimal subset, and improving selection efficiency and accuracy. Given HHO's advantages in avoiding local optima, this paper proposes an Improved Binary HHO (IBHHO) for feature selection in high-dimensional microarray datasets. The main contributions are summarized as follows:

1) *Hybrid feature selection framework for joint optimization of high-dimensional microarray data:* A filter-wrapper integrated framework enabling concurrent optimization of ReliefF parameters, SVM hyperparameters, and feature subsets. ReliefF's subset size parameter is adaptively tuned based on dataset feature dimensionality.

2) *Co-optimization strategy based on HHO's two-stage characteristics:* A differentiated optimization strategy leveraging HHO's phases: the exploration phase optimizes ReliefF and SVM parameters for comprehensive feature space exploration; the exploitation phase refines feature subsets and fine-tunes classifier parameters—significantly enhancing overall classification performance.

3) *Elite feature guidance strategy:* An "Elite Feature Guidance Strategy" that fixes key features (labeled 1) as strict binary 1s. These anchored features steer the algorithm to converge toward verified optimal subsets, mitigating redundant exploration, accelerating convergence, and improving selection accuracy/efficiency.

IBHHO addresses incoordination between filter and wrapper stages in traditional hybrids and optimization drawbacks of conventional HHO. Tested on eight high-dimensional microarray datasets, it outperforms existing popular meta-heuristic methods—verifying its effectiveness for microarray data feature selection.

The study is structured as follows: Section II reviews related research on feature selection for microarray datasets; Section III elaborates on basic method principles; Section IV introduces the optimization problem model; Section V describes IBHHO's application to feature selection; Section VI evaluates IBHHO using eight public datasets; and Section VII concludes the study.

II. Related Work

A. Feature Selection Methods

Filter-based feature selection methods assess feature importance exclusively based on data-intrinsic properties, disregarding the influence of learning algorithms or classifiers on feature utility. Prominent examples include, in reference [13], employed the ReliefF algorithm to select the top 3% feature subsets in the initial screening stage of feature selection for high-dimensional microarrays. In [14], the authors utilized the Simulated Kalman Filter algorithm to select 1% of features in microarray feature selection. In [15], the authors adopted the SLI-γ filter method to choose the top 1% most relevant features.

Wrapper-based feature selection methods incorporate learning algorithms as integral components, directly leveraging classifier performance as the evaluation metric for feature subset optimization. Wrapper methods typically have higher computational costs than filter methods because they rely on iterative evaluation of classifiers. Prominent examples include, in reference [16], proposes a wrapper feature selection method based on the Chimp Optimization Algorithm. In [17], the authors introduces a Multi-objective Binary Harris Hawks Optimization algorithm to directly optimize feature subsets as a wrapper approach. In [18], the authors employ the Genetic Algorithm as a wrapper method, using the accuracy of the KNN classifier as the fitness function.

For high-dimensional microarray datasets—where feature spaces often comprise thousands to tens of thousands of genes—filter-wrapper hybrid approaches offer distinct advantages over pure wrapper methods [19]. These hybrids integrate the dimensionality reduction efficiency of filter methods with the discriminative power of wrapper optimization, striking a balance between computational feasibility and classification performance.

Recently, feature selection methods based on hybrid filter wrappers have been widely used to speed up the selection process and improve classification performance. In reference [20], an altruism mechanism was integrated into the Whale Optimization Algorithm (WOA) to develop the AltWOA algorithm. First, Pasi Luukka's fuzzy entropy filtering method [21] was employed to screen the top 300 genes by entropy values, eliminating features with low class correlation. Subsequently, the improved WOA—incorporating altruistic behavior—used SVM classification accuracy as the fitness function to perform iterative optimization of the feature subset from the preselected 300 genes. Evaluation on eight high-dimensional microarray datasets demonstrates that this algorithm outperforms other algorithms in terms of accuracy. The hybrid feature selection framework proposed in [22] employs a two-stage strategy. In the filter stage, the minimal redundancy maximal relevance algorithm integrates multiple filtering metrics—including ReliefF, chi-square test, and Kullback-Leibler divergence—to quantify gene relevance, selecting the top 50 features with the highest scores. Subsequently, in the wrapper stage, an improved gray wolf optimizer is deployed to search for the optimal feature combination within the prefiltered candidate set, using SVM classification accuracy as the fitness function. In [23], the authors propose a two-stage gene selection framework for microarray data, integrating anomaly detection with genetic algorithms. In the first stage, an autoencoder is employed to reduce the dimensionality of gene expression data, followed by one-class support vector machine (One-class SVM) for anomalous gene detection, where approximately 1% of features are selected as the candidate subset. The second stage then applies a guided genetic algorithm to the candidate genes, refining them into the final set of effective features through fitness evaluation based on classification accuracy. In [24], the authors propose a novel hybrid algorithm, TRF-WGHC, by integrating a ranking-based filter feature selection algorithm with a greedy hill-climbing algorithm for DNA microarray applications. In the first stage, specific ranking metrics—including information gain, gain ratio, and ReliefF—are employed to select the top n percent of genes, discarding those with scores below a predefined threshold. The second stage then utilizes an augmented greedy hill-climbing algorithm to search for the optimal feature subset from the remaining genes. The experimental part was comprehensively tested on 18 microarray datasets, demonstrating that the algorithm is simple and effective.

B. Joint Optimization

On one hand, the performance of Support Vector Machine (SVM) models is significantly influenced by parameter selection, and applying SVM presents two key challenges: selecting the optimal input feature subset and determining the best parameter values, which are interdependent—the chosen feature subset affects the optimal parameter values, and vice versa. On the other hand, the parameters of the filter method used also impact feature subset selection and classification performance. Therefore, this study focuses on synchronously optimizing three variables: SVM parameters, filter method parameters, and feature selection results. Within the synchronous optimization framework, the feature selection vector consists of binary elements—“1” indicates selecting the corresponding feature, while “0” excludes it [25]; the feature mask for selection is a discrete integer variable, whereas SVM classifier parameters and filter method parameters are continuous variables, making the synchronous optimization of feature selection and SVM parameters a mixed-variable optimization problem. In the study of reference [26], researchers discretized continuous parameters into integers and adopted the Firefly Algorithm to synchronously optimize feature selection and SVM parameters to improve the performance of traditional Chinese medicine prescription classification. In [27], the authors used the Genetic Algorithm with a binary encoding system to design chromosomes containing penalty parameter C, kernel function parameters, and feature masks, realizing the joint optimization of feature selection and parameters. In [25], the authors introduced the Salp Swarm Algorithm to address the joint optimization of feature selection and classifier parameter tuning, partitioning the solution space into a feature mask (discrete part) and SVM parameters (continuous part), which are optimized by binary SSA and standard SSA respectively. In this study, the IBHHO algorithm synchronously optimizes the three variables: during the exploration phase of the algorithm, the filter algorithm and SVM parameters are optimized using the original continuous HHO algorithm; during the exploitation phase, SVM parameters and feature subsets are optimized using both the continuous HHO algorithm and binary HHO algorithm.

C. Summary

In summary, existing filter-wrapper hybrid methods for microarray feature selection typically adopt a two-stage process: pre-screening a fixed number of features using filter methods, followed by wrapper optimization. This rigid architecture has two inherent flaws: 1) Static filtering thresholds cannot adapt to the dynamic changes in feature correlations within microarray data. When processing different microarray datasets, due to the lack of flexibility and adaptability, it is difficult to achieve effective optimization of feature selection and the performance of classification models. 2) Optimizing the filtering phase and the wrapper phase separately may result in a lack of close coordination between them. This can cause the feature subsets initially screened by the filtering algorithm to be affected by redundant features, or lead to the omission of key features due to the limitations of statistical evaluation, ultimately impairing the overall performance of the model. In contrast, this study proposes IBHHO algorithm. By integrating the filtering phase into the two-stage mechanism of HHO, it realizes the combination of the filtering phase and the wrapper phase in the iterative update of the HHO algorithm, while achieving the dynamic adaptive adjustment of the weight parameters of the filtering method. This enables the filtering phase to adaptively adjust the number of feature subsets according to the specific distribution of the dataset, thereby effectively avoiding the risks of over-filtering (losing discriminative features) and redundant retention (retaining irrelevant attributes).

III. Preliminary Knowledge

This section elaborates on the preliminary knowledge employed in this study. First, a feature ranking algorithm based on the ReliefF filter is introduced, which is used for the initial screening of feature subsets. Subsequently, the SVM classifier used in this study and the application process of the original HHO algorithm are presented.

A. ReliefF

A notable strength of ReliefF [28] in feature evaluation is its capacity to capitalize on local feature dependencies while still accounting for the global data distribution. This avoids sacrificing the global perspective by overemphasizing local details [29]. Its core principle involves assessing features according to their capability to discriminate between closely neighboring observation samples via feature values. Specifically, the evaluation unfolds iteratively: in each iteration, an observation sample is randomly drawn from the sample space; next, the k nearest neighbors belonging to the same category and the corresponding nearest neighbors from other categories are identified; subsequently, each feature’s score is updated based on feature-value differences between the target sample and its same-category and different - category nearest neighbors. The weights for each feature in distinguishing the target sample from its neighbors are computed using Eq. (1). Generally, a higher weight indicates greater importance for tasks like category classification, translating to higher utility in model construction and related processes.

$$\begin{aligned} W_f = W_f - \frac{1}{mk}\sum_{j=1}^{k}\text{diff}_f(R_i, H_j) \\ + \frac{1}{mk}\sum_{c\neq \text{class}(R_i)} \frac{P(c)}{1-P(\text{class}(R_i))}\sum_{j=1}^{k}\text{diff}_f(R_i, M_j) \end{aligned} \tag{1}$$

where, W_f is the ReliefF weight of feature f, k denotes the number of nearest neighbor samples, and m represents the number of iterations. R_i is the random sample in iteration i. H_j and M_j are the nearest neighbor samples from the same class and different classes of R_i, respectively. $P(c)$ is the prior probability of class c, which is typically determined based on the training samples fed into the ReliefF algorithm. $\text{diff}_f(\cdot,\cdot)$ is the value difference of feature f between two observations.

