Binary Enhanced White Shark Optimization Algorithm and Its Application to Semi-Supervised Feature Selection

Jian Zhou*, Jiawen Yang
School of Management Engineering, Qingdao University of Technology, Qingdao, 266520, China

***Abstract*—With the development of information technology, the generation and storage of data have become convenient, resulting in a large amount of high-dimensional data. Due to the high acquisition cost of labeled data, semi-supervised learning of partially labeled data has emerged. Feature selection aims to select the most representative and informative feature subset from the original feature set and avoid the curse of dimensionality. By utilizing semi-supervised feature selection, valuable information from unlabeled data can be better extracted, thereby enhancing the model's robustness. White Shark Optimizer is a commonly used searching method, but it may fall into a local optimum when dealing with semi-supervised feature selection problems. This study proposes an Enhanced White Shark Optimization (EWSO) with a dynamic dual elite strategy and sinusoidal mutation operation. The former can make full use of the current optimal location information and improve the ability to find potential optimal solutions, and the latter can help the algorithm jump out of the local optimum and improve the accuracy of the algorithm's convergence. For dealing with discrete problems, we further introduce a binary mapping function to the EWSO, abbreviated as BEWSO. Compared with six other meta-heuristic feature selection techniques on nine high-dimensional datasets, experimental results show that the proposed method is superior to other methods. Under 80% and 60% unlabeled data, the average accuracies are 5.03% and 3.64% higher than the suboptimal value, respectively.**

***Keywords*—Feature selection; semi-supervised learning; meta-heuristic algorithm; white shark optimization**

I. Introduction

The swift progress in information technology has resulted in the accumulation of large amounts of data, which holds significant value. Data mining [1], as an important technical means, helps people to obtain value from data by discovering the patterns and knowledge hidden behind the data. However, on account of the large number and dimension of generated data, it leads to the curse of dimensionality, increasing the computational complexity and learning time. Therefore, how to obtain useful information from the data becomes an important challenge. Feature selection [2] is an effective method for dimensionality reduction, as it involves choosing the feature subsets by eliminating irrelevant and redundant ones. In real-world classification situations, obtaining data samples can be challenging due to time and resource limitations, particularly when labeling data is expensive or difficult. Thus, the importance of semi-supervised learning [3] becomes apparent. It can effectively use limited supervised information and a large amount of unlabeled data to enhance feature selection and thus improve the accuracy of learning models.

Feature selection involves choosing N features from a total of M features to enhance a specific system index, which is a key step for decreasing the dataset's dimensionality and boosting the learning algorithm's performance. By identifying the impactful features, it reduces the size and complexity of the dataset, while maintaining or improving the performance of the model. This process not only helps reduce the training time for the model, but also simplifies it, making it easier to understand and interpret. In addition, with feature selection, overfitting can be avoided, as fewer features mean that the model is not overly complex, thus avoiding the curse of dimensionality.

Regarding the usability of label information, feature selection methods can be divided into three types: supervised, semi-supervised, and unsupervised feature selection. The data generated in many practical applications has a great deal of unlabeled data samples and a limited number of labeled data samples. In fact, it requires significant time and effort for users to label each data sample. Consequently, the limited labeled samples may not provide enough information for effective supervised selection, while unsupervised methods overlook the valuable labeling information. In conclusion, semi-supervised feature selection, which considers both labeled and unlabeled data, can effectively identify the optimal subset of features.

The feature selection task is a non-deterministic polynomial-hard problem [4]. As the number of features grows, the time complexity cost of the relevant ideal search becomes unacceptable. Meta-heuristic methods are effective for addressing optimization challenges like feature selection because of their good global searching ability, self-adaptability, and computational efficiency. These algorithms can be grouped into four main types: evolution-based, swarm intelligence-based, physics-based, and human-based algorithms.

The evolution-based algorithms typically utilize genetic manipulation and a selection mechanism of individuals to optimize the solution, which may cause fall into a local optimal solution during the search process. The physics-based algorithms apply physical principles to simulate and optimize problems, which may be affected by the accuracy of physical models. The human-based algorithms often use human decision and search strategies, which may be influenced by human cognition and experience. In contrast, the swarm intelligence-based algorithms leverage the behavior of biological or social groups to simulate and optimize problems, and find better

*Corresponding author.

solutions by cooperating and sharing information, which makes wider applicability and flexibility to solve complex optimization problems.

The White Shark Optimization (WSO) Algorithm [5] is a recent swarm intelligence algorithm inspired by the predation behavior of great white sharks. It aims to address optimization problems by simulating the search strategy of great white sharks. This algorithm is used to locate and track the prey of great white sharks, utilizing their acute hearing, sense of smell, and understanding of fish behavior. It simulates the movement and adjustment of the great white shark in the search space to identify optimal solutions for small, medium, and high-dimensional optimization problems.

The innovations and contributions of this study are summarized as follows:

- The WSO method faces challenges in effectively balancing exploration and development. Its convergence speed is slow and easy to fall into local optimization. This study proposed an Enhanced White Shark Optimization (EWSO) Algorithm with a dynamic dual-elite strategy and sinusoidal mutation operation to improve its performance.
- We further introduce a binary mapping function to the EWSO, abbreviated as BEWSO. It adopts KNN-self-training as a semi-supervised classification model to further explore the binary version of the proposed algorithm in the field of semi-supervised feature selection.
- Based on high-dimensional datasets, the advantages of the proposed algorithm in semi-supervised feature selection optimization problems are verified. When compared to other meta-heuristic feature selection techniques, the average accuracies have improved by 5.03% and 3.64% based on 80% and 60% unlabeled data, respectively.

The rest of this article is organized in the following manner. Section II introduces a review of the literature on semi-supervised feature selection techniques. Section III provides a new Enhanced White Shark Optimizer algorithm and its binary version. Section IV presents a comparison of the results derived from the implementations. Finally, Section V contains the summary and guidelines for future research.

II. Literature Review

The semi-supervised feature selection method focuses on utilizing a limited number of labeled samples alongside a larger set of unlabeled samples for classification tasks. This approach can be categorized into three main types: filter, wrapper, and embedded methods. This section will primarily discuss these techniques.

A. Semi-Supervised Filter Feature Selection

Filtered semi-supervised feature selection leverages the intrinsic characteristics of both labeled and unlabeled data to evaluate features before conducting learning tasks.

This approach primarily encompasses several methods, including those based on Laplacian scores, pairwise constraints, Fisher criteria, and sparse models.

