A Two-Stage Approach to Feature Selection in Image Classification Tasks

Shu-Chuan Chu[1], Xiaoping Cao[1], Zhi-Chao Dou[1], Jie Wu[2], Jeng-Shyang Pan[1,3,4]*

[1] College of Computer Science and Engineering, Shandong University of Science and Technology, China
[2] School of Electrical Engineering, Zhengzhou University of Light Industry, China
[3] School of Computer Science, Nanjing University of Information Science and Technology, China
[4] Department of Information Management, Chaoyang University of Technology, Taiwan
scchu0803@gmail.com, xiaoping120608@163.com, douzhichao2021@163.com,
wujie@zzuli.edu.cn, jengshyangpan@gmail.com

Abstract

Binary classification is pivotal in machine learning and widely applied across diverse domains. However, existing models are predominantly employed for multi-class classification problems. If the binary classification task is handled, it must be re-trained or migrated for training, which involves a significant time cost. To address this problem, this study employs a feature extraction strategy. It combines evolutionary computation for feature optimization by designing a two-module hybrid model for binary classification. Firstly, the features used for binary classification are extracted in VGG19. The next step employs a two-stage feature selection approach to identify the most relevant image features. The initial screening of features derived from the VGG19 model involved utilizing the ReliefF filtering method in the first feature selection stage. The Binary Great Wall Construction Algorithm (BGWCA) was employed in the second step to identify the most essential features further. The ultimate feature set is then used for image classification through a classifier known as the Support Vector Machine (SVM). The model achieved significant performance improvements on ten publicly available datasets. The algorithm outperformed other feature selection techniques, providing an innovative and efficient approach to addressing binary classification problems.

Keywords: Binary classification, Deep learning, Feature selection, BGWCA, VGG19

1 Introduction

Binary classification is essential in machine learning and has garnered considerable scholarly interest. Over the past few decades, scholars have dedicated themselves to developing practical algorithms and methods to address binary classification challenges in various domains.

The development of binary classification has progressed through several stages. Initially, linear models like logistic regression were commonly used [1-2]. As data complexity increased, Support Vector Machines (SVM) were introduced to handle non-linear relationships using kernel functions [3-4]. Ensemble learning methods, such as Random Forests and Gradient Boosting Trees, then gained traction for their model combination capabilities [5-7]. The rise of deep learning, especially Convolutional Neural Networks (CNN), enabled advanced feature extraction from raw data [8]. Transfer and self-supervised learning have recently been applied to address data insufficiency [9-10]. Additionally, multimodal learning focuses on integrating multiple data types to improve classification performance. However, these approaches often need help with challenges like high computational costs and extended training times [11]. A more efficient alternative is to use pre-trained models directly for multi-class tasks. Having undergone extensive training, these models capture rich feature representations and provide strong classification abilities. However, their complexity may be excessive for new binary classification tasks. To optimize their use, it is beneficial to extract features from these models rather than retraining the entire network.

This study aims to design a classification model that can distinguish between two types of images when provided with classified photos. The VGG19 model extracts features from the classification images, resulting in a large-dimensional feature vector (4096 feature attributes). Hence, this study introduces a two-stage choice features approach to identify the most optimal subset. In the first stage, the ReliefF filtering method ranks the feature attributes, selecting the top n tagged features. During the second stage, the Binary Great Wall Construction Algorithm (BGWCA), a metaheuristic algorithm, performs feature selection. This algorithm aims to optimize the feature set that was generated in the initial stage. The optimum feature subset obtained is subsequently fed into a Support Vector Machine (SVM) classifier for prediction.

This manuscript presents the primary contributions as follows:

1. Combining deep learning and metaheuristic wrapper feature selection strategies aims to create a highly effective hybrid model for binary classification tasks.

2. Utilizing the VGG19 architecture for feature extraction, a two-stage feature choice technique is

*Corresponding Author: Shu-Chuan Chu; Email: scchu0803@gmail.com
DOI: https://doi.org/10.70003/160792642026112706001

employed to minimize computing time during the model's training. This is achieved by deleting redundant and non-informative features.

3. Transforming the Great Wall Construction Algorithm into the Binary Great Wall Construction Algorithm.

4. Evaluate the efficacy of the suggested model using ten publicly available datasets.

In the subsequent phase of our suggested two-stage choosing features procedure, the Binary Great Wall Construction Algorithm (BGWCA) is used to predict binary classification problems efficiently. Binary algorithms can handle data with discrete characteristics and high computational efficiency and optimization. Therefore, they are used to solve discrete problems [12]. In this manuscript, BGWCA is enhanced using the Sigmoid function. The improved algorithm has undergone evaluation and comparison with other feature selection techniques, such as the Binary Grey Wolf Optimization Algorithm (BGWO), Binary Bat Algorithm (BBA), and Binary Particle Swarm Optimization (BPSO), after conducting tests on ten datasets. The Binary Great Wall Construction Algorithm (BGWCA) performance has been demonstrated to be superior. Therefore, this algorithm is chosen for application in the proposed work.

The current study is organized as follows: Section 2 analyzes the hybrid model, detailing using the VGG19 model for feature extraction and implementing a two-stage feature selection procedure. Section 3 presents the experimental results and compares binary optimization methods for binary classification tasks. Section 4 concludes the study and suggests future research directions.

2 Proposed Bi-stage Hybrid Mode

In this study, ten binary-class image datasets were used for testing. The process of the proposed model is shown in Figure 1. VGG19 served as the feature extractor in the early stage, producing 4096 feature attributes. However, the high dimensionality of this feature set introduced challenges, such as increased computational complexity [13]. A two-stage feature selection strategy was employed to address these issues and identify the most relevant features. The ReliefF filter ranked and selected the top 300 features in the first stage. These selected features were refined using BGWCA, resulting in the final optimal feature subset. This refined subset was then used for binary classification predictions in the subsequent phase.

The proposed methodology entails utilizing VGG19 for feature extraction and employing a two-stage FS technique to determine the most optimized feature set (as depicted in Figure 1).

2.1 Feature Extraction Using VGG19

The VGG19 architecture is a convolutional neural network with approximately 19 layers, including convolutional, pooling, and fully connected layers. It produces 1000-dimensional features in the final layer for 1000-class classification on the ImageNet dataset [14]. However, applying VGG19 to binary classification directly through migration or retraining can be time- and resource-intensive. Migration training retains the model's knowledge but requires significant tuning. Retraining involves optimizing numerous parameters, which is even more time-consuming. Therefore, this study employs the pre-trained VGG19 model for feature extraction, allowing for quick retrieval of high-dimensional image features (4096 dimensions) optimized through a two-stage feature selection technique. The SVM classifier is then used to predict the class of the input image.

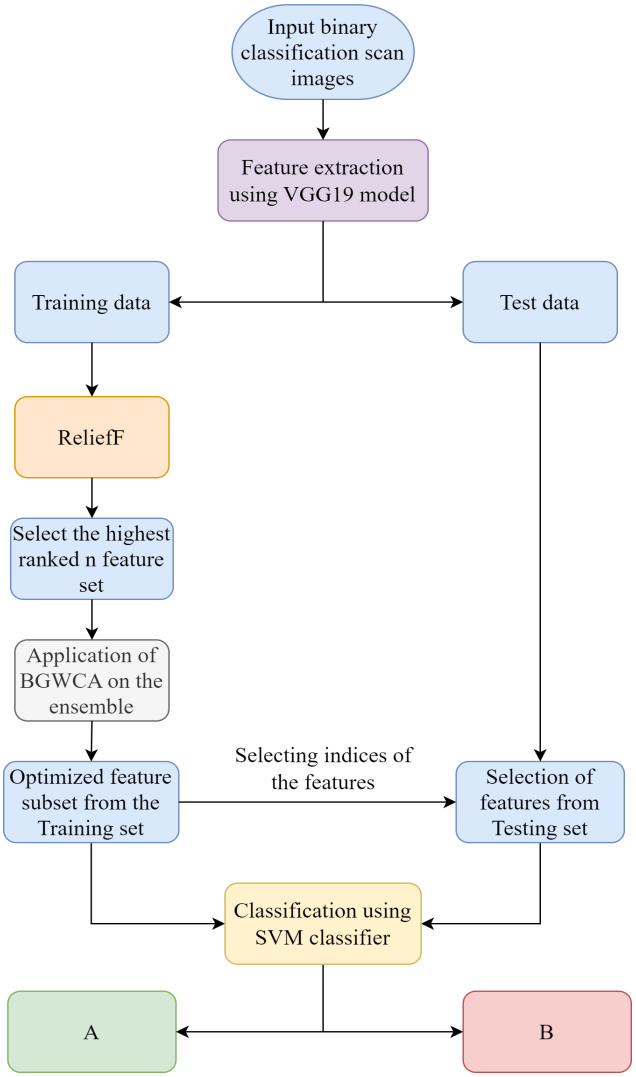

Figure 1. Diagram illustrating the suggested model

The VGG19 model considers an image of size 224×224 in the input layer, the convolutional layer uses a 1×1 step size, and the maximum pooling layer uses a 2×2 step size. Table 1 shows the architecture of the modified VGG19 model. As seen from the table, the convolutional layer of VGG19 was kept, and the last fully connected layer of the third part of the classifier of the original model of VGG19 was removed. The new classifier replaces the original classifier of VGG19, making the output dimension 4096. From the last fully connected layer, each image sample generates 4096 features.