For the ReliefF filtering method employed in the IBHHO algorithm proposed in this study, after ranking features by weights using k-nearest neighbors, the top SN features (where, SN denotes the number of features selected by ReliefF) are chosen as the feature subset output in the initial screening stage. The setting equations for k and SN are presented in Eq. (2) and Eq. (3).

$$k = \lfloor \vec{p}\cdot k_{\max} + 0.5\rfloor \tag{2}$$

$$SN = \lfloor \vec{p}\cdot D + 0.5\rfloor \tag{3}$$

where, $\vec{p}$ represents the position vector of k, $k_{\max}$ is typically set to 10, $\vec{p}\cdot k_{\max}$ means scaling each element of vector $\vec{p}$ by $k_{\max}$ times, and $\lfloor \vec{p}\cdot k_{\max} + 0.5\rfloor$ means rounding $\vec{p}\cdot k_{\max}$ to the nearest integer. D represents the feature dimension of the dataset, and $\vec{p}\cdot D$ means scaling each element of vector is scaled by D times to realize parameter adjustment based on the data dimension.

During the filtering stage, the ReliefF algorithm is employed to assess the significance of each feature. Features are then ranked in descending order of their computed importance scores, and the top SN features are forwarded to the wrapper stage. Setting an excessively low threshold may lead to the exclusion of critical features, whereas an overly high threshold could retain numerous redundant or irrelevant features. To address this challenge, our approach dynamically adjusts the threshold, enabling the selection of feature subsets with varying sizes according to the intrinsic importance distribution of features across diverse datasets.

B. SVM

The core idea of Support Vector Machines (SVM) is to map training data into a higher-dimensional space and separate the categories of training data by establishing an optimal hyperplane, thereby transforming non-linearly separable data in the input feature space into linearly separable data in the high-dimensional feature space.

When using SVM as a classifier, the parameter C serves as a key factor in controlling the model's complexity and the severity of misclassification penalties, thereby balancing the model's bias and variance. Additionally, the kernel function parameter determines the nonlinear transformation from the input space to the high-dimensional space, where the hyperplane used for class separation is identified. This study selects the RBF kernel function based on its simplicity and efficiency, as shown in Eq. (4):

$$\varphi(x_i - x_j) = \exp\left(-\gamma\|x_i - x_j\|^2\right), \gamma > 0 \tag{4}$$

where, γ represents the parameter of kernel function. The parameters of support vector machine (SVM) to be optimized include penalty parameter C and kernel function parameter γ.

C. Harris Hawk Optimization

The HHO algorithm consists of three distinct phases: the exploration phase, the transition phase from exploration to exploitation, and the exploitation phase.

Proposed in [12], the Harris Hawk Optimization algorithm (HHO) is a bio-inspired swarm intelligence optimization algorithm renowned for its robust global search capability and high optimization precision. In this framework, the prey (rabbits in natural hunting scenarios) symbolizes the optimal solution with the highest fitness value in the current iteration. The algorithm operates through two overarching stages: exploration and exploitation, with the transition phase mediating between them. During the exploration phase, the algorithm initializes control parameters (escape energy E) to survey the solution space and locate potential prey regions. The exploitation phase comprises four distinct attack strategies, dynamically adjusted based on the prey's perceived "energy" (solution quality) and escape probability. These strategies simulate hierarchical hunting behaviors, enabling progressive refinement of the optimal solution.

1) Exploration phase: When hunting prey, Harris hawks adopt two distinct exploration strategies. Candidate solutions are designed to approach the prey as closely as possible, with the optimal solution representing the target prey itself. First, Harris hawks select a landing point by considering the positions of other hawks and their prey. In the second method, hawks wait on random tall trees. These two approaches are simulated with equal probability q using Eq. (5):

$$x(t+1) = \begin{cases} x_{\text{r}}(t) - r_1\left|x_{\text{r}}(t) - 2r_2x(t)\right| & q \geq 0.5 \\ x_{\text{p}}(t) - x_{\text{m}}(t) - r_3\left(L + r_4(U-L)\right) & q < 0.5 \end{cases} \tag{5}$$

where, the vector $\mathbf{x}(t)$ is the current position of the hawk, $\mathbf{x}(t+1)$ is the hawk's position in the next iteration, $\mathbf{x}_{\text{r}}(t)$ is a randomly selected position from the Harris hawk population, $\mathbf{x}_{\text{p}}(t)$ is the prey position in the current iteration, r_1, r_2, r_3, r_4 are random numbers with a $(0, 1)$ distribution, q is the transition factor controlling the two strategies, and L/U are the lower/upper bounds of variables. $\mathbf{x}_{\text{m}}(t)$ is the average position of the current hawk population, is calculated as Eq. (6):

$$x_{\mathrm{m}}(t) = \frac{1}{N}\sum_{i=1}^{N} x_i(t) \tag{6}$$

where $\mathbf{x}_i(t)$ is the position of each hawk in iteration t, and N denotes the total number of hawks.

2) Transition from exploration to exploitation: The algorithm switches from exploration to exploitation based on the prey's running or escape energy, as defined by Eq. (7):

$$E = 2E_0\left(1 - \frac{t}{Max\_iter}\right) \tag{7}$$

where, E represents the prey's escape energy, E_0 is the initial energy state, which randomly varies within $(-1, 1)$ at each iteration. When $|E| \geq 1$, global search is performed; otherwise, the exploitation phase begins.

3) Exploitation phase: During the exploitation phase $|E| < 1$, Harris hawks raid and capture prey while the prey avoids predation. HHO determines the appropriate position update strategy from the following four attack modes based on the random number $r \in (0, 1)$, parameter E, and the strategy determinant $|E|$: when $|E| \geq 0.5$, Harris hawks choose the soft besiege strategy; otherwise, they adopt the hard besiege strategy.

a) Soft besiege: When $r \geq 0.5$, Harris hawks can capture the prey; otherwise, the hunt fails. When $r \geq 0.5$ and $|E| \geq 0.5$, the prey has sufficient energy to jump to avoid predation, and Harris hawks use the soft siege strategy with prey energy to complete the hunt, as shown in Eq. (8) and Eq. (9):

$$X(t+1) = \Delta X(t) - E|J \cdot X_{\mathrm{p}}(t) - X(t)| \tag{8}$$

$$\Delta X(t) = X_{\mathrm{p}}(t) - X(t) \tag{9}$$

where, $J = 2(1 - r_5)$ is the prey's movement distance in the jump mode, and r_5 is a random number in $(0, 1)$.

b) Hard besiege: When $r \geq 0.5$ and $|E| < 0.5$, the prey has insufficient energy, and Harris hawks adopt the hard siege strategy for rapid predation, as in Eq. (10):

$$X(t+1) = X_{\mathrm{p}}(t) - E|\Delta X(t)| \tag{10}$$

c) Soft besiege with progressive rapid dives: When $r < 0.5$ and $|E| \geq 0.5$, the prey has sufficient energy to escape. Harris hawks then choose the soft siege combined with a progressive dive tactic, as described by Eq. (11). This strategy includes two hunting methods, with the second selected if the first fails:

$$X(t+1) = \begin{cases} Y : X_{\mathrm{p}}(t) - E|JX_{\mathrm{p}}(t) - X(t)| \\ \quad \text{if } f(Y) < f(X(t)) \\ Z : Y + S \times LF(D) \\ \quad \text{if } f(Z) < f(X(t)) \end{cases} \tag{11}$$

where, D is the spatial dimension, S is a $1 \times D$ random vector, f is the fitness function, and LF is the Levy function simulating the prey's jumping behavior.

d) Hard besiege with progressive rapid dives: When $r < 0.5$ and $|E| < 0.5$, the prey lacks energy but has a chance to escape. Harris hawks then employ the hard siege strategy with a progressive dive to reduce the distance to the prey and form an encirclement, as in Eq. (12):

$$X(t+1) = \begin{cases} Y : X_{\mathrm{p}}(t) - E|JX_{\mathrm{p}}(t) - X_{\mathrm{m}}(t)| \\ \quad \text{if } f(Y) < f(X(t)) \\ Z : Y + S \times LF(D) \\ \quad \text{if } f(Z) < f(X(t)) \end{cases} \tag{12}$$

IV. Problem Model

In the feature selection and parameter optimization model of this study, the decision variables revolve around ReliefF feature screening, performance adjustment of the SVM classifier, and determination of the final feature subset. That is, by adjusting variables such as k, SN, C, γ and FS, the value of the objective function f is minimized, thereby achieving a balance between "high classification accuracy" and "small scale of the feature subset". Therefore, the expression of the optimization problem model in this study is [see Eq. (13)]:

$$\text{Min } f(k, SN, C,\ \gamma,\ FS) \tag{13}$$

The decision variables in the optimization problem model include: ReliefF feature screening parameters (k, the number of nearest-neighbor samples, which is used to determine the scale of neighbor samples referenced when calculating feature weights; SN, the scale of the feature subset selected in the initial screening, i.e., the number of features preliminarily screened out), parameters of the SVM classifier (C, the penalty parameter, which is used to control the penalty intensity for misclassified samples; γ, the kernel function parameter, which defines the mapping complexity of the kernel function in the sample space), and feature selection results (discrete variable FS). The value ranges of each decision variable must satisfy the following constraints [see Eq. (14) to Eq. (18)]:

$$k_{\min} < k = \lfloor \vec{p} \cdot k_{\max} + 0.5 \rfloor < k_{\max} \tag{14}$$

$$SN_{\min} < SN = \lfloor \vec{p} \cdot D + 0.5 \rfloor < SN_{\max} \tag{15}$$

$$C_{\min} < C < C_{\max} \tag{16}$$

$$\gamma_{\min} < \gamma < \gamma_{\max} \tag{17}$$

$$FS_i = \begin{cases} 1 & \text{if the } i\text{th feature is selected} \\ 0 & \text{otherwise} \end{cases} \tag{18}$$

specifically, for the ReliefF parameters: $k_{\min} = 0$, $k_{\max} = 10$; $\text{SN}_{\min} = 0$, $\text{SN}_{\max} = 0.005 \cdot D$. For the SVM classifier parameters, $C_{\min} = 0$, $C_{\max} = 1000$; $\gamma_{\min} = 0$, $\gamma_{\max} = 1$. The feature selection result is characterized by the discrete variable FS, where $\text{FS}_i = 1$ indicates that the i-th feature is selected, and $\text{FS}_i = 0$ indicates that the feature is excluded. The constructed "0/1 binary vector" is used to describe the final feature subset, whose dimension is consistent with the number of original features.