The Laplacian method is utilized to assess and select features by evaluating how well they preserve the proximity between samples in a dataset. Zhao et al. [6] introduced a method that integrates Laplacian scoring with locally sensitive discriminant analysis. This approach constructs in-class and inter-class graphs using semi-supervised datasets. Cheng et al. [7] developed a method that merges semi-supervised Laplacian scores with classification information gain. Additionally, Doquire et al. [8] presented an algorithm tailored for regression tasks, which is based on the concept of Laplace fractions.

Pairwise constraints represent a different form of supervised information. Benabdeslem et al. [9] introduced a method that integrates Laplacian score and constraint score, allowing for an effective assessment of features' ability to maintain local structure and constraints. Kalakech et al. [10] developed constraint scores in semi-supervised regression interval problems by transforming continuous label data into paired constraint information.

Fisher score is a method for measuring the ability to discriminate feature categories. Additionally, the semi-supervised feature selection challenge can be addressed by leveraging the information from unlabeled samples using Fisher's score. Yang et al. [11] introduced a semi-supervised Fisher scoring approach that utilizes both global distribution and local structure information from the dataset. Sun et al. [12] developed an algorithm based on Fisher criteria and the manifold hypothesis, which redefines the interclass scatter matrix to maximize the separation between classes, making it more resilient to complex data distributions.

By incorporating a sparsity constraint, the sparse model can automatically identify key features during training, enhancing computational efficiency. Han et al. [13] proposed a method that combines a divergence matrix with a sparse model. Zeng et al. [14] employed the l2, 1-norm to promote sparsity and introduced a semi-supervised method focused on local discriminant information.

In summary, the filter method evaluates feature subsets using a separate index based on the original features or statistical properties about the data, without integrating a machine learning model into the feature selection process. This leads to a model with high generalization but lower accuracy.

B. Semi-Supervised Wrapper Feature Selection Method

Semi-supervised wrapper feature selection methods utilize either a single learning model or an ensemble learning model to predict class labels for unlabeled data while evaluating the validity of selected feature subsets.

For single learning models, Ren et al. [15] introduced a wrapper forward framework based on semi-supervised feature selection. This method involves training classifiers with selected features and using those classifiers to predict missing labels. Unlabeled data is then randomly chosen and combined with labeled data to create a new dataset.

This process is carried out several times to identify the feature that appears frequently, which is then added to the feature subset. Song et al. [16] explored a novel particle swarm optimization algorithm for variable-size coevolutionary. They introduced a spatial partitioning strategy on feature importance, allowing related features to be grouped into the same subspace with reduced computational costs.

In the context of ensemble learning, Bellal et al. [17] introduced a guided feature ordering method known as semi-supervised ensemble learning. This method combined the bagged integration and out-of-bag feature importance measure. Barkia et al. [18] developed a novel semi-supervised feature importance evaluation method that employed a cooperative training strategy to train an integrated classifier, extending the feature importance evaluation method from the standard random forest model to semi-supervised scenarios. Han et al. [19] suggested a new method based on integration. It used an integrated classifier capable of estimating the confidence of unlabeled data to explore wrapper feature selection, so that more relevant feature subsets can be selected.

In conclusion, the wrapper approach assesses various feature subsets of a specific model through a designated learning algorithm and identifies the subset that yields the best model performance. While this results in high model accuracy, it tends to have low generalizability. Additionally, this method is more resource-demanding compared to the filter method, as it necessitates multiple iterations of the model to verify the effectiveness of the feature subset based on accuracy.

C. Semi-Supervised Embedded Feature Selection

Semi-supervised embedded feature selection employs both labeled and unlabeled data to choose features during the training process. Key concepts involved include sparse modeling, the graph Laplacian operator, and manifold regularization, among others.

Ma et al. [20] introduced a technique that leverages sparse models and the Laplace operator of graphs, primarily focusing on graph-based semi-supervised learning, along with various sparse models for feature selection. Shi et al. [21] presented an approach that utilizes the l2,1/2 matrix norm and graph Laplacian, introducing an effective optimization algorithm for web image annotation. Xu et al. [22] developed a regularization-based algorithm that selects features by enhancing the classification margin between various classes, taking into account the geometric structure of the probability distribution for both labeled and unlabeled data.

Additionally, Feofanov et al. [23] proposed a hybrid method based on Hadoop, within a semi-supervised embedded framework to address outlier and local optimization issues. Chen et al. [24] introduced a method that relies on rescaled linear regression, using the label representation matrix of unlabeled samples as the optimization target during the training of the sparse linear regression model. Yuan et al. [25] proposed a method called discriminant semi-supervised feature selection.

The embedded method integrates feature selection with the learning algorithm, allowing the feature selection and learning model processes to occur simultaneously. This approach reduces time and space requirements, enhances classification accuracy, and strikes a balance between filter and wrapper methods. Consequently, this study adopts the embedded method for feature selection.

III. Binary Enhanced White Shark Optimizer

A. Enhanced White Shark Optimizer

The White Shark Optimizer algorithm may face limitations in its search performance when applied to feature selection tasks, exhibiting slow convergence and a tendency to get stuck in local optima. Therefore, this study introduces two methods to enhance the algorithm: a dynamic double elite strategy and sinusoidal mutation operation. The dynamic dual-elite strategy optimizes the search process of the algorithm by introducing two elite individuals: the current optimal position and the suboptimal position. This strategy leverages the information from the best position and enables the algorithm to reach a high-quality solution more rapidly. At the same time, by utilizing the suboptimal position, the algorithm is capable of preserving the diversity of the population during the iteration process, allowing for broader exploration of the solution space, thus enhancing the ability to discover potential optimal solutions. The sinusoidal mutation operation is another important improvement strategy, utilizing the periodic change characteristic of the sinusoidal function to guide the individuals in the algorithm to mutate. This kind of mutation operation can make the algorithm more flexible and intelligent in the search process, allowing it to avoid getting stuck in local optima. Through the sinusoidal mutation operation, the algorithm can continuously find new and possibly better positions in the solution space, thus enhancing convergence accuracy and global search ability. These two methods are described below.

1) Dynamic dual elite strategy: The elite learning strategy is a widely used approach in optimization and evolutionary computing that focuses on retaining and utilizing high-performing individuals within a population to enhance the search process and solution quality. The core idea is to preserve the best individuals in each generation so that they can enter the next generation, thus avoiding the replacement deletion of the best individuals in the genetic iteration process. In this way, this approach allows the algorithm to reach a global optimal solution or a solution close to it more rapidly. Chen et al. [26] introduced elite opposition learning to modify the sine and cosine algorithm, enhancing population diversity. For the multi-objective scheduling problem, Amer et al. [27] proposed an improved Harris Eagle Optimizer that improves the quality of the standard Harris Hawk Optimization algorithm by using a refined opposition learning method.