Table 1. Detailed information on the architecture of VGG19 used for feature extraction from binary classification images

Layer type	Filter size	Number of filters	Stride
Input	224×224×3	-	-
Conv1	3×3	64	1
Conv2	3×3	64	1
Maxpool1	2×2	-	2
Conv3	3×3	128	1
Conv4	3×3	128	1
Maxpool2	2×2	-	2
Conv5	3×3	256	1
Conv6	3×3	256	1
Conv7	3×3	256	1
Conv8	3×3	256	1
Maxpool3	2×2	-	2
Conv9	3×3	512	1
Conv10	3×3	512	1
Conv11	3×3	512	1
Conv12	3×3	512	1
Maxpool4	2×2	-	2
Conv13	3×3	512	1
Conv14	3×3	512	1
Conv15	3×3	512	1
Conv16	3×3	512	1
Maxpool5	2×2	-	2
Flatten	-	-	-
FC	-	4096	-
Output	-	4096	-

2.2 Selection of Features

During feature extraction, redundant and irrelevant characteristics may be included, affecting classification accuracy and increasing computational cost [15]. To optimize the feature set, feature selection (FS) techniques are often applied [16-18]. FS aims to identify a subset of features that performs best within a specific classification framework [19-21]. FS algorithms are generally classified into filters, wrappers, and embedded methods [22]. Filters use statistical or probabilistic measures to rank features, selecting or removing them before classification. Wrappers use machine learning algorithms to validate feature impact and choose the best subset, while embedded methods integrate FS within machine learning models. This study employs a two-stage methodology to select the most relevant features from the VGG19-generated set, enabling the design of a binary classification system. The following subsections will detail the methods employed.

2.2.1 ReliefF

This study proposes an effective feature selection (FS) method employing a two-stage strategy to reduce the feature space obtained from scanned images. The initial phase involves the application of a filter-based feature selection strategy to get a more refined feature set by eliminating statistically weak characteristics [23]. Subsequently, this feature set is introduced into the wrapper-based FS module. A widely used filter method, the ReliefF algorithm, was chosen in the first stage.

The ReliefF algorithm has proved very useful and successful in attribute selection tasks [24-27]. Its main goal is to determine the importance of features in a categorization task by evaluating the correlation between them. The method is based on nearest-neighbor relationships and scores features by comparing feature differences between instances of the same and different categories [28]. These scores reflect the contribution of features to distinguish cases from various categories. ReliefF considers the weights between instances to ensure that instances from other density regions are effectively considered. Ultimately, the feature scores generated by ReliefF can be used to rank and identify the most essential features for the classification task. This can help improve classification performance and reduce dimensionality, especially in high-dimensional datasets. The equation is shown in Eq. (1) and Eq. (2).

$$S = \sum_{a \in Class(a)} \frac{\frac{p(a)}{1 - p(Class(a))} \sum_{j=1}^{k} diff(a, Mj(a))}{mk} \quad (1)$$

$$W(f) = W(f) - \sum_{j=1}^{k} \frac{diff(a, Hj)}{mk} + S \quad (2)$$

The function $diff(\cdot)$ computes the distance between two samples concerning the feature f, while $p(a)$ denotes the probability associated with a class. $W(f)$ represents the weight of feature f. a represents the categories in the dataset. $M_j(a)$ represents the neighboring set of sample j under category a. H_j represents the feature vector of a sample j. The variable k denotes the quantity of adjacent samples. The variable m denotes the amount of samples present in the dataset.

The ReliefF technique ranks attributes in the VGG19-generated feature set and selects the top 300 features. This reduces the high-dimensional feature space, reaching up to 4096 dimensions and containing redundant or irrelevant information, leading to increased computational complexity for BGWCA. ReliefF helps filter out irrelevant features, retaining only the most important ones, thereby reducing the number of features and simplifying the optimization process. This refined feature set allows BGWCA to operate in a smaller search space, improving search efficiency and accuracy. Additionally, ReliefF ensures feature stability across different datasets, enhancing BGWCA's ability to optimize classification performance. Thus, ReliefF boosts overall algorithm efficiency and provides a strong foundation for BGWCA, improving the model's robustness and practical effectiveness.

2.2.2 Binary Great Wall Construction Algorithm

The Great Wall Construction Algorithm is inspired by the competition and elimination behaviors exhibited by the workers involved in the construction of the Great Wall. This study proposes modifying the Great Wall

construction algorithm using a binary approach. A more common conversion function is used, namely the Sigmoid function, which maps values from negative infinity to positive infinity to intervals from 0 to 1. The introduction of the Sigmoid function facilitates the conversion of the algorithm's continuous outputs to binary decisions and improves the speed of convergence by providing a smooth mapping. This combination allows the modified algorithm to efficiently handle the feature selection task and accurately identify relevant features while maintaining strong search capabilities.

The Sigmoid function is represented by the equation and its corresponding graphic (as depicted in Figure 2).

$$s(x) = \frac{1}{1+e^{-x}} \tag{3}$$

Included among these, x is the input value, which is calculated by the Sigmoid function to produce an output between 0 and 1.

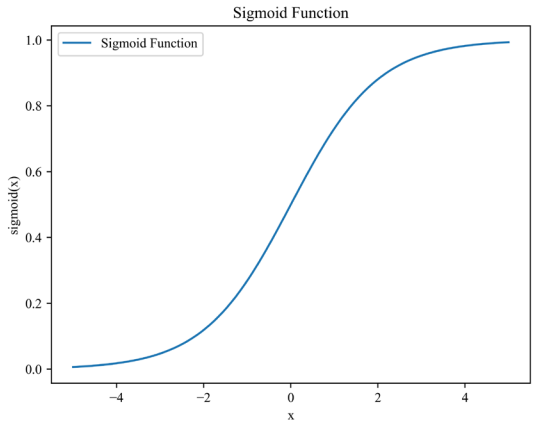

Figure 2. Sigmoid function graph

Workers were classified as engineers, laborers, and soldiers during the construction of the Great Wall. Worker speed was affected by the ruggedness and slope of the terrain. Tasks were assigned before construction began, with engineers using tools like wheelbarrows to improve productivity. Soldiers supervised the workers while laborers transported heavy items such as stones. The project manager eliminated inefficient workers after assessing progress. Subsequent sections will describe the mathematical models for the movements of engineers, laborers, and soldiers and the worker elimination process [29]. Engineers, laborers, and soldiers are represented by the symbols E_i, L_i, and S_i, respectively.

To enhance the readability of the algorithm, the following symbol table has been merged (see Table 2), which includes all relevant variables and their definitions. The merged symbol table simplifies the algorithm's description, making the symbols' meanings and purposes easier to understand.

Table 2. Notations and description

Notations	Description
LB	Lower bounds on the problem space
UB	Upper bounds on the problem space
t	Current number of iterations
MaxIter	Max number of iterations
R	Engineer's promotion radius
R1	Random number located in the interval [0,1]
R2	Random weights of the engineer
v	Speed of engineers
gampdf	Probability density function of the gamma distribution
m	Mass of the rock
g	Gravitational acceleration
T	Thrust generated by the tool
TL	Engineer's tool wear level
H(t)	Height of worker
C(t)	Fractal dimension
P	Parameter of gamma probability distribution function
Q	Parameter of gamma probability distribution function
E_{best}	Position of the engineer
S_{best}	Position of the solider
L_{best}	Position of the labor
N	Number of populations
p	Elimination rate of workers
C_{min}	constant
C_{max}	constant
EL	Number of eliminated workers
PN	Position of a newly recruited workers
PR	Position of workers retained after elimination

Initialization:

The initial stage in the binary method involves initializing a two-dimensional matrix that displays the position of the search agent (the worker's position). This matrix is randomly populated with 1's and 0's. The worker's position is modified according to Eq. (4) and Eq. (5).

$$\overrightarrow{Po_x} = Sigmoid\left(\overrightarrow{Po_x}\right) = \frac{1}{1+e^{-10(Po_x-0.5)}} \tag{4}$$

$$Po_x = \begin{cases} 1, & if\ rand < Sigmoid(Po_x) \\ 0, & otherwise \end{cases} \tag{5}$$

Mathematical model of the engineer movement (exploitation):

An Emperor's representative will select laborers to become engineers. These engineers will use their skills and tools to transport stones efficiently and maximize productivity. They will compete with the most efficient

workers to identify the best candidates. The pertinent equations are as follows:

$$E_i(t+1) = E_{best}(t) + R + E_i(t)v_i(t) * R_2 \quad (6)$$

$$R = (-1)^k (X_{best}(t) - E_i(t))R_1 \quad (7)$$

Where k denotes an integer ranging from 0 to 1, the formula for updating the velocity of the engineer at the i-th position in the t-th iteration is represented as $v_i(t)$. It is formally defined by Eq. (8). In the iteration denoted as t, the symbol $E_{best}(t)$ represents the most efficient worker, indicating the ideal candidate solution. The position of the engineer at the i-th position in the $(t+1)$-th iteration is defined by $E_i(t+1)$. The variable $E_i(t)$ represents the engineer's position at the i-th spot in the t-th iteration. The random number R_1 is distributed equally within the interval [0, 1]. R_1 and R_2 provide random weights for the engineer. The promotion radius, denoted as R, is calculated based on the difference in efficiency between the i-th worker and the leading engineer. The outcome of R is ascertained by multiplying its magnitude by a stochastic integer within the interval [-1, 1], where a positive integer signifies proximity to the agent's position and a negative number signifies distance from it.

$$v_i(t) = \left(\frac{T * TL}{m} - g\frac{H(t)}{\sin\theta_i} \right) C(t) * gampdf(t|P,Q) \quad (8)$$

In this context, m represents the mass of the rock, and g indicates the acceleration due to gravity. θ_i is a randomly generated angle ranging from 0 to 80 degrees, representing the slope at the worker i's position relative to the horizontal. T denotes the thrust produced by the tool, while *gampdf* refers to the probability density function of the gamma distribution, as shown in Eq. (11) [30]. In this function, P and Q are parameters, and TL indicates the wear on the engineer's tool, decreasing linearly from 1 to 0. The fractal dimension $C(t)$ reflects the complexity of the mountain path's topography in the t-th iteration. Additionally, $H(t)$ represents the vertical distance between the worker and the mountain top at position i in the t-th iteration. The mathematical formulations for $H(t)$ and $C(t)$ are provided below:

$$H(t) = 1 - \frac{t}{MaxIter} \quad (9)$$

$$C(t) = \log\left((C_{\max} - C_{\min}) \frac{MaxIter - t}{MaxIter} + C_{\min} \right) \quad (10)$$

$$gampdf(t|P,Q) = \frac{Q^{-P} * t^{P-1} * \exp(-t/Q)}{\Gamma(P)} \quad (11)$$

In Eq. (9) and Eq. (10), t represents the current iteration number. The variable *MaxIter* denotes the upper limit of iterations. The variables C_{max} and C_{min} are fixed values.