The problem model in this study is oriented towards minimizing the objective function f. The objective function and the expression of the optimization goal are as follows [see Eq. (19) and Eq. (20)]:

$$f = \alpha \cdot (1 - acc) + (1 - \alpha) \cdot \frac{SF}{D} \tag{19}$$

$$acc = \frac{TP + TN}{TP + TN + FP + FN} \tag{20}$$

where, α is the balance coefficient ($\alpha = 0.7$), SF is the number of selected features, D is the total number of original features, and acc represents the classification accuracy. The calculation equation is shown in Eq. (20). The model in this study synchronously optimizes continuous and discrete variables, including ReliefF parameters (k, SN), SVM classifier parameters (C, γ), and feature selection results (discrete FS vector). By leveraging the "synergistic effect" between feature subsets and classifier parameters, it more accurately excavates the "classification discriminant information" of the dataset. This overcomes the problems of traditional univariate optimization methods, such as being prone to falling into local optima and the "curse of dimensionality", where computational complexity grows exponentially with the dimension of variables. Ultimately, it achieves the global optimization of classifier performance.

V. IBHHO FOR FEATURE SELECTION

A. Solution Structure

The solution structure of the algorithm in this study is represented by a coordinate array, where each element of the array is a continuous variable with a value range between $[0, 1]$. The solution structure consists of three parts: the parameters k and SN of ReliefF, the parameters C and γ of the SVM classifier, and the feature selection results. Fig. 1 shows a schematic diagram of the solution structure of the algorithm in this study.

For the first part, the parameters k and SN of ReliefF are calculated by Eq. (2) and Eq. (3), with maximum values set to 10 and 0.005, respectively. For the second part, the parameters C and γ of SVM are set to range between $(0, 1000)$ and $(0, 1)$, respectively. The third part represents the result of discrete feature selection. Given that feature selection is inherently discrete and HHO is designed for continuous optimization problems, a discrete version of HHO (BHHO) must be employed. In the BHHO algorithm, the result vector of feature selection is represented by Eq. (21):

$$FS_i = \begin{cases} 1 & \text{if } w_i < g_{i+4} \\ 0 & \text{otherwise} \end{cases} \tag{21}$$

where, w_i is a random number within the interval $(0, 1)$, g_{i+4} representing the $i+4$-th position in the solution structure.

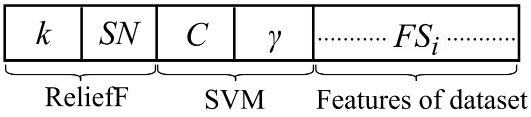


Fig. 1. Solution structure.

B. Fitness function

The fitness function is used to measure the quality of each Harris hawk individual in the IBHHO algorithm. Each Harris hawk represents a potential feature subset along with parameters of the classifier and ReliefF, and the fitness function provides an evaluation criterion for these individuals. The algorithm uses this criterion to select and evolve toward better feature subsets and classifier parameters. Feature selection is a multi-objective optimization problem that should simultaneously consider higher classification accuracy and fewer features. Linear weighting methods can transform multiple objectives into a single fitness value to facilitate algorithmic solution. In this study, a linear combination of classification accuracy and the number of selected features is used as the fitness function, as defined in Eq. (19).

C. Co-Optimization Strategy for Parameters and Features Based on HHO Two-Stage Characteristics

As described in the first subsection, the solution structure of IBHHO is divided into three parts (ReliefF, classifier parameters, and feature selection results). Since the HHO algorithm has both exploration and exploitation components, it effectively balances global search and local search.

The proposed staged optimization strategy fully capitalizes on the complementary characteristics of the HHO algorithm's exploration and exploitation phases:

During the exploration phase, the search space is optimized to identify potential high-quality solution regions. Therefore, ReliefF and SVM parameters are optimized during this phase. By dynamically adjusting the ReliefF parameters, the importance of features is evaluated from multiple perspectives, generating stable feature rankings. Concurrently, SVM parameters are adjusted to rapidly adapt to the data distribution, thereby enhancing model performance;

The development phase focuses on fine-grained search around high-quality solutions. At this stage, feature subset and SVM parameter optimization are performed. Based on the reliable feature subset generated by ReliefF parameters in the exploration phase, the focus shifts to fine-tuning SVM parameters and further controlling the feature subset size, achieving a balance between computational efficiency and optimization accuracy.

This design avoids the computational bottleneck of direct feature selection in high-dimensional data while reducing the search space complexity through separate parameter optimization, providing a new optimization form for the hybrid feature selection framework.

D. Elite Feature Guidance Strategy

In the feature selection process of the original HHO algorithm, the optimization of feature subsets is simulated as the process of hawk flocks chasing prey. The candidate feature subsets correspond to the positions of hawks, while the optimal feature subset is analogized to the prey's position. The algorithm continuously adjusts the positions of hawks through two stages—exploration and exploitation—to approach the prey. During the exploitation phase, hawks mainly update their positions using Eq. (8), Eq. (10), Eq. (11) and Eq. (12), to attempt getting closer to the prey. However, this updating approach in high-dimensional feature spaces may lead to extensive redundant searches in non-critical regions. Moreover, the lack of a protection mechanism for verified high-quality features makes the algorithm prone to missing the true optimal feature subset, resulting in slow convergence and limited result accuracy.

The Elite Feature Guidance Strategy proposed in this study improves the exploitation phase of the original HHO algorithm by introducing pre-identified key features (marked as 1) as fixed "anchor points", significantly optimizing the feature selection optimization process. Specifically, the algorithm directly assigns the "prey positions" of these key features to strict binary value 1, keeping them unchanged throughout the search process. These anchored features act as navigation coordinates, guiding the hawk flock to converge preferentially toward the optimal feature subset containing these key features. During the iterative update process, the algorithm first ensures that the anchored features in the candidate feature subset always remain 1, avoiding the loss of these key features during the search. Eq. (22) shows the position update method after adding the guidance mechanism. Among them, X_{new} is the new prey position (candidate solution), and i is the position in the solution structure array.

When $i < 4$, it represents the first 4 positions of the solution structure, i.e., the ReliefF and SVM parameters, which do not participate in the elite guidance mechanism; when $i > 4$, it represents the position of the feature selection result, which is the part involved in the elite guidance.

$$X'_{\text{p}}(t)_i = \begin{cases} X_{\text{p}}(t)_i & (i \in [0,3]) \\ FS_{i+4} & (i \geq 4) \end{cases} \tag{22}$$

For non-anchored features, during the exploitation phase, the optimization of the feature subset in the new position of the hawk is updated using Eq. (23), Eq. (24), Eq. (25) and Eq. (26), where X'_{rabbit} represents the position of prey containing fixed anchoring features and $X_{\text{new}}(t+1)$ is the updated candidate feature subset.

$$\begin{cases} X_{\text{new}}(t+1) = \Delta' X(t) - E|J \cdot X'_{\text{p}}(t) - X(t)| \\ \Delta' X(t) = X'_{\text{p}}(t) - X(t) \end{cases} \tag{23}$$

$$X(t+1) = X'_{\text{p}}(t) - E|\Delta' X(t)| \tag{24}$$

$$X_{\text{new}}(t+1) = \begin{cases} Y : X'_{\text{p}}(t) - E|J \cdot X'_{\text{p}}(t) - X(t)| \\ \qquad f(Y) < f(X(t)) \\ Z : Y + S \times LF(D) \\ \qquad f(Z) < f(X(t)) \end{cases} \tag{25}$$

$$X_{\text{new}}(t+1) = \begin{cases} Y : X'_{\text{p}}(t) - E|J \cdot X'_{\text{p}}(t) - X_{\text{m}}(t)| \\ \qquad f(Y) < f(X(t)) \\ Z : Y + S \times LF(D) \\ \qquad f(Z) < f(X(t)) \end{cases} \tag{26}$$

In this way, the algorithm always builds upon anchored features when exploring new feature combinations, effectively reducing search attempts in non-critical regions. By fixing key features, this strategy significantly narrows the effective search space and minimizes redundant exploration, thereby enhancing the algorithm's convergence speed.

Meanwhile, since anchored features are pre-verified high-quality features, the algorithm can more accurately approach the true optimal feature subset, ultimately improving the accuracy and efficiency of feature selection–particularly suitable for feature selection tasks on high-dimensional complex datasets.

E. Algorithm Flow of IBHHO

In summary, addressing the challenge of feature selection for high-dimensional microarray datasets, this study proposes three innovative strategies. Firstly, a hybrid feature selection framework integrating filter (ReliefF) and wrapper (SVM) methods is constructed to synchronously optimize ReliefF parameters, SVM hyperparameters, and feature subset size. The feature subset size parameter is adaptively adjusted based on the intrinsic dimensionality of the dataset, enabling dynamic adaptation. Secondly, a differentiated optimization strategy is designed based on the two-stage characteristics of the HHO algorithm: the exploration phase optimizes ReliefF and SVM parameters to comprehensively search the feature space, while the exploitation phase focuses on feature subset refinement and classifier parameter fine-tuning, significantly enhancing classification performance. Finally, an "Elite Feature Guidance Strategy" is proposed, which fixes pre-identified key features as binary value 1 as "anchor points" in the search space. This guides the algorithm to converge rapidly to the optimal feature subset, effectively reducing redundant exploration and substantially improving the efficiency and accuracy of feature selection. The flowchart of the proposed IBHHO algorithm is shown in Fig. 2.

Initialization Phase: The basic parameters of the algorithm (population size N and maximum number of iterations T). Then initialize the positions of hawks, including ReliefF parameters (k and SN), SVM parameters (C and γ), and feature subsets (where binary values indicate whether features are selected).

Fitness Calculation: The fitness is computed using Eq. (19). For the current hawk positions (candidate solutions), feature selection is performed on the microarray dataset based on the current ReliefF parameters to obtain the feature subset SN for SVM training. The selected feature subset SN is used to train the SVM and further refine the optimal subset, after which the classification accuracy acc is calculated. Finally, the fitness value is computed using Eq. (19). A smaller fitness value indicates better comprehensive performance of the candidate solution (higher classification accuracy with fewer features).

Exploration Phase: Execute "broad-spectrum optimization of ReliefF and SVM parameters". The core objective is to simulate the behavior of hawk flocks searching for prey in a vast space using the HHO algorithm, performing extensive and diverse adjustments to ReliefF feature evaluation parameters and SVM classifier hyperparameters. This aims to identify potential high-quality parameter intervals for the subsequent exploitation phase.

Exploitation Phase: Fine-tune SVM parameters and refine feature subsets. In this phase, the Elite Feature Guidance Mechanism is introduced, where features marked as 1 (pre-identified key features) have their "prey positions" directly assigned as strict binary value 1. The positions of hawks are updated using Eq. (23), Eq. (24), Eq. (25) and Eq. (26), guiding the algorithm to converge rapidly to the verified optimal feature subset.

Iteration Termination Criterion ($t < T$?): If $t < T$, the process returns to the fitness calculation phase for the next round of optimization; if $t > T$, the iteration terminates and the optimal solution is output.