Building on the enhanced algorithm mentioned earlier, this study presents a dynamic dual-elite learning strategy. This approach leverages the current optimal and suboptimal positions within the population to create the next generation of candidate solutions, which accelerates convergence and enhances the ability to identify potential optimal solutions. The strategy is chosen for the following reasons. First, by retaining and utilizing two elite individuals, it provides more search directions and possibilities, thereby increasing its convergence accuracy. Second, the number and function of elite individuals can be

adjusted dynamically based on the algorithm's operational context, allowing it to better adapt to changing environments and enhancing its robustness and adaptability. In addition, dynamic learning factors are introduced. This factor is closely linked to the fitness value of the elite white shark, and the weight of the two elite individuals can be controlled by dynamically adjusting its value. The equation for creating potential solutions and adjusting learning factors dynamically is described below.

$$X_i^*(t+1) = r_1 * F * \left(X_{second}(t) - X_i(t)\right) - r_2 * (1-F) * X_{best}(t) \quad (1)$$

$$F = \frac{F_{best}(t)}{F_{best}(t)+F_{second}(t)} \quad (2)$$

where, r_1 and r_2 are random vectors within [0,1], and $X_{best}(t)$ and $X_{second}(t)$ are individual positions with optimal and sub-optimal fitness values at the t iteration, respectively. The dynamic learning factor is denoted as F, and $F_{best}(t)$ and $F_{second}(t)$ represent the fitness values of the best and sub-best individual positions during the t iteration, respectively.

We use a selection mechanism based on hybrid sorting. First, $2N$ white sharks $X_1(t), X_2(t), \dots, X_N(t)$ and $X_1^*(t), X_2^*(t), \dots, X_N^*(t)$ are combined and sorted based on their fitness values. Then, the first N white sharks with the best fitness values were selected for the next generation, thus conducive to better population evolution. The advantages of this hybrid sorting-based selection mechanism are as follows. First, compared with traditional methods of selection based only on the original population, the mechanism doubles the number of great white sharks involved in selection during each iteration. This not only broadens the selection range but also helps identify and utilize more individuals with potentially outstanding traits, thereby significantly enhancing population diversity and evolutionary effectiveness. Secondly, this mechanism allows the algorithm to constantly explore new solution spaces throughout the search process. By introducing hybrid competition between elite and regular individuals in each iteration, the algorithm's global search capability is enhanced, leading to a higher likelihood of ultimately discovering the global optimal solution. After evaluating the fitness of each white shark in the initial population, the dynamic dual-elite strategy is applied to generate the next generation of candidate solutions based on the current best and second-best positions of the white shark population. The population is then updated according to their fitness values. The corresponding pseudocode for this algorithm is outlined in Algorithm 1.

Algorithm 1 Dynamic dual elite strategy for enhanced white shark optimizer

Initialize population size *N*;
Create the initial population *X*;
Evaluate the fitness of the initial population *X*;
for $i < N$ **do**
  Generate the learning white shark X^* using Eq. (1);
  Calculate the fitness value of X^*;
end for
Obtain the population and fitness of *2N* white sharks;
Sort them according to their fitness values;
Update population X' by selecting the best *N* white sharks with the best fitness;

2) Sinusoidal mutation operation: The sine function, with its inherent periodicity and continuous and smooth change characteristics, is widely known and highly valued in the field of science and engineering. Recently, advancements in computing technology and algorithm theory have led to various algorithms that incorporate the sine function concept in order to obtain significant enhancement and improvement in performance. For instance, Draa et al. [28] utilized a compound sine formula to modify the scale factor and crossover rate in the differential evolution algorithm. Similarly, Khalilpourazari et al. [29] introduced an enhanced crow search algorithm for complex optimization challenges by integrating sine and cosine functions.

Inspired by the above methods, the proposed algorithm uses the sinusoidal mutation theory to improve. Sinusoidal mutation theory is a method to guide the optimal individual to the mutation operation, which helps the algorithm escape local optima and increases the chances of discovering better solutions. When the sine function has the value [−1,1] on the interval [0,2π], the algorithm focuses on advancing the next generation of individuals within promising areas of the search space, instead of exploring a broader region. Therefore, the algorithm can be studied in a small range around the current global optimal location, improving the local expansion ability of the algorithm and avoiding falling into a local optimum. In addition, the algorithm aims to balance the critical requirements of exploration and exploitation. On the one hand, it actively explores the unknown and possibly better solution domain through the periodic fluctuation characteristics of the sinusoidal mutation process. On the other hand, when the sine wave amplitude drops to the local lowest point, the algorithm can accurately focus on the small change near the known best point. Eq. (3) illustrates the sinusoidal mutation process of the current globally optimal great white shark.

$$X'_{best} = \sin\left(2\pi * rand(1,\dim)\right) \oplus X_{best}, if\ rand < c \quad (3)$$

$$X_{best} = \begin{cases} X'_{best}, f(X'_{best}) < f(X_{best}) \\ X_{best},\ otherwise \end{cases} \quad (4)$$

where, X_{best} denotes the global optimal position, $\oplus$ signifies the dot product, and c is the mutation probability. As shown in Eq. (3), if the fitness value of the individual after the sinusoidal mutation improves, the result is accepted. Otherwise, the original position is maintained.

After the keen hearing, smell, and fish behavior of the great white shark, the sinusoidal mutation operation is applied to identify the global optimal position X_{best}. Initially, the sinusoidal mutation operation generates a new position, and its corresponding fitness value is compared with that of the global optimal position X_{best}, leading to an update of the algorithm's optimal solution. The pseudocode for this process is outlined in Algorithm 2.

Algorithm 2 Sinusoidal mutation for enhanced white shark optimizer

Obtain the global best position X_{best};
if $rand < c$ **then**
  Generate mutation white shark X'_{best} using Eq. (3);
  Calculate the fitness value of X'_{best};
  if $f(X'_{best}) < f(X_{best})$
    $X_{best} = X'_{best}$;
    $f\left(X_{best}\right) = f(X'_{best})$;
  end if
end if

B. Binary Enhanced White Shark Optimizer

White Shark Optimizer is a powerful algorithm designed specifically for continuous optimization problems. It can find the optimal solution by constantly adjusting variable values in a continuous search space. However, when dealing with feature selection problems, the situation is quite different. The search space in these cases is discrete rather than continuous, which means that methods suitable for continuous spaces may not be directly applied. In particular, the search space for feature selection is depicted in binary form, where 0 indicates features that are not selected, and 1 signifies those that are included. To effectively tackle feature selection issues, it is necessary to modify and optimize the enhanced white shark optimizer algorithm to accommodate binary variables. This adjustment is crucial for ensuring the algorithm's relevance and effectiveness in feature selection tasks.