Mathematical model of soldier movement (exploration):

The velocity formula for the soldiers is given by Eq. (12), and Eq. (13) outlines their displacement from previous positions. Soldiers move toward the nearest workers to supervise their tasks, guided by the positions of E_{best}, $S_{best,}$ and L_{best}. They scatter while searching for a leader and regroup when changing leaders. Random weights R_1, R_2, R_3, and R_4 are assigned to enhance exploration and reduce the risk of getting trapped in local optima. The sign function helps determine the soldiers' following positions, facilitating the algorithm's global search and exploration. When $sign(f(ide) - f(i))= 1$, the soldiers move closer to the leader, and when $sign(f(ide) - f(i))= -1$, the soldiers move away from the leader.

$$v_i(t) = mg\frac{H(t)}{\sin_i} C(t) * gampdf\left(\frac{t}{MaxIter} | P,Q \right) \quad (12)$$

$$\begin{aligned} S_i(t+1) = {} & S_i(t) + (S_{best}(t) - S_i(t)) * R_3 \\ & + sign(f(ide) - f(i)) \times (Po_{ide} - S_i(t))v_i(t) * R_4 \end{aligned} \quad (13)$$

The symbol *ide* represents the soldier closest to the worker at position i. The variable S_{best} denotes the soldier who has achieved the highest level of performance. The variable $S_i(t+1)$ indicates the location of the laborer at the *i-th* position in the $(t+1)$-*th* iteration, whereas $S_i(t)$ represents their position at the *t-th* position in the iteration. The variables R_3 and R_4 are created randomly within the interval [0, 1] uniformly.

Mathematical model of labor movement:

All workers, excluding engineers and soldiers, are categorized as laborers by the agent. Eq. (14) illustrates the mathematical model for laborers in this process.

$$\begin{aligned} L_i(t+1) = {} & L_i(t) + 2(L_{best}(t) - S_i)R_5 \\ & + (PB_i(t) - S_i) * gampdf\left(\frac{t}{MaxIter} | P,Q \right) \end{aligned} \quad (14)$$

Let $L_i(t+1)$ represent the position of the laborer at position i in the iteration at position $(t+1)$, and $L_i(t)$ represent their position at position t in the iteration at position t. The variable PB_i denotes the personal best, while the variable L_{best} symbolizes the laborer who performs the best.

Mathematical model for replacing and recombination:

Workers were proportionally terminated during the construction of the Great Wall owing to potential loss of

labor capacity or insufficient efficiency. The BGWCA algorithm assigns a value of 10% to the variable p. The Emperor's agent recruits new workers to replace those who are eliminated. The evaluation function utilizes the worker positions derived from Eq. (6), Eq. (13), and Eq. (14) for computational purposes. Subsequently, workers with high fitness are eliminated to ensure the construction project's progress is completed on time.

$$EL = ceil(N * p) \tag{15}$$

$$\begin{cases} Po^{j}_{i,new} = (UB_j - LB_j)R_6 + LB_j, \\ i = 1, 2, \ldots, El \\ j = 1, 2, \ldots, d \end{cases} \tag{16}$$

$$PN = [PR, PE] \tag{17}$$

The ceil function in the context yields the minimum integer bigger than or equal to the given phrase. Meanwhile, R_6 represents a randomly generated number that is uniformly distributed across the interval [0, 1]. PR represents the retained work positions after elimination. PN represents the positions of newly recruited workers, and EL represents the number of workers eliminated.

It is important to note that, in this context, the fitness value of a solution vector (in our case, a feature vector) represents both the classification error rate produced by the Support Vector Machine (SVM) classifier and the number of selected features. This approach evaluates the effectiveness of the feature selection technique. Algorithm 1 outlines the critical steps of the BGWCA process.

Algorithm 1. Pseudocode of the steps used in BGWCA

Input: Condensed set of features acquired from guided-FS

Output: Optimal feature set with its corresponding fitness

Randomly initialize the workers to create population.

Calculate the length of the input features and set the upper and lower bounds (*UB* and *LB*).

Initialize the position of the worker

Determine the best workers

for i = 1 **to** *MaxIter* **do**

  Calculate the fitness values of the position vectors of workers.

  update the position of engineers (Eq. **(6)**).

    update the position of Soldiers (Eq. **(13)**).

    update the position of Labors (Eq. **(14)**).

    Determine the new position of best workers.

    Remove individuals with a lower fitness score.

end

The lowest fitness value and the number of features in the optimal solution are to be returned.

BGWCA initially generates a collection of random solutions (precisely ten) in which the placements of the workers are randomly assigned inside the upper and lower boundaries. The algorithm then proceeds to the next step by updating the positions of three types of workers to determine the best feature vector. The highest-ranking worker (i.e., the engineer) represents the selected best feature vector. The condition is configured to the maximum number of iterations. The algorithm determines the best feature subset for classification purposes upon completion.

2.3 Computational Complexity

This subsection examines the computational complexity of the proposed BGWCA, which is influenced by three factors:

1. Population initialization: The complexity of initializing the population is $O(N \times D)$, where N represents the population size, and D is the number of parameters (dimensions) of the problem.

2. Fitness evaluation: Evaluating the surrogate fitness requires $O(MaxIter \times N \times D)$, where MaxIter refers to the number of iterations.

3. Solution update: The complexity of updating the solution is $O(El)$, where El refers to worker elimination and recruitment processes.

Therefore, the computational complexity of the proposed BGWCA is $O(MaxIter \times N \times D \times El)$.

2.4 The SVM Classifier

The Support Vector Machine (SVM) is a supervised learning model in machine learning that was first introduced by Vapnik in 1995 [31]. It is generally used to solve binary classification issues. The main objective of this study was to identify the most suitable hyperplane $f(w, x) = w \cdot x + b$ to classify a dataset with characteristics x within the range of R^m into two distinct classes. The Support Vector Machine (SVM) training process constructs a model based on training examples, effectively classifying samples into two classes.

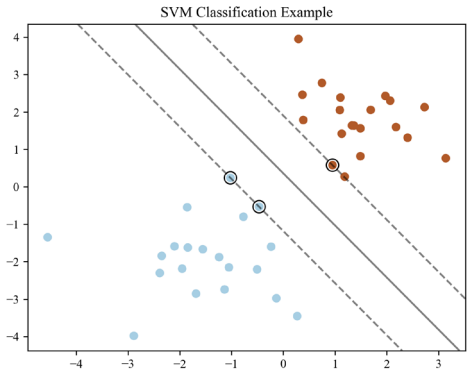

Figure 3. The working principle of SVM

Figure 3 illustrates the distribution of data points, with different colors representing their categories and highlighting the features of the data. It also shows

the decision boundary of the model, which marks the classification boundary in feature space, visualized with contour lines. The colored regions indicate predictions for different categories. Support vectors, crucial data points closest to the decision boundary, are highlighted as they influence its position and orientation. This visual representation enhances the understanding of the Support Vector Machine (SVM) algorithm's operation and demonstrates the classification process effectively.

3 Experimental Results and Discussion

Based on the previous discussion, ten binary image datasets were employed. The VGG19 model was utilized as a feature extractor, generating a 4096-dimensional feature vector in the current work. This vector underwent initial feature selection using the Relief method and then final feature selection through BGWCA. In order to evaluate the efficacy of the proposed approach, a comparative analysis was conducted with ten feature selection algorithms, including Binary Particle Swarm Optimization (BPSO) [32-33], Binary Differential Evolution (BDE) [34-35], and Binary Genetic Algorithm (BGA) [36], Binary Sparrow Search Algorithm (BSSA) [37], Binary Grey Wolf Optimization Algorithm (BGWO) [38], Binary Sine Cosine Algorithm (BSCA) [39], Binary Whale Optimization Algorithm (BWOA) [40], Binary Jaya Algorithm (BJaya) [41], Binary Flower Pollination Algorithm (BFPA) [42] and Binary Bat Algorithm (BBA) [43].

In addition, we not only compare our method with ten binary algorithms. Several state-of-the-art classification methods are also evaluated, including the Marine Predator Chaos Algorithm (MPCA) [44], Spider Monkey Optimization followed by Cuckoo Search (CSSMO) [45-46], and Harris Hawk Optimisation with Cuckoo Search Algorithm (HHOCSA) [47]. This comprehensive comparison aims to highlight the strengths and effectiveness of our methods and provide strong support for the advancement of current classification techniques.

All experiments were conducted under the same setup and conditions, and a Support Vector Machine (SVM) classifier was used in image prediction. The experiments were conducted twenty times independently to account for randomness, and the results were averaged. The number of features in each dataset is consistent with the problem's dimensionality. The evaluation criteria include accuracy, fitness value, and standard deviation. Table 3 provides information about the algorithms and the parameters they use. The formula for calculating the fitness function in this manuscript is as follows:

$$f(x) = \alpha * error + \beta * \frac{Fs}{Fa} \tag{18}$$

In the present study, the variable *error* denotes the classification error rate attained by the Support Vector Machine (SVM) classifier. Additionally, α and β are parameters that determine the weights of features and classification, respectively. The value of α ranges from 0 to 1, and β can be estimated as (1-α). *Fs* denotes the number of selected features, while *Fa* indicates the size of the features filtered by the Relief filter in the dataset. Since classifier performance is considered the most crucial metric in this research, α is 0.99, and β is selected as 0.01.