VI. Results and Discussion

In this section, we evaluate the performance of the proposed method in feature selection tasks for medical datasets through comparative analysis from different aspects. All experiments were conducted on the MATLAB 2023b platform, with the hardware configuration as follows: Windows 11 operating system, Intel (R) Core (TM) i5-13500H CPU @ 2.6 GHz, and 16.0 GB of RAM.

A. Description of Datasets

To analyze the performance of the proposed method, we utilized eight high-dimensional DNA microarray gene expression datasets, including both binary and multi-class classifications. These datasets were downloaded from two public websites: http://case.szu.cn/stff/zhuzx/Datssets.html and https://github.com/kivancguckiran/microarray-data. Table I provides the specific descriptions of the datasets.

Overall, the 8 microarray datasets selected in this study are distinctly diverse and representative, and they have also been used in existing literature, such as references [30], [31], and [24]. These datasets can comprehensively and rigorously verify the performance of the proposed method: in terms of classification tasks, they include both binary classification (such as Colon Cancer, CNS, Leukemia, Lung cancer, Ovarian cancer, Breast cancer) and multi-class classification (SRBCT with 4 classes, Burcyznski with 3 classes), which can test the adaptability of the method to different category complexities; in terms of sample composition, the total number of samples ranges from 60 (CNS) to 253 (Ovarian); the dimensionality of disease features is distributed in a gradient from 2000 to 22,283, which fully verifies the generalization ability of this method in mining key features and improving classification accuracy. The selection of these datasets provides a comprehensive experimental basis for evaluating the effectiveness and reliability of the method.

TABLE I. Datasets Descriptions

Dataset	Samples	Features	Classification type
Colon Cancer	62	2000	Binary-class
SRBCT	63	2308	Multi-class
Central Nervous System (CNS)	60	7129	Binary-class
Leukemia	72	7129	Binary-class
Lung cancer	181	12533	Binary-class
Ovarian cancer	253	15154	Binary-class
Breast cancer	118	22215	Binary-class
Burcyznski	127	22283	Multi-class

B. Experimental Design and Performance Indicators

To evaluate the performance of machine learning models on different datasets, this study employed k-fold cross-validation. The datasets were further randomly divided into training sets and independent test sets via k-fold cross-validation. The main advantage of cross-validation lies in the independence of each test set, thereby enhancing the reliability of the results. In this study, the value of k was set to five. Thus, the datasets were partitioned into five parts, with each part containing data in equal proportion from each class. Four parts were used for training, while the remaining one part served as the test set.

The analysis of feature selection results for the eight datasets employed multiple evaluation metrics, including accuracy, fitness value, number of selected features, and execution time. These statistical metrics aim to analyze experimental outcomes and determine the algorithm's performance.

To robustly evaluate the reliability of the proposed method, this study is designed to compare all performance indicators of the proposed algorithm with those of all comparative algorithms in the experimental section, and the finally reported results are the average values of 30 independent executions.

C. Overall Performance Evaluation

This section presents a comparison of the IBHHO algorithm with other optimization algorithms in terms of evaluation metrics including classification fitness, accuracy, number of selected features, and execution time. It is divided into two parts: comparison of IBHHO with various optimization algorithms aided by ReliefF, and comparison of IBHHO with various state-of-the-art methods.

1) Comparison of IBHHO with various optimization algorithms aided by ReliefF: In this subsection, four optimization algorithms integrated with ReliefF are employed as comparative algorithms, namely GA [25], PSO [32], FA [33], and WCA [34].

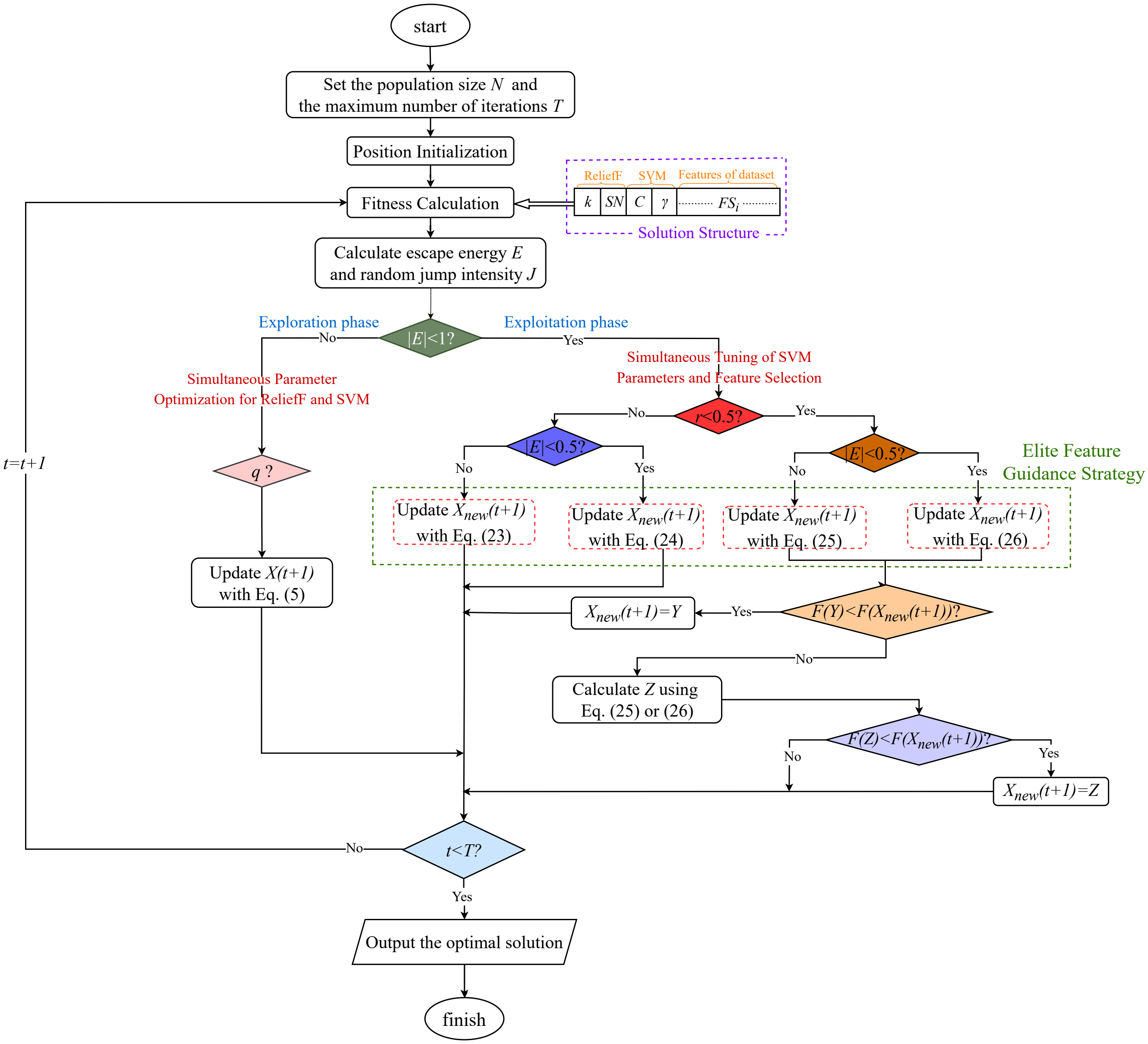

start
Set the population size N and the maximum number of iterations T
Position Initialization
Fitness Calculation
ReliefF
SVM
Features of dataset
k
SN
C
γ
FS_i
Solution Structure
Calculate escape energy E and random jump intensity J
Exploration phase
Exploitation phase
$|E|<1?$
No
Yes
Simultaneous Tuning of SVM Parameters and Feature Selection
Simultaneous Parameter Optimization for ReliefF and SVM
$r<0.5?$
No
Yes
$|E|<0.5?$
No
Yes
$|E|<0.5?$
No
Yes
Elite Feature Guidance Strategy
$t=t+1$
q ?
Update $X_{new}(t+1)$ with Eq. (23)
Update $X_{new}(t+1)$ with Eq. (24)
Update $X_{new}(t+1)$ with Eq. (25)
Update $X_{new}(t+1)$ with Eq. (26)
Update $X(t+1)$ with Eq. (5)
$X_{new}(t+1)=Y$
Yes
$F(Y)<F(X_{new}(t+1))?$
No
Calculate Z using Eq. (25) or (26)
$F(Z)<F(X_{new}(t+1))?$
No
Yes
$X_{new}(t+1)=Z$
No
$t<T?$
Yes
Output the optimal solution
finish

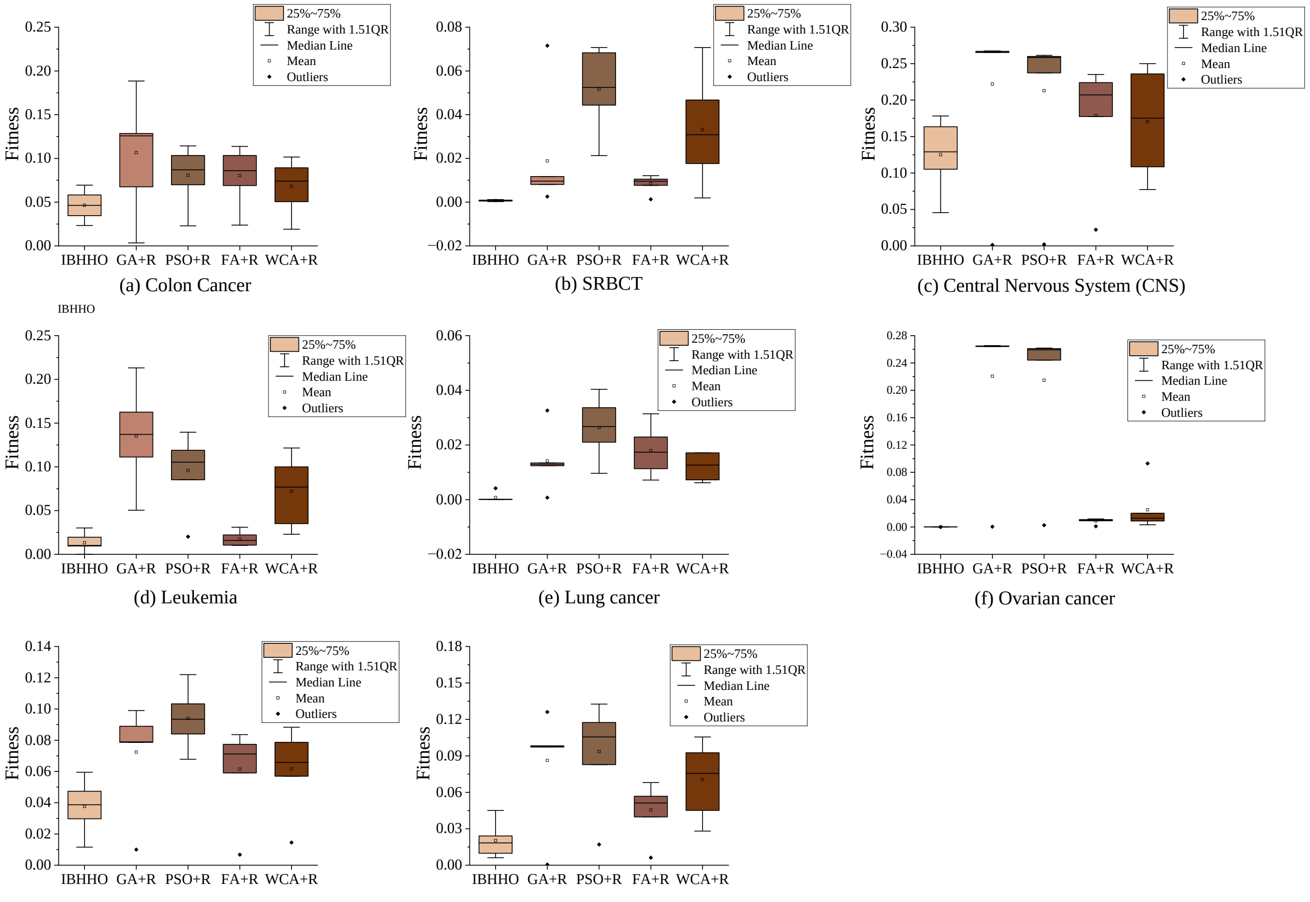
Fig. 2. The flowchart of the IBHHO.