In this context, transfer functions play a vital and prevalent role. These mathematical tools can efficiently map continuous values to discrete ones, particularly in converting a single continuous value into a binary state to fulfill the specific needs of feature selection. Among various transfer functions, the S-type and V-type are the most commonly utilized and effective [30]. The S-type transfer function is known for its smooth transitions and monotonically increasing nature, effectively representing the gradual change in feature importance and facilitating a smooth shift from 0 to 1 at a defined threshold. Conversely, the V-type transfer function, with its symmetrical "V" shape, is better suited for detailed feature classification, especially when feature importance is evenly distributed or has multiple peaks, allowing for more precise capture of these features. The details are shown in Table I.

In this research, we transformed continuous white shark feature data into binary format using a two-step transfer function approach, specifically employing the S1 transfer function, to enhance compatibility with subsequent feature selection and processing tasks.

The first step involves calculating the probability of the binary conversion value based on Eq. (5), $w_d^i(t)$ denotes the position of the ith white shark in the dth dimension in the tth iteration.

$$P\left(w_d^i(t)\right) = TF(w_d^i(t)) \tag{5}$$

TABLE I. S-TYPED AND V-TYPED TRANSFER FUNCTIONS

S-typed	Transfer function	V-typed	Transfer function
S1	$T(x) = \frac{1}{1+e^{-2x}}$	V1	$T(x) = \lvert\text{erf}(\frac{\sqrt{\pi}}{2}x)\rvert$
S2	$T(x) = \frac{1}{1+e^{-x}}$	V2	$T(x) = \lvert\tanh(x)\rvert$
S3	$T(x) = \frac{1}{1+e^{(-x/2)}}$	V3	$T(x) = \lvert(x)/\sqrt{1+x^2}\rvert$
S4	$T(x) = \frac{1}{1+e^{(-x/3)}}$	V4	$T(x) = \lvert\frac{2}{\pi}\arctan(\frac{\pi}{2}x)\rvert$

$$TF(w) = \frac{1}{1+e^{-2w}} \tag{6}$$

Secondly, the probability of Eq. (5) is substituted into Eq. (7), and the value of each great white shark's location is converted to binary form.

$$w_d^i(t) = \begin{cases} 1, & if\ P\left(w_d^i(t)\right) \geq rand, \\ 0, & otherwise \end{cases} \tag{7}$$

The flow chart in Fig. 1 illustrates the corresponding binary algorithm.

Feature selection is typically considered a multi-objective optimization problem, and its core challenge is how to find a balance point that maximizes classification accuracy while reducing the number of features chosen. This balance is essential for enhancing the efficiency, interpretability, and generalization of machine learning models. The fitness function is crucial in this context, serving as the primary criterion for assessing the selected feature subset. Specifically, the fitness function outlined in Eq. (8) is essential for evaluating the selected subset of features, as it balances the goal of maximizing classification accuracy with the need to minimize the number of features used in the learning model. The smaller the function value, the better the performance.

$$F = \alpha * (1 - acc) + \beta * \frac{l_f}{n_f} \tag{8}$$

The classification accuracy depends on the selected classifier, which is represented by acc in this study. The parameters α and β are inverse parameters, representing the weight given to the quality of classification and the feature selection ratio, respectively, ranging from 0 to 1, where $\beta = 1 - \alpha$. In this study, α is 0.99 and β is 0.01. The values of l_f and n_f refer to the number of selected features and the sum of features in the dataset, respectively.

Several semi-supervised learning models have been developed to enhance learning efficiency and accuracy by utilizing a small amount of labeled data in conjunction with a larger volume of unlabeled data. These models include self-training classification models, mixed classification models based on large expectations, cooperative training models, graph-based semi-supervised learning models, and semi-supervised learning clustering models. Among these, the self-training classification model is popular due to its effective training results and relatively straightforward implementation. The K-

Nearest Neighbors (K-NN) algorithm is an efficient classification method known for its simplicity, ease of use, strong predictive performance, insensitivity to outliers, and suitability for large datasets. As a result, it is frequently applied in areas such as text sentiment analysis, risk assessment, and medical diagnosis. To sum up, this study adopts KNN-self-training as a semi-supervised classification model algorithm, with k set to 5. Each run utilizes 5-fold cross-validation to evaluate model accuracy. Initially, 20% of the data is chosen as the test dataset, with the remainder serving as the training dataset. Given the semi-supervised nature of the algorithm, the experiment randomly selects 80% and 60% of the training dataset as unlabeled data (class labels are removed), and the rest as labeled data for model training, respectively. Finally, after making predictions on the unlabeled data, 20% of the test data is used as experimental results to verify the classification model accuracy.

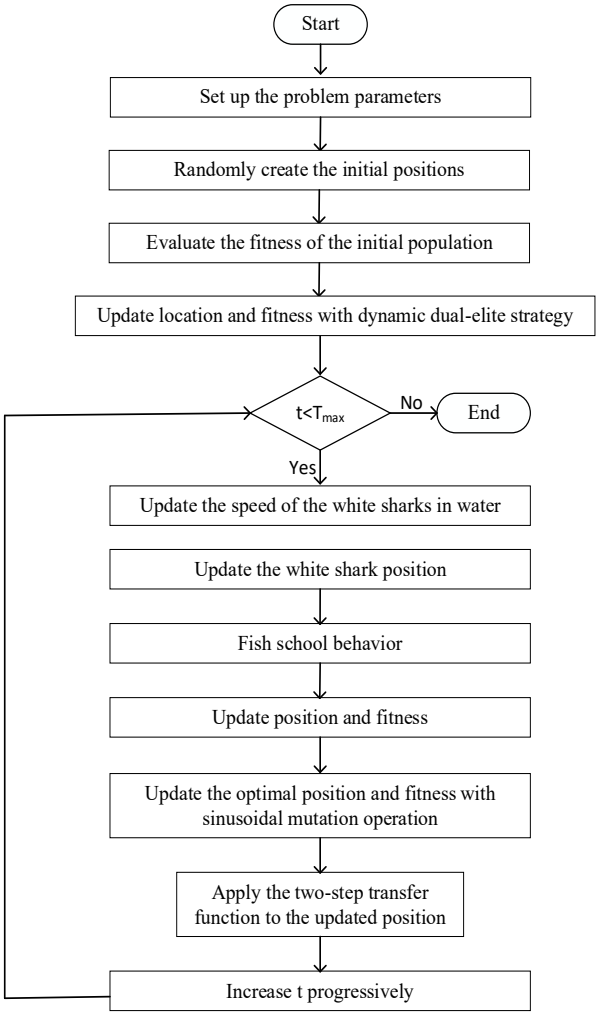


Fig. 1. The flowchart of the binary enhanced white shark optimizer algorithm.