Table 3. Parameter settings of tested algorithms

Parameter	Setting
Population size	10
Number of iterations	100
Number of runs	20
BPSO	c1=2, c2=2, w=0.9
BGA	Crossover rate=0.8, mutation rate =0.01
BDE	Crossover rate=0.9, F=0.5
BGWO	A=2
BBA	Max.frequency=2, min.frequency =0, pulse emission rate=0.5, Loudness=0.5
BSSA	C1=2
BSCA	A=2
BWOA	B=1, random numbers r1,r2 And p in [0,1]
BFPA	β=1.5, P=0.8
MPCA	P=0.5
CSSMO	α=1, Pa_{min}=0.3, Pa_{max}=0.5
HHOCSA	α=1, p=0.25

3.1 A Comparative Examination and Evaluation of Fitness Values

Figure 4, Table 4 and Table 5 present the average fitness values and standard deviations (STD) for each algorithm, including the proposed BGWCA. BGWCA generally demonstrates outstanding performance, surpassing the best average fitness and outperforming other binary and three advanced algorithms in many datasets, validating its effectiveness. Thus, BGWCA effectively identifies optimal feature subsets. STD measures the stability and variability of algorithms in high-dimensional data. Table 5 shows the STD results compared to other binary algorithms. Table 4 displays the STD results of BGWCA against three advanced algorithms. While some algorithms perform better on specific datasets, BGWCA still ranks the highest overall, indicating its stability and effectiveness.

3.2 A Comparative Examination and Evaluation of Accuracy

Table 6 and Table 7 present the standard deviation (STD) of classification accuracy for the latest metaheuristic algorithms. BGWCA performs excellently in STD compared to three advanced algorithms across most datasets. Although not the top performer in some instances, BGWCA ranks highest overall, indicating excellent

reliability and consistency. Table 6 further confirms BGWCA's superiority over other binary algorithms.

Boxplots are used to compare accuracy values among different algorithms. These plots visually represent a dataset's distribution, including central tendency and variability (Frigge et al., 1989). Figure 5 Boxplots compare the accuracy of proposed and other algorithms across datasets and display the classification accuracies of various algorithms against BGWCA. The boxplots for datasets D1–D10 reveal that BGWCA has a higher median accuracy, and its mean accuracy is more significant than all other algorithms. Overall, BGWCA maintains a higher level of classification accuracy.

3.3 A Comparative Examination and Evaluation of Convergence Curves

Figure 6 illustrates the convergence curves of the Binary Great Wall Construction Algorithm (BGWCA) compared to other algorithms on ten datasets. In feature selection problems, convergence speed is a crucial measure of the algorithm›s efficiency. BGWCA, shown by the red curves in the figure, exhibits a faster descending trend on most of the datasets, which is significantly better than the other algorithms, demonstrating its excellent convergence performance. By analyzing these curves, it can be concluded that BGWCA is efficient and stable in the feature selection process and able to find the optimal feature subset quickly, further proving its effectiveness and superiority in practical applications.

3.4 A Comparative Examination and Evaluation of the Best and Worst Fitness Values

Figure 7 and Figure 8 compare the proposed algorithm with others, presenting the best and worst fitness values from twenty experiments across various datasets. The results indicate that the proposed algorithm consistently outperforms other algorithms, including binary ones and three state-of-the-art methods, regarding fitness values, demonstrating superior effectiveness and adaptability. Notably, it shows more excellent stability in worst fitness values, which are significantly lower than those of other algorithms, indicating its ability to minimize the impact of extreme situations across different datasets. Additionally, the algorithm's robustness during optimization enhances its applicability in various real-world scenarios.

Overall, these results affirm the effectiveness and reliability of the proposed algorithm for solving optimization problems and provide a solid theoretical basis for its application in related fields.

3.5 Comparison with the VGG19 Model

In the experiments, comparisons were made with the VGG19 model, and the experimental results are shown in Table 8. In contrast, the model in this paper does not require pre-training, which significantly reduces the training time and saves a lot of resources. Although the VGG19 model has certain advantages in terms of accuracy, in practical applications, considering the resource consumption and training efficiency, the model proposed in this paper shows higher practicality in many cases.

Table 4. Compare the suggested method with different methods in terms of the standard deviation (STD) of fitness values

Datasets	Meature	MPCA	CSSMO	HHOCSA	BGWCA
D1	Std	**0.003046**	0.004564	0.003627	0.003247
	Rank	1	4	3	2
D2	Std	0.025796	0.012544	0.016258	**0.011121**
	Rank	4	2	3	1
D3	Std	0.014366	0.013344	0.015543	**0.010231**
	Rank	3	2	4	1
D4	Std	0.009114	0.007405	**0.006659**	0.008057
	Rank	4	2	1	3
D5	Std	0.002306	0.003510	0.002433	**0.002236**
	Rank	2	4	3	1
D6	Std	0.007759	0.008396	0.006342	**0.006082**
	Rank	3	4	2	1
D7	Std	**0.003432**	0.004099	0.004870	0.003683
	Rank	1	3	4	2
D8	Std	0.014550	0.019488	**0.009165**	0.012596
	Rank	3	4	**1**	2
D9	Std	0.009104	0.008840	0.011231	**0.008443**
	Rank	3	2	4	1
D10	Std	0.013653	0.014841	0.013198	**0.012424**
	Rank	3	4	2	1
	Mean Rank	2.7	3.1	2.7	1.5
	Overall Rank	3	4	2	1

Table 5. Compare the suggested method with different metaheuristic algorithms in terms of the standard deviation (STD) of fitness values

Datasets	Meature	BPSO	BGA	BDE	BGWO	BBA	BSSA	BSCA	BWOA	BFPA	BJaya	BGWCA
D1	Mean	0.003897	**0.003246**	0.003486	0.003556	0.003806	0.003971	0.005405	0.003941	0.003391	0.004196	0.003247
	Rank	7	1	4	5	6	9	11	8	3	10	2
D2	Mean	0.14927	0.013723	0.018439	0.013546	0.015093	0.017523	0.015818	0.014757	0.013555	0 .015319	**0.011121**
	Rank	6	4	11	2	7	10	9	5	3	8	1
D3	Mean	**0.010059**	0.0112164	0.011867	0.010204	0.013916	0.014535	0.013596	0.012585	0 .012226	0.013768	0.010231
	Rank	1	4	5	2	9	11	8	7	6	10	3
D4	Mean	0.008415	**0.007793**	0.011474	0.010116	0.011902	0.011086	0.014704	0.011652	0.008486	0 .009711	0.008057
	Rank	3	1	8	6	10	7	11	9	4	5	2
D5	Mean	0.004303	0.002601	0.002556	0.002712	0.003204	0.006132	0.005383	0.003211	0.004656	0.003808	**0.002236**
	Rank	8	3	2	4	5	11	10	6	9	7	1
D6	Mean	0.007824	0.008736	0.011537	0.006206	0.010129	0.007601	0.011681	0.009419	0.011631	0.008066	**0.006082**
	Rank	4	6	9	2	8	3	11	7	10	5	1
D7	Mean	0.003743	0.003926	0.003663	**0.002464**	0.008806	0.004173	0.009085	0.009026	0.008798	0.004078	0.003683
	Rank	4	5	2	1	9	7	11	10	8	6	3
D8	Mean	0.013209	0.012602	0.020175	0.019804	0.016238	0.015154	0.024459	0.024304	0.016211	0.020002	**0.012596**
	Rank	3	2	8	7	5	4	11	10	6	9	1
D9	Mean	0.014901	0.011855	0.014903	**0.005377**	0.022185	0.010977	0.016939	0.017817	0.011016	0.011269	0.008443
	Rank	8	6	7	1	11	3	9	10	4	5	2
D10	Mean	**0.008827**	0.016254	0.014738	0.011894	0.013231	0.019943	0.017601	0.018018	0.013128	0.023867	0.012424
	Rank	1	7	6	3	5	10	8	9	4	11	2
	Mean Rank	4.5	3.9	6.2	3.3	7.5	7.5	9.9	8.1	5.7	7.6	1.8
	Overall Rank	4	3	6	2	7	8	11	10	5	9	1

Table 6. Compare the suggested method with various metaheuristic algorithms in terms of standard deviation (STD) accuracy