Boxplot analysis is the optimal method to represent the data distribution characteristics of the collected results and identify data anomalies (such as skewness and outliers). A boxplot displays data distribution in the form of different quartiles, specifically the lower (the lowest point/edge of the whisker) and upper (the highest point/edge of the whisker) quartiles, which represent the minimum and maximum values of the data distribution. The lower quartile and upper quartile are indicated by the corners of the rectangle. A small boxplot rectangle indicates a stronger consistency of the data.

To verify the performance of the IBHHO algorithm, four algorithms integrated with ReliefF (GA-R, PSO-R, FA-R, WCA-R) were selected as comparative algorithms. Experiments were conducted on 8 datasets, with fitness as the evaluation metric (a smaller fitness value indicates better optimization performance of the algorithm). The boxplot results are shown in Fig. 3.

On the Colon Cancer dataset, it is intuitively visible that the box of the IBHHO algorithm is the narrowest, which clearly indicates that the fluctuation range of its multiple running results is the smallest, and its stability is the best among the five algorithms. In contrast, the GA+R algorithm has the widest box and the worst stability. The small squares inside the boxes represent the average values of the algorithm runs. By comparison, the small square corresponding to IBHHO is located at the lowest position, which strongly proves that the average value of its fitness is the smallest and its classification performance is more excellent. The performances of the PSO+R, FA+R, and WCA+R algorithms are relatively close. Their box widths are between those of IBHHO and GA+R, with moderate stability, and their mean points are concentrated in a specific interval, with average fitness better than that of GA+R.

On the SRBCT, Lung cancer, and Ovarian cancer datasets,



(a) Colon Cancer

(b) SRBCT

(c) Central Nervous System (CNS)

(d) Leukemia

(e) Lung cancer

(f) Ovarian cancer

(g) Breast cancer

(h) Burcyznski

Fig. 3. Comparison of average fitness among different feature selection optimization algorithms.

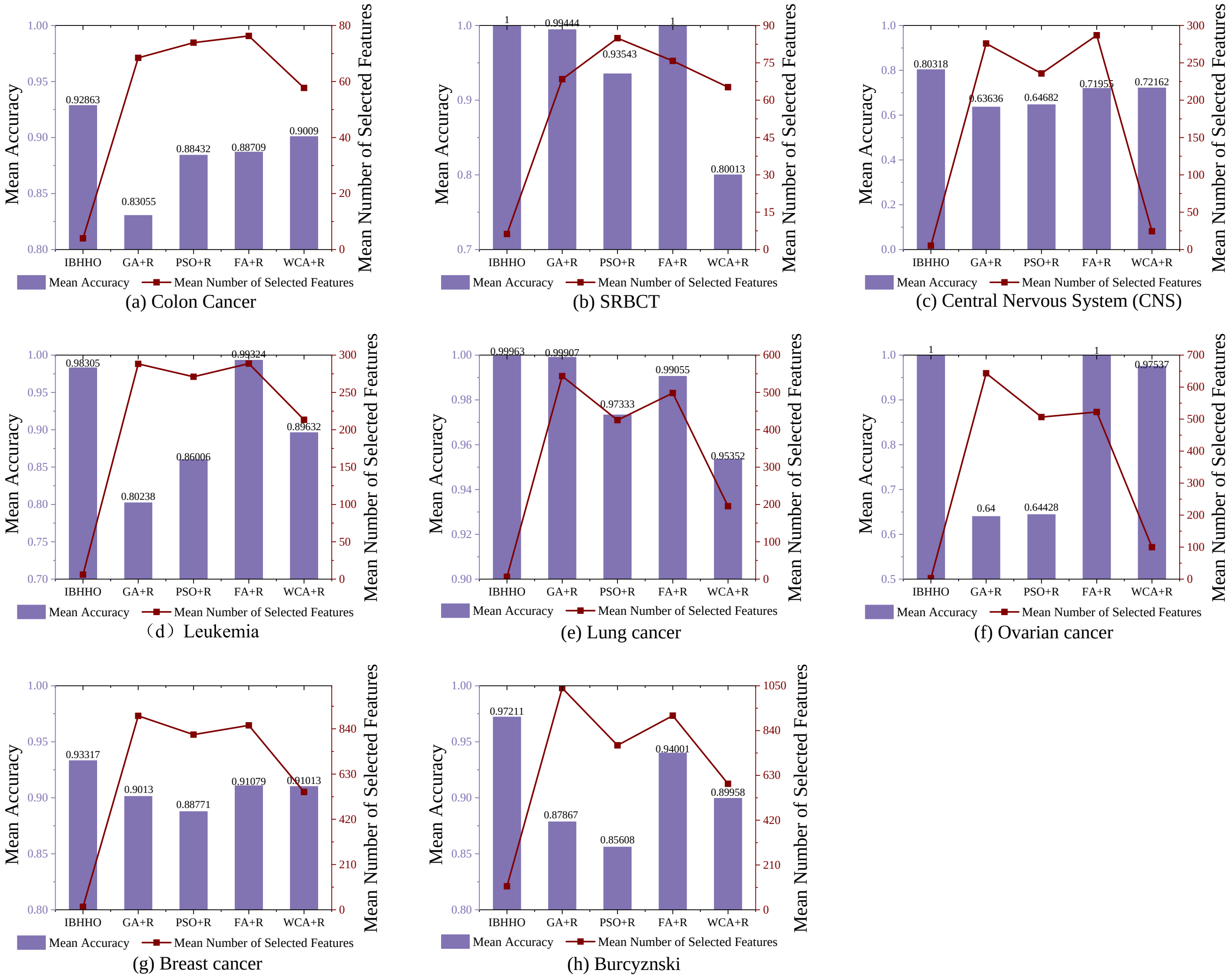
the box of the IBHHO algorithm appears as a thin line. Compared with the other four algorithms, its box has the smallest width, indicating that the results of multiple runs are highly convergent and its stability is optimal. When comparing the small squares representing the average values of each algorithm, the small square corresponding to the IBHHO algorithm has the lowest fitness, demonstrating its best classification performance.

On the Central Nervous System (CNS), Breast cancer, and Burcyznski datasets, the box of the GA+R algorithm is the narrowest, showing the best stability, while the stability of the IBHHO algorithm is relatively inferior on these three datasets. However, when comparing the small squares inside the boxes that represent the average fitness values, the small square of IBHHO is located at the lowest position, whereas that of GA+R is at the highest. This proves that the average fitness of the IBHHO algorithm is the lowest, indicating better fitness performance, while the fitness of the GA+R algorithm is the worst.

From the boxplots of all test datasets, the key statistical metrics such as the median and mean of the IBHHO algorithm's boxplot are significantly lower than those of the comparison algorithms, and the overall distribution of the box is more concentrated in the low fitness interval. This fully indicates that, relying on its unique population update and search strategies, IBHHO can more efficiently traverse the search space and explore low-fitness solutions in different data scenarios. Its optimization stability and effectiveness are superior to those of the comparison algorithms, which strongly verifies that IBHHO exhibits excellent performance across multiple types of datasets.

To verify the comprehensive performance of the IBHHO algorithm in feature selection tasks, experiments were conducted on eight datasets using the proposed algorithm and four comparative algorithms with ReliefF assistance, as shown in Fig. 4. The classification performance was measured by the average classification accuracy of 30 independent runs, and the feature screening efficiency was evaluated by the average number of selected features. The results were visualized through a combination of bar charts and line charts (the purple bars represent the average accuracy, and the red lines represent the average number of selected features).

As shown in the figure, on the four datasets of Colon Cancer, Central Nervous System (CNS), Breast cancer, and Burcyznski, the accuracy of IBHHO is higher than that of the other four comparative algorithms. Moreover, the number of selected features by the IBHHO algorithm is the smallest. This demonstrates that on these four datasets, IBHHO can ensure higher precision while screening key features more efficiently, well-verifying the advantages of the IBHHO algorithm.



(a) Colon Cancer

(b) SRBCT

(c) Central Nervous System (CNS)

（d）Leukemia

(e) Lung cancer

(f) Ovarian cancer

(g) Breast cancer

(h) Burcyznski

Fig. 4. Comparison of the number of features and accuracy.

On the SRBCT and Ovarian cancer datasets, the accuracy of IBHHO is on par with that of the FA algorithm, while the accuracies of the GA+R algorithm and the WCA+R algorithm are slightly lower than those of IBHHO and FA+R, indicating that these three algorithms have advantages in classification accuracy on the SRBCT dataset. However, considering the number of selected features, on the SRBCT dataset, the number of selected features by the IBHHO algorithm (6) is much lower than that by the GA+R and FA+R algorithms (68 and 75 respectively); on the Ovarian cancer dataset, the number of selected features by IBHHO is 2, compared to 522 and 99 by FA+R and WCA+R. This proves that IBHHO can screen key features while ensuring high precision, and performs better in the "classification accuracy-feature efficiency" dimension.