IV. Experimental Results

This section outlines the experiments conducted and the results achieved to showcase the efficacy of the suggested algorithm when applied with feature selection in established classification tasks. We utilize the MATLAB 2022b platform to implement the proposed algorithm on nine high-dimensional datasets. Additionally, we analyzed and compared it with six other prominent meta-heuristic techniques commonly employed in feature selection.

A. Datasets

This article uses 9 high-dimensional datasets from the UCI Machine Learning Repository [31] and the Arizona State University (ASU) Feature Selection Repository [32]. Table II provides a comprehensive description of these datasets, including the feature dimensions, number of categories, sample sizes, and dataset sources.

TABLE II. Classified Datasets About UCI and ASU

Number	Dataset	Feature	Class	Instance	Source
1	Isolet	617	26	1560	ASU
2	lung	3312	5	203	UCI
3	COIL20	1024	20	1440	ASU
4	ALLAML	7129	2	72	ASU
5	darwin	450	2	174	UCI
6	Toxicity	1203	2	171	UCI
7	DrivFace	6400	3	606	UCI
8	USPS	256	10	2007	ASU
9	leukemia	7070	2	72	ASU

B. Compared Algorithms and Parameter Settings

To assess the proposed algorithm, the results are compared with six meta-heuristic optimization algorithms. The first is the original Binary White Shark Optimizer (BWSO) [33]. The other five commonly used methods include Binary Differential Evolution (BDE) [34], Binary Gravity Search Algorithm (BGSA) [35], Binary Atom Search Optimization (BASO) [36], Binary Grasshopper Optimization Algorithm (BGOA) [37], and Binary Particle Swarm Optimization Algorithm (BPSO) [38]. Table III shows the related parameter settings.

TABLE III. Compared Algorithm Parameter Settings

Algorithm	Parameter	Value
BEWSO	Population size	30
	f_{min}	0.07
	f_{max}	0.75
	p_{min}	0.5
	p_{max}	1.5
BDE	Crossrate	0.9
	N	10
BGSA	a	20
	G_0	100
BASO	Alpha	50
	Beta	0.2
	V_{max}	6
	N	10
BGOA	N	1
BPSO	V_{max}	6
	W_{max}	0.9
	W_{min}	0.4
	N	10
Universal	Maximum iterations	100
	Runs	30

C. Results and Discussion

Table IV presents the average fitness values achieved by the proposed algorithm and the other algorithms across nine datasets using 80% unlabeled data. The optimal and sub-optimal results for each dataset are indicated in bold and underlined, respectively. Notably, the average fitness score of the suggested approach is 36.52% lower than the sub-optimal value of BPSO. The proposed algorithm achieves the lowest average fitness value across all nine datasets. The BASO method ranks second, achieving four sub-optimal results, while both BWSO and BDE obtain two sub-optimal results each. Conversely, the BGSA method may perform poorly, with the highest overall average fitness value and no optimal or sub-optimal results across all datasets.

Using 60% unlabeled data, Table V displays the average fitness values obtained on the nine high-dimensional datasets. The overall average fitness value of the proposed algorithm is 35.29% lower than the sub-optimal value of BPSO, further showcasing its effectiveness. Similarly, the proposed algorithm achieves the lowest average fitness value across all datasets, with BPSO following closely behind and achieving four sub-optimal results, while BWSO secures three sub-optimal results.

TABLE IV. AVERAGE FITNESS VALUE UNDER DATA WITH 80% UNLABELED

Dataset	BEWSO	BWSO	BDE	BGSA	BASO	BGOA	BPSO
1	**0.1657**	0.2682	0.2140	0.2311	0.2080	0.2619	0.2150
2	**0.0282**	0.0565	0.0564	0.1016	0.0472	0.0792	0.0489
3	**0.0764**	0.0991	0.1321	0.1515	0.0987	0.1329	0.1169
4	**0.0391**	0.1111	0.0904	0.1677	0.1973	0.1535	0.1066
5	**0.1195**	0.1511	0.1697	0.2086	0.1927	0.2517	0.1548
6	**0.1362**	0.1655	0.2560	0.2603	0.2437	0.1695	0.1738
7	**0.0275**	0.0712	0.0717	0.0754	0.0598	0.0664	0.0771
8	**0.0756**	0.1064	0.0987	0.1184	0.1011	0.0912	0.1101
9	**0.0119**	0.0732	0.0395	0.0852	0.0671	0.0827	0.0691
Total average	**0.0756**	0.1225	0.1254	0.1555	0.1351	0.1432	0.1191

TABLE V. AVERAGE FITNESS VALUE UNDER DATA WITH 60% UNLABELED

Dataset	BEWSO	BWSO	BDE	BGSA	BASO	BGOA	BPSO
1	**0.1291**	0.1721	0.1683	0.2025	0.1577	0.1763	0.1570
2	**0.0172**	0.0571	0.0308	0.0875	0.0421	0.0546	0.0296
3	**0.0443**	0.0647	0.0577	0.0796	0.0622	0.0670	0.0545
4	**0.0275**	0.0761	0.0622	0.0757	0.0693	0.0957	0.0947
5	**0.0946**	0.1240	0.1552	0.1445	0.1556	0.1659	0.1463
6	**0.1099**	0.1508	0.1633	0.1985	0.2101	0.1423	0.1487
7	**0.0198**	0.0371	0.0470	0.0533	0.0505	0.0547	0.0593
8	**0.0517**	0.0983	0.0879	0.1053	0.0814	0.0692	0.0634
9	**0.0420**	0.0656	0.0749	0.0898	0.0707	0.0757	0.0755
Total average	**0.0596**	0.0940	0.0941	0.1152	0.1000	0.1002	0.0921

Table VI displays the average accuracy of the proposed method and other test methods using 80% unlabeled data. The results indicate the proposed algorithm performs well in accuracy and obtains the best overall average value, which is 5.03% higher than the sub-optimal value from BPSO, aligning with the best overall objective function value shown in Table IV. Specifically, it records the highest average classification accuracy when considering all datasets. The average accuracy of the lung, COIL20, ALLAML, DrivFace, USPS, and leukemia datasets reached more than 90%. In contrast, BWSO, BDE, and BASO also have better average accuracy. Among them, the BWSO algorithm and BASO algorithm get three sub-optimal results, respectively, and the BDE algorithm gets two sub-optimal results. However, the BGSA method performs poorly, exhibiting the lowest overall average accuracy and failing to achieve any optimal or sub-optimal results across all datasets.