Datasets	Meature	BPSO	BGA	BDE	BGWO	BBA	BSSA	BSCA	BWOA	BFPA	BJaya	BGWCA
D1	Mean	0.005043	0.005314	**0.004994**	0.005337	0.007333	0.00534	0.006492	0.006304	0.005685	0.005848	0.005076
	Rank	2	4	1	5	11	6	10	9	7	8	3
D2	Mean	0.024334	0.023014	0.025341	0.023048	0.024076	0.028804	0.024843	0.024494	0.032475	0 .032858	0.022741
	Rank	5	2	8	3	4	9	7	6	10	11	1
D3	Mean	0.018925	0.014996	0.018139	**0.011899**	0.021306	0.026135	0.041019	0.01811	0 .022356	0.017196	0.01369
	Rank	7	3	6	1	8	10	11	5	9	4	2
D4	Mean	0.011908	**0.006876**	0.010678	0.010607	0.016558	0.013989	0.0147388	0.010211	0.013273	0 .008568	0.010832
	Rank	7	1	4	3	11	9	10	2	8	6	5
D5	Mean	0.005008	0.005811	0.005352	0.006127	0.005694	0.007071	0.007041	0.006576	0.005772	0.006981	**0.004138**
	Rank	2	6	3	7	4	11	10	8	5	9	1
D6	Mean	0.016824	0.012741	0.013096	0.014005	0.014944	0.017002	0.016617	0.013875	0.013843	0.013358	**0.011984**
	Rank	10	2	3	7	8	11	9	6	5	4	1
D7	Mean	0.005365	0.005402	0.007272	0.006899	0.012883	0.007811	0.008037	0.007141	0.007088	0.005719	**0.003924**
	Rank	2	3	8	5	11	9	10	7	6	4	1
D8	Mean	0.016396	0.016268	0.025452	0.01422	0.017755	0.023073	0.024487	0.023607	0.024347	0.027364	**0.012649**
	Rank	4	3	10	2	5	6	9	7	8	11	1
D9	Mean	0.022513	0.020174	0.020879	0.021916	0.039154	0.023073	0.023601	0.023063	0.028767	0.026085	**0.018634**
	Rank	5	2	3	4	11	7	8	6	10	9	1
D10	Mean	0.037138	**0.029495**	0.038764	0.03342	0.044704	0.033564	0.046708	0.037761	0.038188	0.042218	0.032155
	Rank	5	1	8	3	10	4	11	6	7	9	2
	Mean Rank	4.9	2.7	5.4	4	8.3	8.2	9.5	6.2	7.5	7.5	1.8
	Overall Rank	4	2	5	3	10	9	11	6	7	8	1

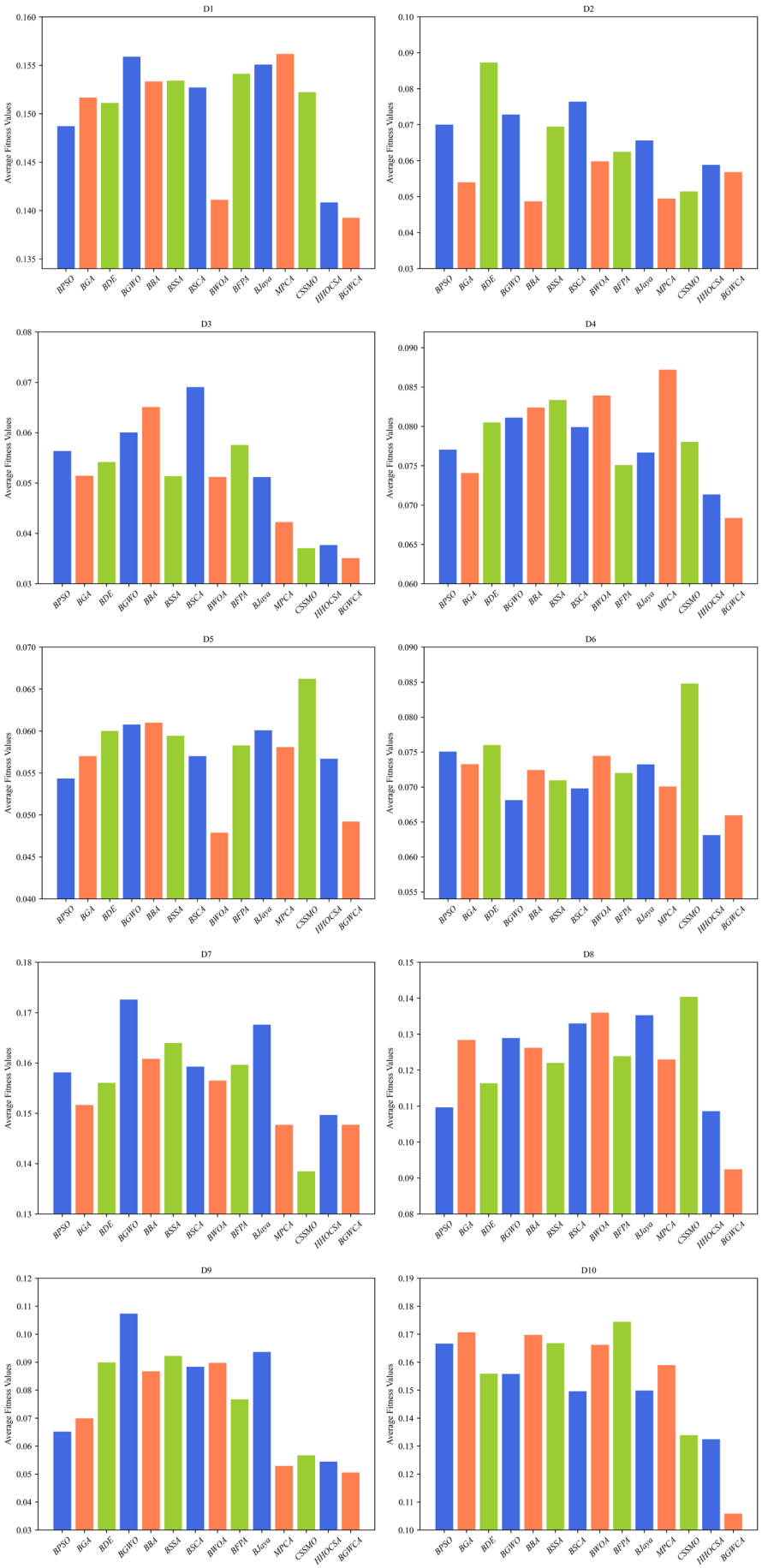

Figure 4. Bar chart depicting average fitness values across algorithms for all datasets

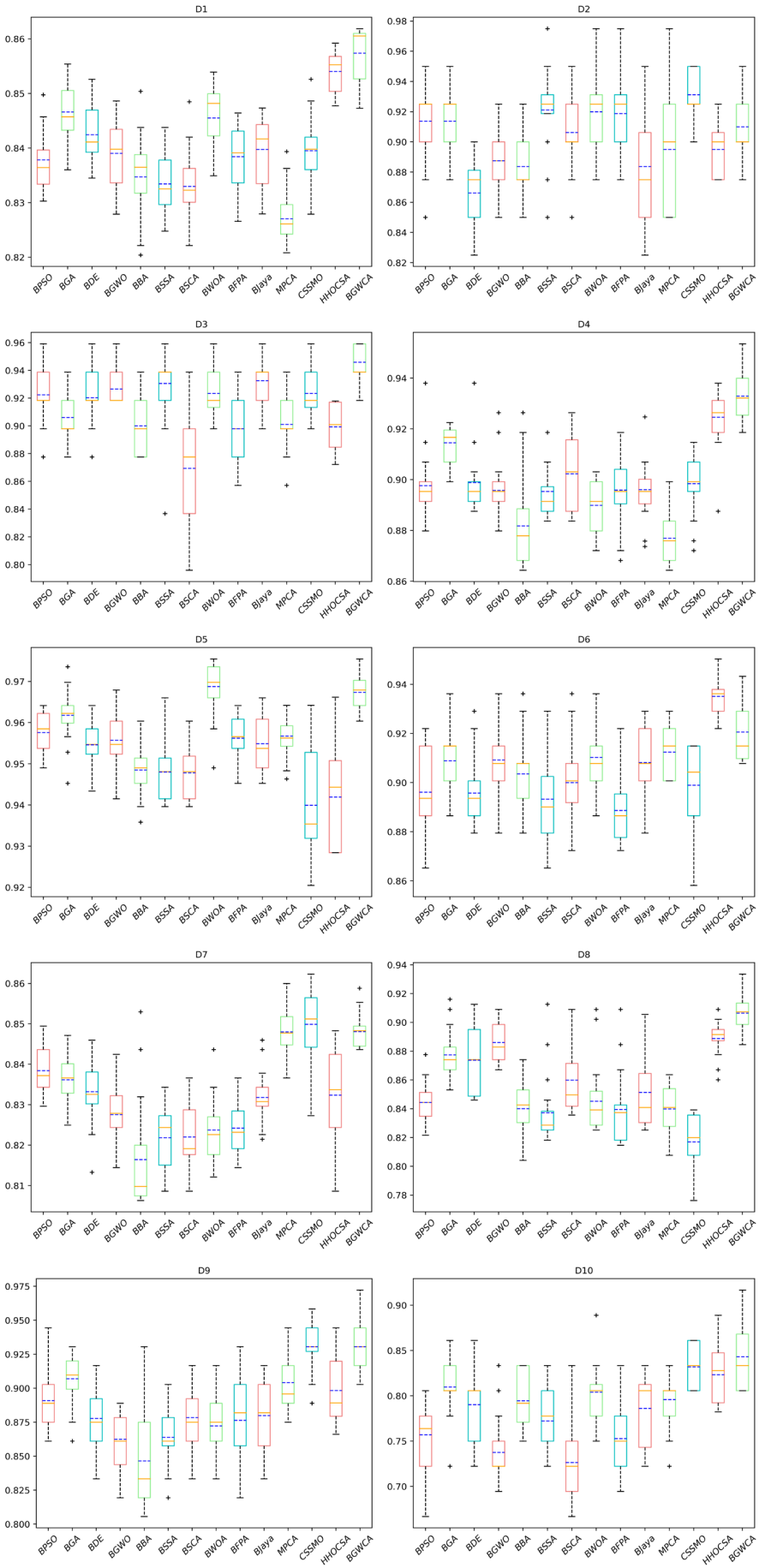

Figure 5. Boxplots comparing accuracy between proposed and other algorithms across datasets