On the Leukemia dataset, the FA+R algorithm has the highest classification accuracy (0.99), with IBHHO being slightly less accurate (also 0.99). However, the number of selected features by IBHHO is 6, while that by the FA+R algorithm reaches 288. This shows that the ability of IBHHO to screen key features is far superior to that of FA+R. Therefore, it can be concluded that IBHHO performs better in meeting the dual requirements of "accuracy+number of selected features".

On the Lung cancer dataset, all algorithms exhibit classification performance with accuracy close to 1. Nevertheless, the number of selected features by the IBHHO algorithm is much lower than that of the comparative algorithms, which proves that the IBHHO algorithm can select the most critical features while ensuring high accuracy.

From the perspective of meeting the dual requirements of "accuracy+number of selected features" across all test datasets, compared with comparative algorithms such as GA+R, PSO+R, FA+R, and WCA+R, the IBHHO algorithm, relying on its unique search and update mechanisms, achieves efficient screening of key features while ensuring high accuracy. This not only reduces model complexity but also verifies the advantages of the IBHHO algorithm.

Table II presents the comparison of the average execution times of the proposed algorithm and comparative algorithms over 30 independent runs under the same operating environment.

As shown in the table, on the two low-dimensional datasets of Colon Cancer (62 samples, 2000 genes) and SRBCT (63 samples, 2308 genes), GA+R achieves the shortest execution time. This is because in low-dimensional spaces, its genetic operations can converge directly, and it rapidly screens features through simple operations such as selection and crossover, thereby reducing redundancies. PSO+R has a slightly longer execution time, which stems from the fact that its particle update rules require more iterations to adjust particle distribution during the initial exploration of small-scale data. IBHHO ranks third with an execution time close to that of PSO+R, mainly because its unique search mechanism involves fine-grained exploration of the solution space during initialization or early iterations, resulting in additional computations and thus a slightly longer execution time.

On two medium-high dimensional datasets, Central Nervous System (CNS) (60 samples, 7129 genes) and Leukemia (72 samples, 7129 genes), PSO+R exhibits the shortest execution time. This is because its particles can rapidly focus on key features in high-dimensional spaces by sharing global optimal information, thus avoiding the computational bottlenecks of GA+R's high-dimensional crossover operations and the exploratory redundancy of IBHHO. IBHHO ranks second with an execution time comparable to those of GA+R and PSO+R, as its strategy to escape local optima during high-dimensional searches introduces additional computational overhead.

On the Lung Cancer dataset (181 samples, 12,533 genes), GA+R runs the fastest. The larger sample size strengthens the selection pressure of GA+R, enhancing its efficiency in eliminating redundant genes and achieving directional evolution in high-dimensional spaces. However, for IBHHO, under high-dimensional and large-sample scenarios, the high computational cost associated with its complex feature screening strategy results in it being slower than both GA+R and PSO+R.

On ultra-large-scale high-dimensional datasets such as Ovarian cancer (253 samples, 15,154 genes), Breast cancer (118 samples, 22,215 genes), and Burcyznski (127 samples, 22,283 genes), IBHHO achieves the shortest running time. Its heuristic search or group-based screening mechanisms can directly skip redundant features, overcoming the computational breakdown issues of GA+R and PSO+R in ultra-high-dimensional spaces, and is particularly adaptable to extreme data scenarios involving multi-class classification and ultra-large-scale genes. In these cases, IBHHO ranks first, while GA+R and PSO+R exhibit significantly longer running times due to the high-dimensional computational complexity and inefficiency in handling multi-class classification tasks, which are much longer than that of IBHHO.

In summary, IBHHO is not always the fastest; instead, its performance varies dynamically with data dimensionality: it is outperformed by GA+R in low-dimensional cases (as GA+R features more straightforward computations), and by PSO+R in medium-to-high-dimensional scenarios (since PSO+R achieves more efficient convergence in high dimensions). However, in ultra-large-scale high-dimensional scenarios, it emerges as the optimal choice by virtue of its unique mechanisms, reflecting its targeted adaptation to different data characteristics. Moreover, on all datasets, IBHHO demonstrates superior classification performance compared to the comparative algorithms.

2) *Comparison between proposed algorithm and various state-of-the-art methods:* To verify the performance of the IBHHO algorithm, four advanced optimization algorithms, namely AltWOA [20], SAGA [35], AIEOU [36], and BSNDO [37], were selected as comparative algorithms. Experiments were conducted on 8 datasets, with all experiments independently executed 30 times under the same environment.

Fig. 5 presents the box plots of the average fitness values obtained from 30 independent runs. On the Colon Cancer dataset, the box of the SAGA algorithm appears as a thin line, indicating a high degree of convergence (minimal fluctuation) among the results of multiple runs and thus demonstrating the best stability among all algorithms. When comparing the mean points (small squares) within the boxes, IBHHO exhibits a significantly lower average fitness value than the other comparative algorithms, indicating the best classification performance. Moreover, the lowest position of IBHHO's box also confirms its lowest fitness value and optimal classification performance.

On the SRBCT, Lung cancer, and Ovarian cancer datasets, the box of the IBHHO algorithm appears as a thin line with the lowest position among all boxes. The small square within its box (representing the mean fitness value) is also lower than those of other comparative algorithms, which proves that IBHHO achieves the best classification performance. Following that, the performances of the SAGA and AltWOA algorithms are second to IBHHO, while the BSNDO algorithm exhibits the worst performance.

On the Central Nervous System (CNS), Breast cancer, and Burcyznski datasets, the SAGA algorithm exhibits the narrowest box, indicating the smallest fluctuation in results across multiple runs and the best stability among all algorithms. However, comparing the mean points (small squares) within the boxes, IBHHO's mean point is positioned lowest, meaning its average fitness value is far superior to SAGA (whose mean point is higher, reflecting the worst fitness performance). This shows that although IBHHO is slightly less stable than SAGA (with a marginally wider box), it achieves a breakthrough in fitness optimization through its unique search strategy. It compensates for the stability difference with a lower average fitness (better classification performance), demonstrating strong adaptability to high-dimensional complex data.

On the Leukemia dataset, the box width of IBHHO is close to that of SAGA (moderate stability), but its mean point is the lowest, showing a significant advantage in fitness. Compared with algorithms like AltWOA, AIEOU and BSNDO, IBHHO is superior in both stability and fitness. Its box is concentrated in the low-fitness interval with short whiskers (data is concentrated without extreme fluctuations), verifying its efficient optimization ability on medium-dimensional data.

Boxplots across all datasets show that statistical metrics such as the median and mean of IBHHO all lie in the lowest interval; its box is overall concentrated in the low-fitness range with short whiskers (indicating high aggregation of non-outlier data). This indicates that through its unique population update and search mechanisms, IBHHO can efficiently traverse the solution space and explore low-fitness solutions across different data scenarios (low-dimensional, medium-to-high-dimensional, and ultra-large-scale high-dimensional), avoiding the drawbacks of comparative algorithms (such as the high fluctuation of SAGA and the local convergence bias of AltWOA). Even in datasets where SAGA dominates in

TABLE II. COMPARISON OF AVERAGE EXECUTION TIME(S)

Dataset	IBHHO	GA+R	PSO+R	FA+R	WCA+R
Colon Cancer	76.1	47.8	75.8	342.8	476.5
SRBCT	205.7	179.5	194.7	935.6	1156.7
Central Nervous System (CNS)	107.4	104.0	103.6	1711.5	103.3
Leukemia	115.5	119.7	111.9	936.8	341.6
Lung cancer	207.5	173.5	197.9	793.6	258.8
Ovarian cancer	283.0	365.6	365.3	920.2	361.3
Breast cancer	209.3	221.3	205.1	1025.7	364.0
Burcyznski	325.7	408.6	408.0	1535.3	277.0

stability (CNS), IBHHO still outperforms with a lower average fitness (a core performance metric), fully verifying its excellent generalization ability and optimization efficiency on multi-type microarray data.

To verify the comprehensive performance of the IBHHO algorithm in feature selection tasks, the algorithm proposed in this study and four comparative algorithms from various state-of-the-art methods were tested on 8 datasets, as illustrated in Fig. 6. The classification performance is evaluated using the average classification accuracy from 30 independent runs (represented by blue bar charts), while the feature selection efficiency is depicted by orange curves (showing the average number of selected features). This dual-metric visualization (precision+feature count) precisely assesses the algorithm's balance between "classification accuracy" and "feature selection efficiency".

On the Colon Cancer and SRBCT datasets, IBHHO achieves the best classification accuracy (reaching above 0.9), while other algorithms are around 0.8. Looking at the number of selected features, although GA selects fewer key features (1 and 4, respectively, compared to IBHHO's 4 and 6), its classification accuracy is lower. Overall, IBHHO demonstrates a better balance between "accuracy" and "efficiency". On the Central Nervous System (CNS), Lung cancer, Ovarian cancer, and Breast cancer datasets, the IBHHO algorithm achieves the best classification accuracy, with SAGA being slightly inferior. However, when comparing the number of selected features, the number of key features selected by the IBHHO algorithm is less than that selected by SAGA, which proves that IBHHO can search for more critical features while ensuring high accuracy.

On the Leukemia dataset, IBHHO achieves a classification accuracy of 0.98 with only 6 selected key features, demonstrating the best balance performance between "accuracy and feature selection efficiency". In contrast, the BSNDO algorithm exhibits a classification accuracy of only 0.62 while selecting up to 3519 key features, thus showing the worst performance in balancing "accuracy and number of features". On the Burcyznski dataset, the AltWOA algorithm selects the fewest features (26), while IBHHO selects 110 features. However, in terms of classification accuracy, IBHHO reaches 0.97, whereas AltWOA only achieves 0.52. Therefore, IBHHO still delivers the best comprehensive performance on the Burcyznski dataset.