Table VII outlines the average accuracy of the proposed approach and other tested methods utilizing unlabeled data. The proposed method outperforms the WSO method as well as all other competing techniques, achieving the best overall average value, which is 3.64% higher than the corresponding sub-optimal value for BDE. It also secures the highest average accuracy across all nine high-dimensional datasets, with over 90% accuracy on seven of them. In contrast, BWSO, BDE, and BPSO also have better average accuracy, with the BPSO algorithm achieving four sub-optimal results, while BDE and BWSO each attain three.

TABLE VI. AVERAGE ACCURACY UNDER DATA WITH 80% UNLABELED

Dataset	BEWSO	BWSO	BDE	BGSA	BASO	BGOA	BPSO
1	**0.8373**	0.7372	0.7917	0.7717	0.7949	0.7404	0.7885
2	**0.9750**	0.9500	0.9500	0.9025	0.9550	0.9250	0.9542
3	**0.9301**	0.9063	0.8750	0.8520	0.9028	0.8715	0.8828
4	**0.9643**	0.8929	0.9167	0.8357	0.8048	0.8500	0.8976
5	**0.8824**	0.8529	0.8363	0.7941	0.8078	0.7539	0.8461
6	**0.8676**	0.8382	0.7490	0.7422	0.7578	0.8333	0.8292
7	**0.9752**	0.9304	0.9339	0.9267	0.9422	0.9380	0.9280
8	**0.9302**	0.8953	0.9077	0.8855	0.9027	0.9127	0.8952
9	**0.9952**	0.9286	0.9643	0.9191	0.9381	0.9238	0.9357
Total average	**0.9286**	0.8813	0.8805	0.8477	0.8673	0.8610	0.8841

TABLE VII. AVERAGE ACCURACY UNDER DATA WITH 60% UNLABELED

Dataset	BEWSO	BWSO	BDE	BGSA	BASO	BGOA	BPSO
1	**0.8758**	0.8333	0.8381	0.8005	0.8430	0.8269	0.8430
2	**0.9875**	0.9500	0.9750	0.9167	0.9625	0.9500	0.9750
3	**0.9584**	0.9444	0.9479	0.9247	0.9444	0.9375	0.9479
4	**0.9762**	0.9286	0.9443	0.9286	0.9333	0.9076	0.9095
5	**0.9118**	0.8824	0.8510	0.8612	0.8451	0.8412	0.8560
6	**0.8935**	0.8529	0.8428	0.8052	0.7941	0.8611	0.8549
7	**0.9835**	0.9669	0.9587	0.9512	0.9527	0.9497	0.9452
8	**0.9535**	0.9052	0.9177	0.8988	0.9227	0.9352	0.9402
9	**0.9627**	0.9367	0.9286	0.9143	0.9331	0.9286	0.9287
Total average	**0.9448**	0.9112	0.9116	0.8890	0.9034	0.9042	0.9112

Overall, in high-dimensional datasets using both 80% and 60% unlabeled data, the improved algorithm consistently outperforms the original and other algorithms in classification accuracy. As the proportion of labeled data increases and the proportion of unlabeled data decreases, classification accuracy improves, possibly due to the enhanced correctness of labeled samples, leading to a more effective trained model.

From a deeper perspective, the proposed algorithm shows good performance regarding classification accuracy. By introducing a dynamic bi-elite strategy and sinusoidal mutation operation, the method can utilize the current optimal position information, which improves the ability to find potentially superior solutions and allows for more effective data dimensionality reduction, resulting in improved classification accuracy and efficiency.

Table VIII illustrates the mean number of features chosen by the proposed and other testing methods when utilizing 80% unlabeled data. In general, the proposed method performs admirably in feature selection, ranking third overall, following BWSO and BASO. Specifically, out of 9 datasets, the proposed method obtains 2 best results (Isolet, ALLAML) and 1 suboptimal result (lung); BWSO obtains 3 best results (DrivFace, USPS, leukemia). Additionally, BASO achieves 2 best results (lung, Toxicity) and four suboptimal results (COIL20, ALLAML, Darwin, DrivFace). BPSO obtains the best results in 2 cases (COIL20, Darwin). Combined with Table IV and Table VI, for these two datasets (Isolet, ALLAML), the suggested approach not only obtains the optimal number of feature selections, but also obtains the optimal average fitness and the optimal average accuracy.

Table IX illustrates the mean number of features chosen by the proposed methods and other testing methods when 60% unlabeled data is used. Overall, the suggested method ranks second in the average number of selected features, just behind BASO. Specifically, the suggested approach has the fewest average selected features across three datasets (lung, Toxicity, DrivFace), while BASO has 2 cases (ALLAML, darwin) and BPSO has 2 cases (Isolet, USPS) each. Additionally, BWSO has 1 optimal value (leukemia), and BGOA has 1 optimal value (COIL20) each.

The analysis shows that this algorithm selects fewer features than most algorithms, while maintaining better fitness values. However, despite having some advantages over other methods, it is difficult to reach the ideal number of selected features. This is primarily because the method emphasized in this study prioritizes feature selection to enhance classification accuracy, focusing on optimizing the fitness function with accuracy as the main objective. The proposed algorithm effectively reduces the number of features selected while retaining essential information.