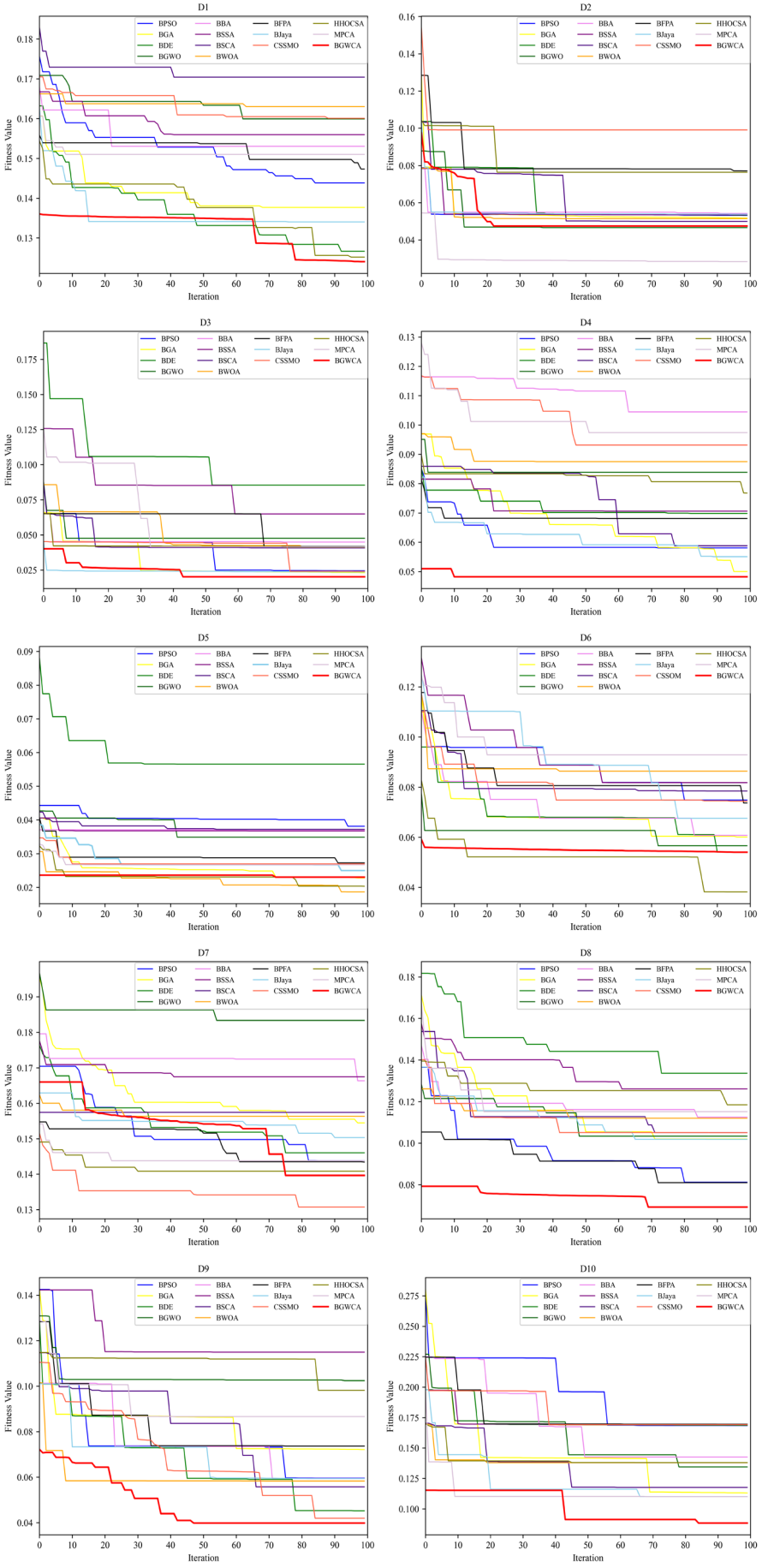

Figure 6. Comparison of convergence curves between proposed and other algorithms across datasets

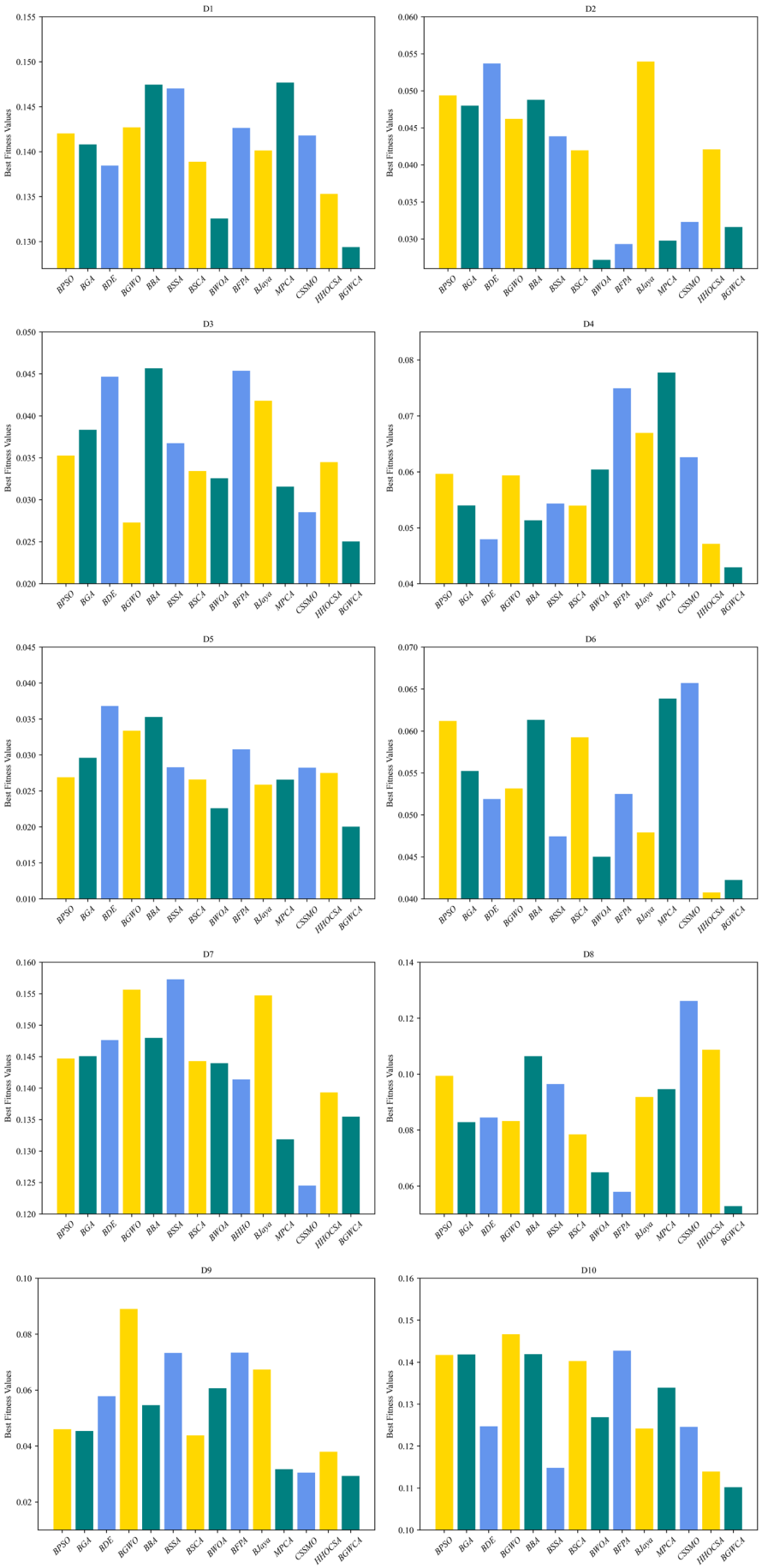

Figure 7. Optimal fitness values for the proposed algorithm and other various algorithms across various datasets