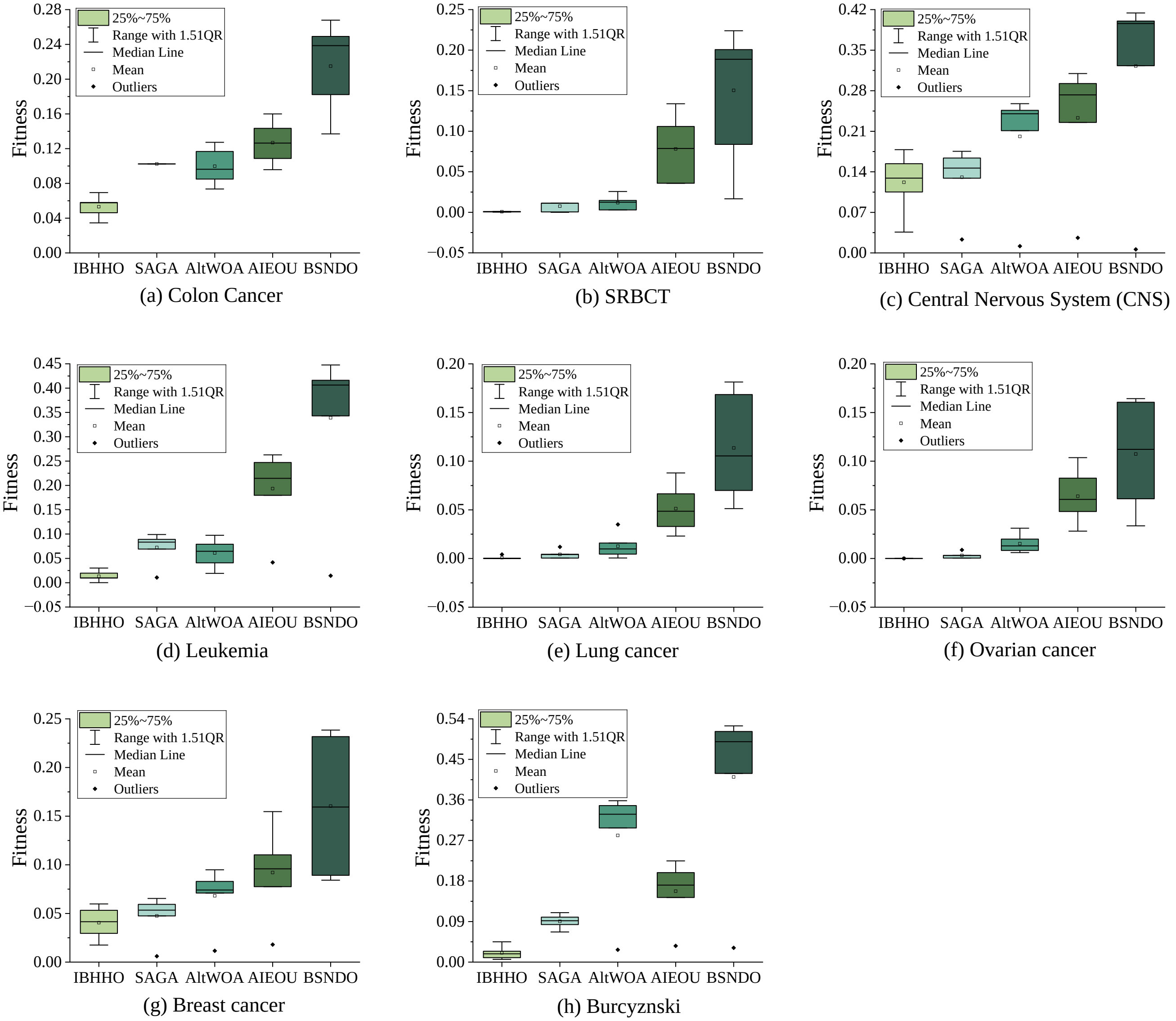
The above results fully demonstrate IBHHO's advantages in simultaneously maximizing classification accuracy and minimizing the number of features. Through efficient search mechanisms, this method can generate sparse yet high-quality feature sets, reducing model complexity while enhancing performance. This makes IBHHO a powerful tool in the field of bioinformatics—its unique advantages are particularly prominent in research scenarios where both accuracy and feature efficiency are equally emphasized to obtain reliable biological insights (such as identifying the minimal gene panel required for cancer diagnosis).

Table III presents a comparison table of the average execution time from 30 independent runs of the proposed algorithm and the comparative algorithms under the same operating environment. As can be observed from the table, on the Colon cancer (62 samples, 2000 features) and SRBCT (63 samples, 2308 features) datasets, which belong to low-dimensional spaces, the simple search strategy of BSNDO avoids the fine-grained search overhead during the initialization of IBHHO and converges directly, resulting in shorter computation time. For the Central Nervous System (CNS) dataset (60 samples, 7129 features), when the sample size is limited and the feature dimension is relatively high, the heuristic search of IBHHO avoids the blind exploration of the particle swarm of AltWOA in the initial iteration and directly converges to the key features.

On the Leukemia (72 samples, 7129 features), Lung cancer (181 samples, 12,533 features), and Breast cancer (118 samples, 22,215 features) datasets, where the sample size is relatively small but the feature dimension is extremely high, AltWOA exhibits faster running speed. This is likely due to the optimizations for high-dimensional computational efficiency in its improved version (AltWOA), which may adopt feature subspace operations or highly optimized vector computation implementations. These optimizations significantly reduce the constant factor overhead of high-dimensional vector operations in each iteration.

On the Ovarian cancer (15,154 features, 253 samples) and Burcyznski (22,283 features, 127 samples) datasets, where both sample sizes and feature dimensions are higher, IBHHO runs faster. Its advantage may lie in its more complex and adaptive exploration-exploitation balance strategies. These strategies allow IBHHO to more effectively utilize information and accelerate the convergence process in the more complex search spaces induced by increased sample sizes, thereby gaining a significant advantage in the total number of iterations and offsetting the potentially higher computational cost of its single iteration.

To summarize, in low-dimensional spaces, although IBHHO is not the fastest in such scenarios, its fitness values



(a) Colon Cancer

(b) SRBCT

(c) Central Nervous System (CNS)

(d) Leukemia

(e) Lung cancer

(f) Ovarian cancer

(g) Breast cancer

(h) Burcyznski

Fig. 5. The box plots of the average fitness values obtained from 30 independent runs.

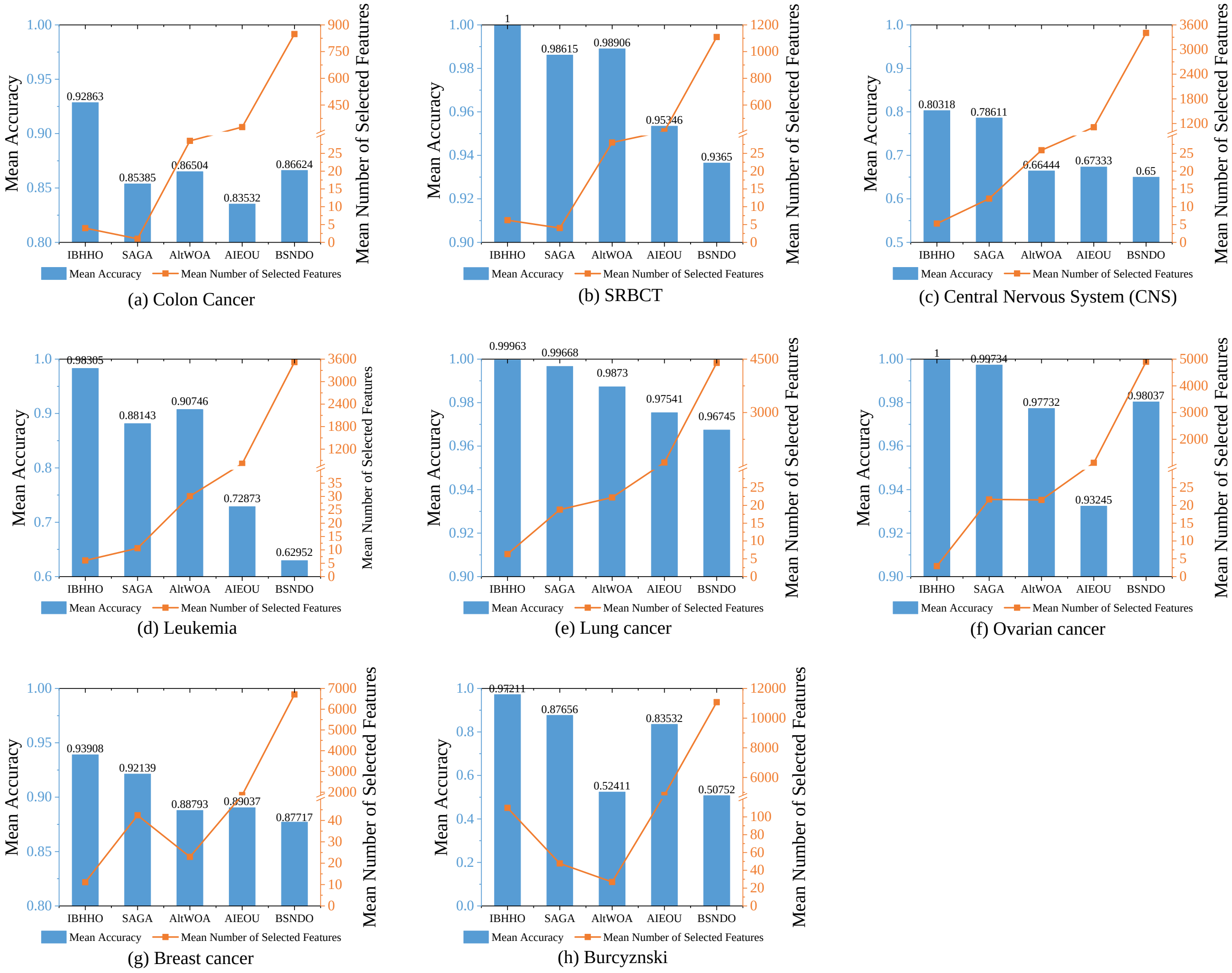
and classification performance (classification accuracy, number of selected features, etc.) far surpass those of BSNDO, demonstrating decisive advantages. In high-dimensional small-sample scenarios (Leukemia, Lung cancer, and Breast cancer), AltWOA achieves the fastest running speed by significantly reducing the constant factor overhead of each iteration through feature subspace operations and vectorized computation optimizations. However, IBHHO still maintains acceptable timeliness (slightly slower than AltWOA) while comprehensively outperforming AltWOA in both fitness and classification performance. In high-dimensional large-sample datasets (Ovarian cancer and Burczynski), IBHHO relies on adaptive exploration-exploitation strategies to efficiently guide the search direction in complex solution spaces, surpassing AltWOA in speed with fewer iterations while maintaining a leading position in classification performance.