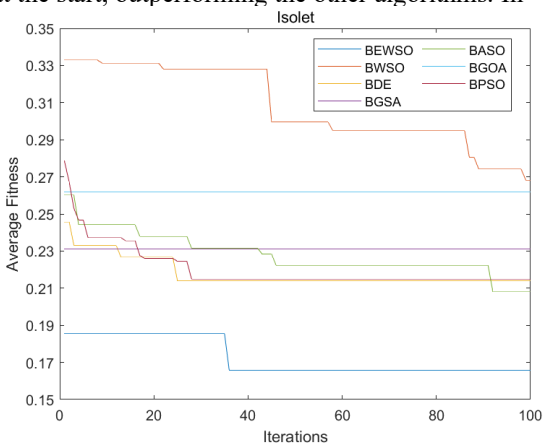
TABLE VIII. THE AVERAGE NUMBER OF FEATURES SELECTED UNDER DATA WITH 80% UNLABELED

Dataset	BEWSO	BWSO	BDE	BGSA	BASO	BGOA	BPSO
1	**285.50**	496.15	477.17	310.47	304.10	302.10	341.50
2	1129.12	2312.04	2278.03	1667.17	**801.08**	1640.14	1163.87
3	735.06	642.08	850.00	511.20	246.00	405.50	**89.10**
4	**2696.50**	3613.07	5652.77	3567.73	2841.83	3557.33	3734.90
5	134.33	249.00	344.43	216.00	112.57	362.07	**108.07**
6	627.50	645.10	905.90	601.93	**479.77**	537.57	567.30
7	1920.10	**1480.10**	4020.14	3198.70	1639.08	3209.50	2786.00
8	165.10	**69.17**	189.10	129.33	123.10	123.00	162.87
9	5100.53	**1780.10**	2952.13	3539.83	4100.17	5131.13	3854.07
Total average	1421.53	1254.09	1963.30	1526.93	**1183.08**	1696.48	1423.08

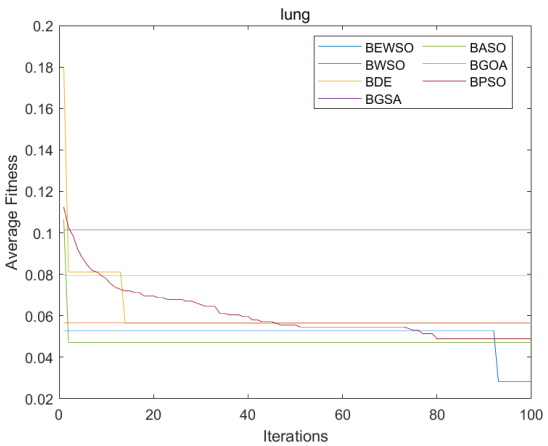
TABLE IX. THE AVERAGE NUMBER OF FEATURES SELECTED UNDER DATA WITH 60% UNLABELED

Dataset	BEWSO	BWSO	BDE	BGSA	BASO	BGOA	BPSO
1	378.10	436.05	494.27	306.90	138.10	306.10	**96.10**
2	**1595.04**	2519.10	2018.00	1658.70	1638.57	1673.05	1620.00
3	316.05	991.12	627.05	510.73	741.10	**52.03**	301.02
4	2778.67	3844.67	5007.53	3564.30	**2360.20**	3001.90	3641.13
5	324.10	339.12	345.33	320.12	**100.80**	390.03	169.60
6	**540.70**	626.20	926.85	678.10	751.00	578.20	601.37
7	**2186.11**	2826.02	3925.13	3201.57	2346.20	3100.60	3256.10
8	145.10	114.03	164.12	130.60	125.10	128.00	**106.02**
9	3500.34	**2013.34**	3012.23	3527.03	3156.23	3526.10	3450.24
Total average	1307.13	1523.29	1835.61	1544.23	**1261.92**	1417.33	1471.29

Fig. 2 illustrates the convergence patterns of seven optimization algorithms by analyzing the average fitness function values over 100 search iterations. The evaluation includes the suggested algorithm alongside six additional algorithms tested on nine datasets, each comprising 80% unlabeled data. The X-axis represents the number of iterations, and the Y-axis displays the average fitness value obtained. For the Isolet, COIL20, ALLAML, and USPS datasets, the proposed method shows good convergence in early iterations and continues to improve until the end. In the Toxicity and DrivFace datasets, the proposed method rapidly reaches the minimum fitness value at the start, outperforming the other algorithms. In the lung dataset, the proposed method initially lags behind BASO, but by the ninetieth iteration, it surpasses BPSO and maintains superior values. For the Darwin dataset, BWSO and BPSO are competitive algorithms. It can be seen that the proposed method starts strong, but BPSO excels after about 40 iterations, while BWSO performs best after around 50 iterations, and the proposed method again reaches the minimum value after about 70 iterations. In the leukemia dataset, the proposed method has the best initial convergence, which is surpassed by BDE in the middle, but the proposed method soon maintains the minimum fitness value again.



(a)



(b)