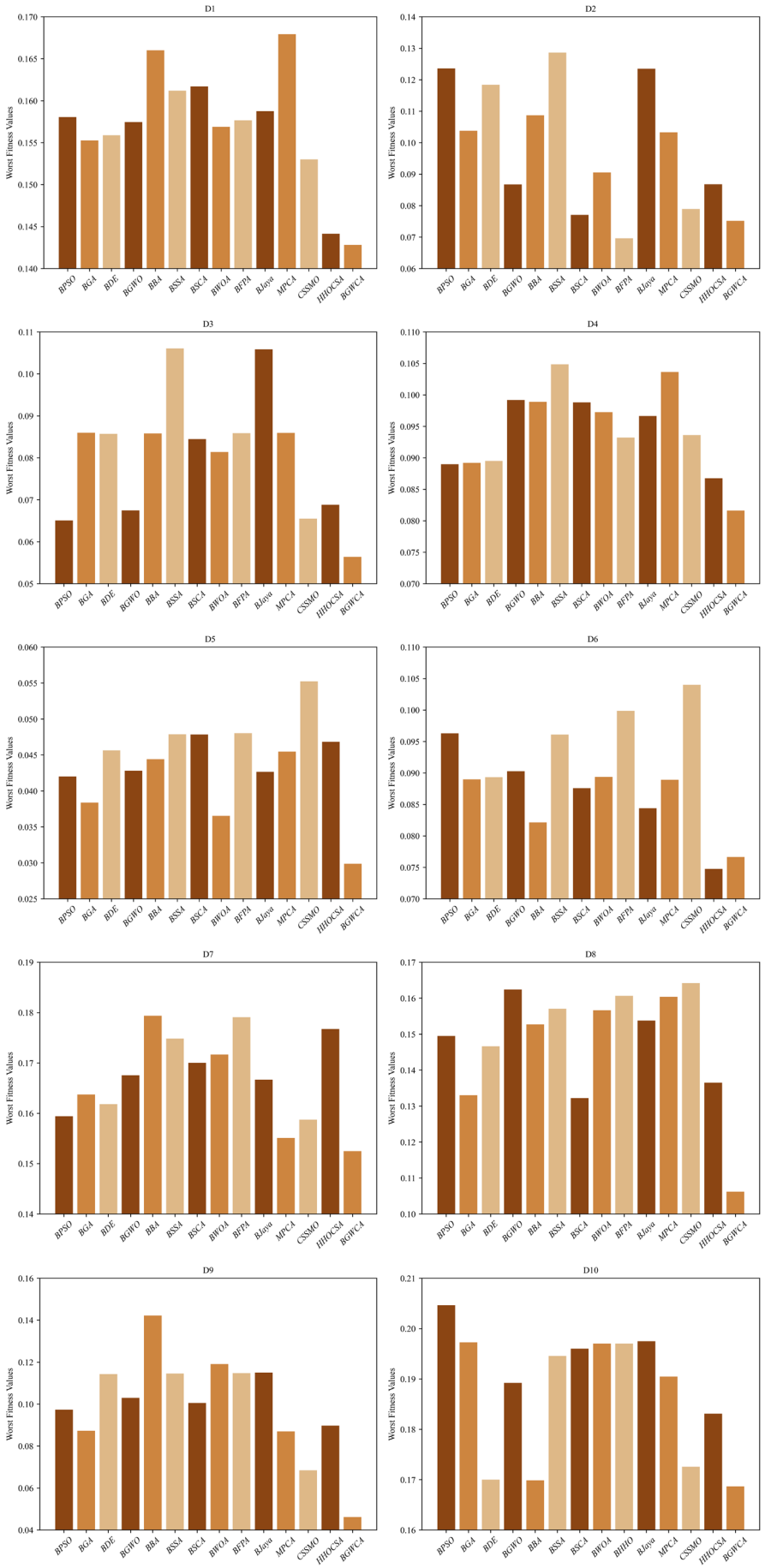

Figure 8. Minimum fitness values for the proposed algorithm and other various algorithms across various datasets

Table 7. Compare the suggested method with different methods in terms of the standard deviation (STD) of accuracy

Datasets	Meature	MPCA	CSSMO	HHOCSA	BGWCA
D1	Std	0.005257	0.005554	**0.004908**	0.005076
	Rank	3	4	1	2
D2	Std	0.037583	**0.018455**	0.020310	0.022741
	Rank	4	1	2	3
D3	Std	0.020686	0.019225	0.018564	**0.01369**
	Rank	4	3	2	1
D4	Std	0.011206	0.020648	0.260797	**0.010832**
	Rank	2	3	4	1
D5	Std	0.004823	0.005461	0.005107	**0.004138**
	Rank	2	4	3	1
D6	Std	0.011675	0.028966	**0.009587**	0.011984
	Rank	2	4	1	3
D7	Std	0.004867	0.006543	0.008440	**0.003924**
	Rank	2	3	4	1
D8	Std	0.017351	0.020810	0.015660	**0.012649**
	Rank	3	4	2	1
D9	Std	0.020078	**0.017894**	0.026294	0.018634
	Rank	3	1	4	2
D10	Std	0.028157	0.021336	0.022352	**0.032155**
	Rank	4	2	3	1
	Mean Rank	2.9	2.9	2.7	1.6
	Overall Rank	4	3	2	1

Table 8. Comparison of the performance of the VGG19 model with the model in this paper

Datasets	VGG19 (accuracy)	VGG19 (time)	Proposed model (accuracy)	Proposed model (time)
D1	**0.861**	14922.13s	0.857	**1023.45s**
D2	**0.942**	10556.36s	0.925	**836.67s**
D3	**0.959**	3689.18s	0.938	**237.12s**
D4	0.925	8765.24s	**0.945**	**398.45s**
D5	**0.97**	1462.72s	0.967	**89.42s**
D6	**0.93**	3211.91s	0.914	**187.6s**
D7	**0.87**	2863.16s	0.848	**168.52s**
D8	**0.925**	12907.72s	0.9	**965.16s**
D9	0.916	1390.92s	**0.925**	**93.73s**
D10	0.85	831.8s	**0.861**	**50.24s**

4 Conclusion

The binary classification problem is a critical challenge in machine learning with wide-ranging applications. This manuscript proposes a hybrid model using the VGG19 architecture for feature extraction and a dual-stage feature selection strategy. The study reduces redundant data, improving computational efficiency. In the second stage, the Binary Great Wall Construction Algorithm (BGWCA) is introduced to enhance the binary algorithm, significantly boosting performance. Experimental results on ten publicly available datasets demonstrate the model's superiority over other binary optimization strategies. Combining pre-trained models with evolutionary computation, this approach is efficient and resource-conscious.

However, BGWCA's relatively slow convergence, especially with high-dimensional feature data, increases computational complexity and makes the feature selection process more time-consuming. At the same time, movement strategies enhance solution diversity and add to the computational load. Multiple parameter settings significantly affect performance, and tuning them can incur additional costs. Additionally, though maintaining population diversity, the worker elimination mechanism may occasionally result in the loss of optimal solutions, affecting convergence.

Although the binary classification model proposed in this study shows significant advantages in feature selection and image classification tasks, there is infeasibility in practical applications when the model is directly applied to multiclassification problems. This is mainly due to the fact that the multi-classification task is more complex in terms of label processing and more diverse in terms of decision boundaries compared to the binary classification task. Meanwhile, the phenomenon of category imbalance in the multiclassification task can lead to a decrease in the classification accuracy of the model, especially when using strategies such as 'one-to-many' or 'one-to-one', and the superposition of multiple binary classification models can significantly increase the computational cost. In addition, error propagation between multiple models can also affect the final classification results. Therefore, future research should further optimise binary classification models to improve their applicability in multi-classification tasks and reduce their computational complexity.

Although the binary classification model in this study performs well on several publicly available datasets, future research can further optimize the BGWCA algorithm to improve its convergence speed and computational efficiency when processing high-dimensional feature data. In particular, introducing new acceleration strategies or algorithm structure improvements could be explored. In addition, the research can be extended to multi-class classification problems and adapted to a broader range of application scenarios, especially in areas such as medical image analysis and industrial vision inspection. Meanwhile, combining other advanced deep learning architectures (e.g., Transformer) for feature extraction

and the application of hybrid intelligence algorithms may further enhance the performance of feature selection. In addition, studying adaptive parameter optimization strategies will help improve the robustness and generality of the algorithms.

References

[1] R. Zakharov, P. Dupont, Ensemble logistic regression for feature selection, *Pattern Recognition in Bioinformatics: 6th IAPR International Conference, PRIB 2011, Delft, The Netherlands, November 2-4, 2011. Proceedings 6*, Delft, Netherlands, 2011, pp. 133–144.

[2] S. Zhou, T. Li, Y. Li, Recursive feature elimination based feature selection in modulation classification for mimo systems, *Chinese Journal of Electronics*, Vol. 32, No. 4, pp. 785–792, July, 2023.

[3] M. A. Chandra, S. S. Bedi, Survey on svm and their application in image classification, *International Journal of Information Technology*, Vol. 13, No. 5, pp. 1–11, October, 2021.

[4] J. Neumann, C. Schnörr, G. Steidl, Combined svm-based feature selection and classification, *Machine learning*, Vol. 61, No. 1–3, pp. 129–150, November, 2005.

[5] X. Lin, Z. Wu, J. Chen, L. Huang, Z. Shi, A credit scoring model based on integrated mixed sampling and ensemble feature selection: Rbr_xgb, *Journal of Internet Technology*, Vol. 23, No. 5, pp. 1061–1068, September, 2022.

[6] J. Deshmukh, U. Bhosle, A study of mammogram classification using adaboost with decision tree, knn, svm and hybrid svm-knn as component classifiers, *Journal of Information Hiding and Multimedia Signal Processing*, Vol. 9, No. 3, pp. 548–557, May, 2018.

[7] Z. Meng, K. Xia, Persistent spectral– based machine learning (perspect ml) for protein-ligand binding affinity prediction, *Science advances*, Vol. 7, No. 19, pp. 1–10, May, 2021.

[8] E. Dönmez, Enhancing classification capacity of cnn models with deep feature selection and fusion: a case study on maize seed classification, *Data & Knowledge Engineering*, Vol. 141, Article No. 102075, September, 2022.

[9] H. K. Suh, J. Ijsselmuiden, J. W. Hofstee, E. J. van Henten, Transfer learning for the classification of sugar beet and volunteer potato under field conditions, *Biosystems engineering*, Vol. 174, pp. 50–65, October, 2018.

[10] M. R. Islam, S. Sakamoto, Y. Yamada, A. W. Vargo, M. Iwata, M. Iwamura, K. Kise, Self-supervised learning for reading activity classification, *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, Vol. 5, No. 3, pp. 1–22, September, 2021.

[11] Q. Yang, S.-C. Chu, C.-C. Hu, L. Kong, J.-S. Pan, A task offloading method based on user satisfaction in c-ran with mobile edge computing, *IEEE Transactions on Mobile Computing*, Vol. 23, No. 4, pp. 3452–3465, April, 2024.

[12] S.-C. Chu, Z.-C. Dou, J.-S. Pan, L. Kong, V. Snášel, J. Watada, Dwsr: an architecture optimization framework for adaptive super-resolution neural networks based on meta-heuristics, *Artificial Intelligence Review*, Vol. 57, No. 2, Article No. 23, February, 2024.

[13] X.-D. Wang, R.-C. Chen, F. Yan, High-dimensional data clustering using k-means subspace feature selection, *Journal of Network Intelligence*, Vol. 4, No. 3, pp. 80–87, August, 2019.

[14] S. Mascarenhas, M. Agarwal, A comparison between vgg16, vgg19 and resnet50 architecture frameworks for image classification, *2021 International conference on disruptive technologies for multi-disciplinary research and applications (CENTCON)*, Bengaluru, India, 2021, pp. 96–99.