TABLE III. Comparison of Average Execution Time(s) from 30 Independent Runs

Dataset	IBHHO	SAGA	AltWOA	AIEOU	BSNDO
Colon Cancer	76.1	178.7	65.0	64.9	62.0
SRBCT	205.7	133.3	100.4	266.6	81.1
Central Nervous System (CNS)	107.4	510.5	163.2	299.3	208.9
Leukemia	115.5	464.9	95.5	204.0	201.8
Lung cancer	207.5	3262.9	68.1	648.7	528.7
Ovarian cancer	283.0	3382.5	376.3	799.2	894.7
Breast cancer	209.3	12832.0	93.6	839.2	735.4
Burcyznski	325.7	11428.3	501.5	827.3	714.6

VII. Conclusion

This study aims to address the challenge of feature selection in high-dimensional microarray datasets, where redundant



(a) Colon Cancer

(b) SRBCT

(c) Central Nervous System (CNS)

(d) Leukemia

(e) Lung cancer

(f) Ovarian cancer

(g) Breast cancer

(h) Burcyznski

Fig. 6. Comparison chart of the number of features and accuracy.

features and the “curse of dimensionality” severely hinder the performance of machine learning models. To tackle this issue, IBHHO is proposed, which integrates a hybrid feature selection framework, a two-stage parameter-feature co-optimization strategy, and an elite feature guidance mechanism. By fusing filter (ReliefF) and wrapper (SVM) methods, this framework enables the simultaneous optimization of ReliefF parameters, SVM hyperparameters, and feature subsets. Leveraging the two-stage characteristics of HHO (exploration and exploitation), the algorithm focuses on optimizing ReliefF and SVM parameters during the exploration phase, while refining feature subsets in the exploitation phase, thus achieving a balance between global search and local optimization. The elite feature strategy further accelerates the algorithm’s convergence speed and reduces redundant exploration by retaining key features as fixed anchors. These innovations collectively enhance IBHHO’s ability to identify optimal feature subsets, not only improving classification accuracy but also effectively reducing feature dimensionality. Experimental results demonstrate that IBHHO outperforms other comparative algorithms in terms of comprehensive performance across eight microarray datasets, verifying its effectiveness in handling high-dimensional biological data. In future, it will also be possible to combine bioinformatics knowledge to conduct in-depth analyses of the biological relevance of feature subsets selected by IBHHO, providing more specific guidance for research on cancer gene functions.

References

[1] N. Altman and M. Krzywinski, “The curse(s) of dimensionality,” Nature Methods, vol. 15, no. 6, pp. 399-400, 2018.

[2] F. Alharbi and A. a.-O. Vakanski, “Machine Learning Methods for Cancer Classification Using Gene Expression Data: A Review,” Bioengineering (Basel, Switzerland), vol. 10, no. 2, p. 173, 2023.

[3] Kim, Jingeun, Yourim Yoon, Hye-Jin Park, and Yong-Hyuk Kim, “Comparative Study of Classification Algorithms for Various DNA Microarray Data,” Genes, vol. 13, no. 3, p. 494, 2022.

[4] H. Jiang, Y. Yang, Q. Wan, and Y. Dong, “Feature selection based on dynamic crow search algorithm for high-dimensional data classification,” Expert Systems with Applications, vol. 250, p. 123871, Sep. 2024.

[5] K. Balakrishnan and R. Dhanalakshmi, “Feature selection techniques for microarray datasets: a comprehensive review, taxonomy, and future directions,” Frontiers of Information Technology & Electronic Engineering, vol. 23, no. 10, pp. 1451-1478, 2022.

[6] M. A. Ganjei and R. Boostani, “A hybrid feature selection scheme for high-dimensional data,” Engineering Applications of Artificial Intelligence, vol. 113, p. 104894, 2022.

[7] R. S. Preyanka Lakshme and S. Ganesh Kumar, "Feature selection using binary horse herd optimization algorithm with lightGBA ensemble classification in microarray data," Knowledge-Based Systems, vol. 312, p. 113168, Mar. 2025.

[8] Y. Wang, W. Li and T. Li, "Single-stage filter-based local feature selection using an immune algorithm for high-dimensional microarray data," Applied Soft Computing, vol. 172, p. Article 112895, 2025.

[9] C. Cortes and V. Vapnik, "Support-vector networks," Machine learning, vol. 20, pp. 273-293, 1995.

[10] H. Faris, M. A. Hassonah, A. M. Al-Zoubi, S. Mirjalili and I. Aljarah, "A multi-verse optimizer approach for feature selection and optimizing SVM parameters based on a robust system architecture," Neural Computing and Applications, vol. 30, no. 8, pp. 2355-2369, 2018.

[11] F. Wang, H. Zhang and A. Zhou, "A particle swarm optimization algorithm for mixed-variable optimization problems," Swarm and Evolutionary Computation, vol. 60, p. 100808, 2021.

[12] A. A. Heidari, S. Mirjalili, H. Faris, I. Aljarah, M. Mafarja and H. Chen, "Harris hawks optimization: Algorithm and applications," Future Generation Computer Systems, vol. 97, pp. 849-872, 2019.

[13] M. Li, Y. Zhao, R. Cao, J. Wang, and D. Wu, "A recursive framework for improving the performance of multi-objective differential evolution algorithms for gene selection," Swarm and Evolutionary Computation, vol. 87, p. 101546, Jun. 2024.

[14] N. Ahmad Zamri, N. A. Ab. Aziz, T. Bhuvaneswari, N. H. Abdul Aziz, and A. K. Ghazali, "Feature selection of microarray data using simulated kalman filter with mutation," Processes, vol. 11, no. 8, 2023.

[15] S. Abasabadi, H. Nematzadeh, H. Motameni, and E. Akbari, "Hybrid feature selection based on SLI and genetic algorithm for microarray datasets," The Journal of Supercomputing, vol. 78, no. 18, pp. 19725–19753, Dec. 2022.

[16] E. Pashaei and E. Pashaei, "Hybrid binary COOT algorithm with simulated annealing for feature selection in high-dimensional microarray data," Neural Computing and Applications, vol. 35, no. 1, pp. 353–374, Jan. 2023.

[17] A. Dabba, A. Tari, and S. Meftali, "A new multi-objective binary harris hawks optimization for gene selection in microarray data," Journal of Ambient Intelligence and Humanized Computing, vol. 14, no. 4, pp. 3157–3176, Apr. 2023.

[18] E. H. Houssein, H. N. Hassan, N. A. Samee, and M. M. Jamjoom, "A novel hybrid runge kutta optimizer with support vector machine on gene expression data for cancer classification," Diagnostics, vol. 13, no. 9, 2023.

[19] C. Pragadeesh, R. Jeyaraj, K. Siranjeevi, R. Abishek, and G. Jeyakumar, "Hybrid feature selection using micro genetic algorithm on microarray gene expression data," Journal of Intelligent & Fuzzy Systems, vol. 36, pp. 2241–2246, Mar. 2019.

[20] R. Kundu, S. Chattopadhyay, E. Cuevas and R. Sarkar, "AltWOA: Altruistic Whale Optimization Algorithm for feature selection on microarray datasets," Computers in Biology and Medicine, vol. 144, p. 105349, 2022.

[21] P. Luukka, "Feature selection using fuzzy entropy measures with similarity classifier," Expert Systems with Applications, vol. 38, no. 4, pp. 4600–4607, Apr. 2011.

[22] O. A. Alomari, S. N. Makhadmeh, M. A. Al-Betar, Z. A. A. Alyasseri, and R. A. Zitar, "Gene selection for microarray data classification based on grey wolf optimizer enhanced with TRIZ-inspired operators," Knowledge-Based Systems, vol. 223, p. 107034, Apr. 2021.

[23] M. Akhavan and S. M. H. Hasheminejad, "A two-phase gene selection method using anomaly detection and genetic algorithm for microarray data," Knowledge-Based Systems, vol. 262, p. 110249, Feb. 2023.

[24] M. Li, M. Lou, S. Deng and L. Wang, "TRF-WGHC—Top-Ranking filter and wrapper-based greedy hill-climbing gene selection for microarray-based cancer classification," Biomedical Signal Processing and Control, vol. 86, p. 105309, 2023.

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

[26] H. Yan, Q. Li, M.-L.Tseng, and X. Guan, "Joint-optimized feature selection and classifier hyperparameters by salp swarm algorithm in piano score difficulty measurement problem," Applied Soft Computing, vol. 144, p. 110464, Jun. 2023.

[27] W. Shi, J. Liu, J. Zhang, Y. Men, H. Chen, D. Wang and Y. Cao, "Feature selection and parameter optimization of support vector machines based on a local search based firefly algorithm for classification of formulas in traditional Chinese medicine," IEICE Transaction Fundamentals, vol. 105, no. 5, pp. 882–886, 2022.

[28] P. Wang, B. Xue, J. Liang, and M. Zhang, "Feature selection using diversity-based multi-objective binary differential evolution," Information Sciences, vol. 626, May 2023.

[29] M. Robnik-Sikonja and I. Kononenko, "Theoretical and empirical analysis of ReliefF and RReliefF," Machine Learning, vol. 53, pp. 23–69, Oct. 2003.

[30] A. Chaudhuri and T. P. Sahu, "Multi-objective feature selection based on quasi-oppositional based jaya algorithm for microarray data," Knowledge-Based Systems, vol. 236, p. 107804, Jan. 2022.

[31] W. Xie, Y. Fang, K. Yu, X. Min, and W. Li, "MFRAG: Multi-fitness RankAggreg genetic algorithm for biomarker selection from microarray data," Chemometrics and Intelligent Laboratory Systems, vol. 226, p. 104573, Jul. 2022.

[32] J. Kennedy and R. Eberhart, "Particle swarm optimization," in Proceedings of ICNN'95 - International Conference on Neural Networks, vol.4, pp. 1942–1948, Dec. 1995.

[33] X.-S. Yang, "Firefly algorithms for multimodal optimization," in Stochastic Algorithms: Foundations and Applications, O. Watanabe and T. Zeugmann, Eds., Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 169–178, 2009.

[34] H. Eskandar, A. Sadollah, A. Bahreininejad, and M. Hamdi, "Water cycle algorithm – a novel metaheuristic optimization method for solving constrained engineering optimization problems," Computers & Structures, vol. 110–111, pp. 151–166, Nov. 2012.

[35] S. Marjit, T. Bhattacharyya, B. Chatterjee, and R. Sarkar, "Simulated annealing aided genetic algorithm for gene selection from microarray data," Computers in Biology and Medicine, vol. 158, p. 106854, May 2023.

[36] S. Ahmed, K. K. Ghosh, S. Mirjalili, and R. Sarkar, "AIEOU: Automata-based improved equilibrium optimizer with U-shaped transfer function for feature selection," Knowledge-Based Systems, vol. 228, p. 107283, Sep. 2021.

[37] S. Ahmed, K. H. Sheikh, S. Mirjalili, and R. Sarkar, "Binary simulated normal distribution optimizer for feature selection: Theory and application in COVID-19 datasets," Expert Systems with Applications, vol. 200, p. 116834, Aug. 2022.