[15] L. A Belanche, F. F. González, Review and evaluation of feature selection algorithms in synthetic problems, *arXiv preprint*, arXiv:1101.2320, January, 2011. https://arxiv.org/abs/1101.2320

[16] S.-C. Chu, Z. Zhuang, J.-S. Pan, A. W. Mohamed, C.-C. Hu, Enhanced sparseea for large-scale multi-objective feature selection problems, *Complex & Intelligent Systems*, Vol. 10, No. 1, pp. 485–507, February, 2024.

[17] J.-S. Pan, L. Yue, S.-C. Chu, P. Hu, B. Yan, H. Yang, Binary bamboo forest growth optimization algorithm for feature selection problem, *Entropy*, Vol. 25, No. 2, Article No. 314, February, 2023.

[18] J.-S. Pan, A.-Q. Tian, S.-C. Chu, J.-B. Li, Improved binary pigeon-inspired optimization and its application for feature selection, *Applied Intelligence*, Vol. 51, No. 12, pp. 8661–8679, December, 2021.

[19] A. Jain, D. Zongker, Feature selection: Evaluation, application, and small sample performance, *IEEE transactions on pattern analysis and machine intelligence*, Vol. 19, No. 2, pp. 153–158, February, 1997.

[20] A. K. Jain, B. Chandrasekaran, 39 dimensionality and sample size considerations in pattern recognition practice, in: P. R. Krishnaiah, L. N. Kanal (Eds.), *Handbook of statistics, Volume 2*, 1982, pp. 835–855.

[21] L. A. Tawfeeq, S. S. Hussein, M. J. Mohammed, S. S. Abood, Predication of most significant features in medical image by utilized cnn and heatmap, *Journal of Information Hiding and Multimedia Signal Processing*, Vol. 12, No. 4, pp. 217–225, December, 2021.

[22] S. Das, Filters, wrappers and a boosting-based hybrid for feature selection, *ICML'01: Proceedings of the Eighteenth International Conference on Machine Learning*, Williamstown, MA, USA, 2001, pp. 74–81.

[23] W. Duch, T. Winiarski, J. Biesiada, A. Kachel, Feature selection and Discretization, *International conference on artificial neural networks (ICANN) and International conference on neural information processing (ICONIP)*, 2003, Istanbul, Turkey, pp. 251–254.

[24] Z. Wang, Y. Zhang, Z. Chen, H. Yang, Y. Sun, J. Kang, Y. Yang, X. Liang, Application of relieff algorithm to selecting feature sets for classification of high resolution remote sensing image, *2016 IEEE international geoscience and remote sensing symposium (IGARSS)*, Beijing, China, 2016, pp. 755–758.

[25] Y. Zhang, C. Ding, T. Li, Gene selection algorithm by combining relieff and mrmr, *BMC genomics*, Vol. 9, No. Supplement 2, pp. 1–10, September, 2008.

[26] N. Spolaôr, M. C. Monard, Evaluating relieff-based multi-label feature selection algorithm, *Advances in Artificial Intelligence–IBERAMIA 2014: 14th Ibero-American Conference on AI, Santiago de Chile, Chile, November 24-27, 2014, Proceedings 14*, Santiago de Chile, Chile, 2014, pp. 194–205.

[27] M. Robnik-Šikonja, I. Kononenko, Theoretical and empirical analysis of relieff and rrelieff, *Machine learning*, Vol. 53, No. 1–2, pp. 23–69, October, 2003.

[28] K. Kira, L. A. Rendell, A practical approach to feature

selection, *Machine learning proceedings 1992*, Aberdeen, United Kingdom, 1992, pp. 249–256.

[29] Z. Guan, C. Ren, J. Niu, P. Wang, Y. Shang, Great wall construction algorithm: A novel meta-heuristic algorithm for engineer problems, *Expert Systems with Applications*, Vol. 233, Article No. 120905, December, 2023.

[30] N. Yu, G. Delrieu, B. Boudevillain, P. Hazenberg, R. Uijlenhoet, Unified formulation of single- and multimoment normalizations of the raindrop size distribution based on the gamma probability density function, *Journal of Applied Meteorology and Climatology*, Vol. 53, No. 1, pp. 166–179, January, 2014.

[31] C. Cortes, V. Vapnik, Support-vector networks, *Machine learning*, Vol. 20, No. 3, pp. 273–297, September, 1995.

[32] H. Yin, P. Fu, Z. Sun, Face feature selection and recognition using separability criterion and binary particle swarm optimization algorithm, *Chinese Journal of Electronics*, Vol. 23, No. 2, pp. 361–365, April, 2014.

[33] P. Hu, J.-S. Pan, S.-C. Chu, C. Sun, Multi-surrogate assisted binary particle swarm optimization algorithm and its application for feature selection, *Applied soft computing*, Vol. 121, Article No. 108736, May, 2022.

[34] M. Z. Baig, N. Aslam, H. P. Shum, L. Zhang, Differential evolution algorithm as a tool for optimal feature subset selection in motor imagery eeg, *Expert Systems with Applications*, Vol. 90, pp. 184–195, December, 2017.

[35] J.-S. Pan, N. Liu, S.-C. Chu, A competitive mechanism based multi-objective differential evolution algorithm and its application in feature selection, *Knowledge-Based Systems*, Vol. 245, Article No. 108582, June, 2022.

[36] T. Zhou, H. Lu, W. Wang, X. Yong, Ga-svm based feature selection and parameter optimization in hospitalization expense modeling, *Applied soft computing*, Vol. 75, pp. 323–332, February, 2019.

[37] X. Yuan, J.-S. Pan, A.-Q. Tian, S.-C Chu, Binary sparrow search algorithm for feature selection, *Journal of Internet Technology*, Vol. 24, No. 2, pp. 217–232, March, 2023.

[38] M. Abdel-Basset, D. El-Shahat, I. El-Henawy, V. H. C. De Albuquerque, S. Mirjalili, A new fusion of grey wolf optimizer algorithm with a two-phase mutation for feature selection, *Expert Systems with Applications*, Vol. 139, Article No. 112824, January, 2020.

[39] S. Taghian, M. H. Nadimi-Shahraki, Binary sine cosine algorithms for feature selection from medical *data, arXiv preprint*, arXiv:1911.07805, November, 2019. https://arxiv.org/abs/1911.07805

[40] M. Mafarja, S. Mirjalili, Whale optimization approaches for wrapper feature selection, *Applied Soft Computing*, Vol. 62, pp. 441–453, January, 2018.

[41] H. Das, B. Naik, H. S. Behera, A jaya algorithm based wrapper method for optimal feature selection in supervised classification, *Journal of King Saud University-Computer and Information Sciences*, Vol. 34, No. 6, pp. 3851–3863, June, 2022.

[42] D. Rodrigues, X.-S. Yang, A. N. De Souza, J. P. Papa, Binary Flower Pollination Algorithm and Its Application to Feature Selection, in: X.-S. Yang (Ed.), *Recent advances in swarm intelligence and evolutionary computation*, Springer Publishing Company, Incorporated, 2015, pp. 85–100.

[43] R. Y. Nakamura, L. A. Pereira, K. A. Costa, D. Rodrigues, J. P. Papa, X. S. Yang, Bba: a binary bat algorithm for feature selection, *2012 25th SIBGRAPI conference on graphics, patterns and images*, Ouro Preto, Brazil, 2012, pp. 291–297.

[44] A. Saxena, S. S. Chouhan, R. M. Aziz, V. Agarwal, A comprehensive evaluation of Marine predator chaotic algorithm for feature selection of COVID-19, *Evolving Systems*, Vol. 15, No. 4, pp. 1235–1248, August, 2024.

[45] R. Mahto, S. U. Ahmed, R. ur Rahman, R. M. Aziz, P. Roy, S. Mallik, A. Li, M. A. Shah, A novel and innovative cancer classification framework through a consecutive utilization of hybrid feature selection, *BMC bioinformatics*, Vol. 24, No. 1, Article No. 479, December, 2023.

[46] A. A. Joshi, R. M. Aziz, A two-phase cuckoo search based approach for gene selection and deep learning classification of cancer disease using gene expression data with a novel fitness function, *Multimedia Tools and Applications*, Vol. 83, No. 28, pp. 71721–71752, August, 2024.

[47] A. Yaqoob, N. K. Verma, R. M. Aziz, A. Saxena, Enhancing Feature Selection Through Metaheuristic Hybrid Cuckoo Search and Harris Hawks Optimization for Cancer Classification, in: K. Kalita, N. Ganesh, S. Balamurugan (Eds.), *Metaheuristics for Machine Learning: Algorithms and Applications*, Wiley, 2024, pp. 95–134.

Biographies



Shu-Chuan Chu received her Ph.D. from Flinders University, Australia, in 2004. Since 2009, she has been a Research Fellow at Flinders University and later at Shandong University of Science and Technology. Her research focuses on swarm intelligence, intelligent computing, and data mining.



Xiaoping Cao received the B.S. degree from Shandong University of Science and Technology, in 2023. He is currently pursuing the M.S. degree with Shandong University of Science and Technology, Qingdao, China. Her research interests include meta-heuristic algorithms and image classification.



Zhi-Chao Dou received his B.S. degree. B.S. in Software Engineering from Yantai Nanshan College. Master degree in Software Engineering from Shandong University of Science and Technology. His current research interests include population intelligence and image processing.



Jie Wu received the M.Eng. from Hubei University of Technology in 2005 and the Ph.D. from VŠB-TU Ostrava in 2012. Since 2017, he has been a Visiting Professor at the University of Kentucky and is currently an Associate Professor at Zhejiang University of Water Resources and Electric Power, focusing on AI and electrical engineering.



Jeng-Shyang Pan received the B.S. from NTUST in 1986, the M.S. from NCTU in 1988, and the Ph.D. from the University of Edinburgh in 1996. He is a Professor at SDUST, an IET Fellow, and a former Vice Chair of the IEEE Tainan Section.