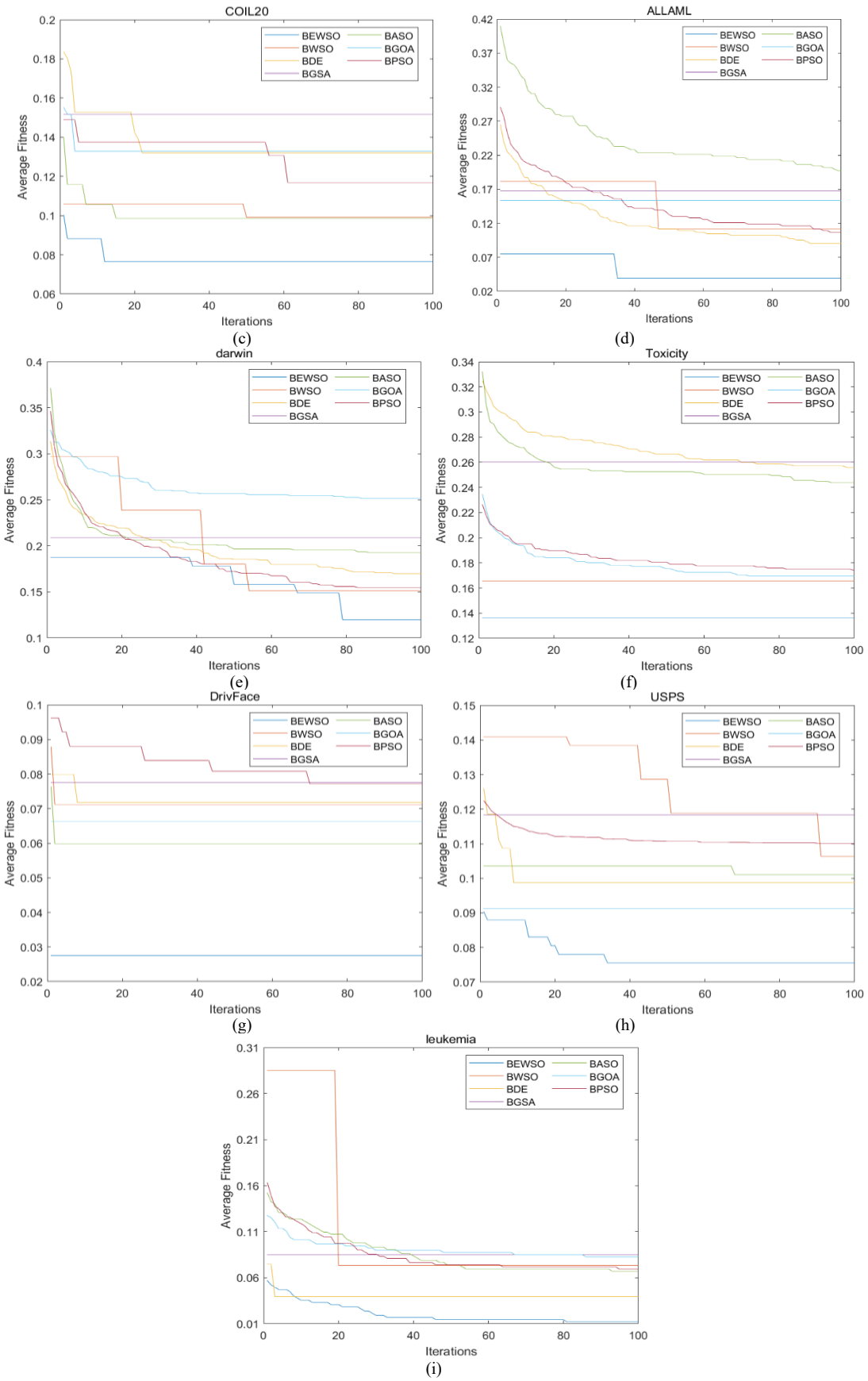
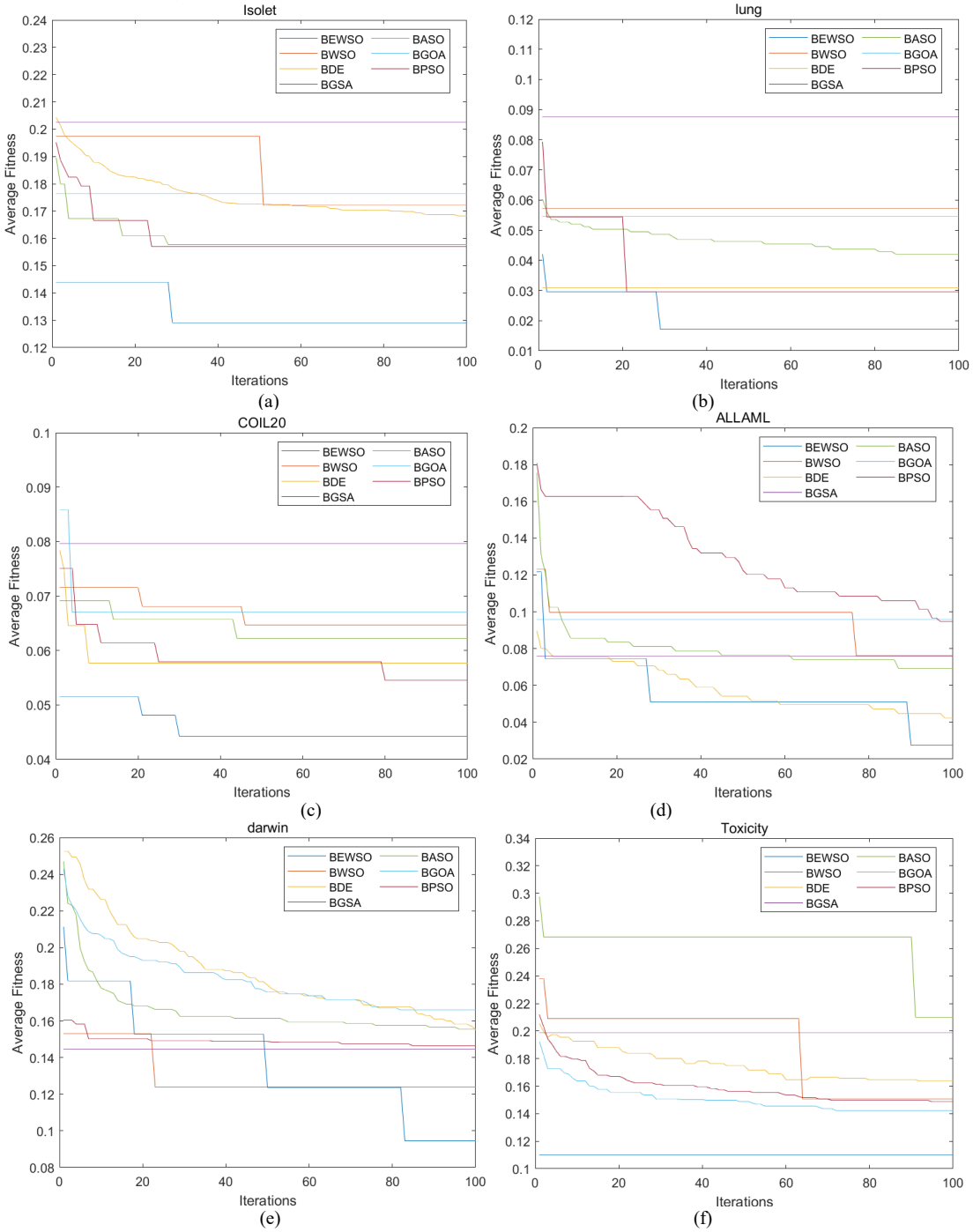


Fig. 2. The average convergence curve of the algorithms under data with 80% unlabeled.

Fig. 3 uses 9 datasets with 60% unlabeled data to describe the convergence behavior of the proposed method and the other six methods based on the mean value of the fitness function during 100 iterative searches. For most datasets, it is evident that the convergence curve becomes stagnant after more than 50 iterations. Examining each convergent graph individually, it reveals that the proposed method outperforms its competitors in the Isolet, lung, COIL20, DrivFace, and USPS datasets, showing good convergence in early iterations, with the average fitness value decreasing further with more iterations. In the Toxicity dataset, the suggested algorithm attains the best fitness value during the initial iteration, with BGOA following closely behind. In the ALLAML dataset, BGSA and BDE are the most competitive algorithms. BGSA gets the best fitness value in the first iteration, and as iterations progress, BDE and the proposed method alternately get the best fitness value until the proposed method finally gets the best value around the 90th iteration. In the Darwin dataset, BGSA starts with the best fitness value, BDE takes the lead around the 20th iteration, and both the proposed algorithm and BDE reach the same minimum fitness value around the 50th iteration, with the proposed method showing stronger convergence by the 80th iteration. For the leukemia dataset, the proposed method shows good convergence around the 30th iteration and maintains this performance until the end.



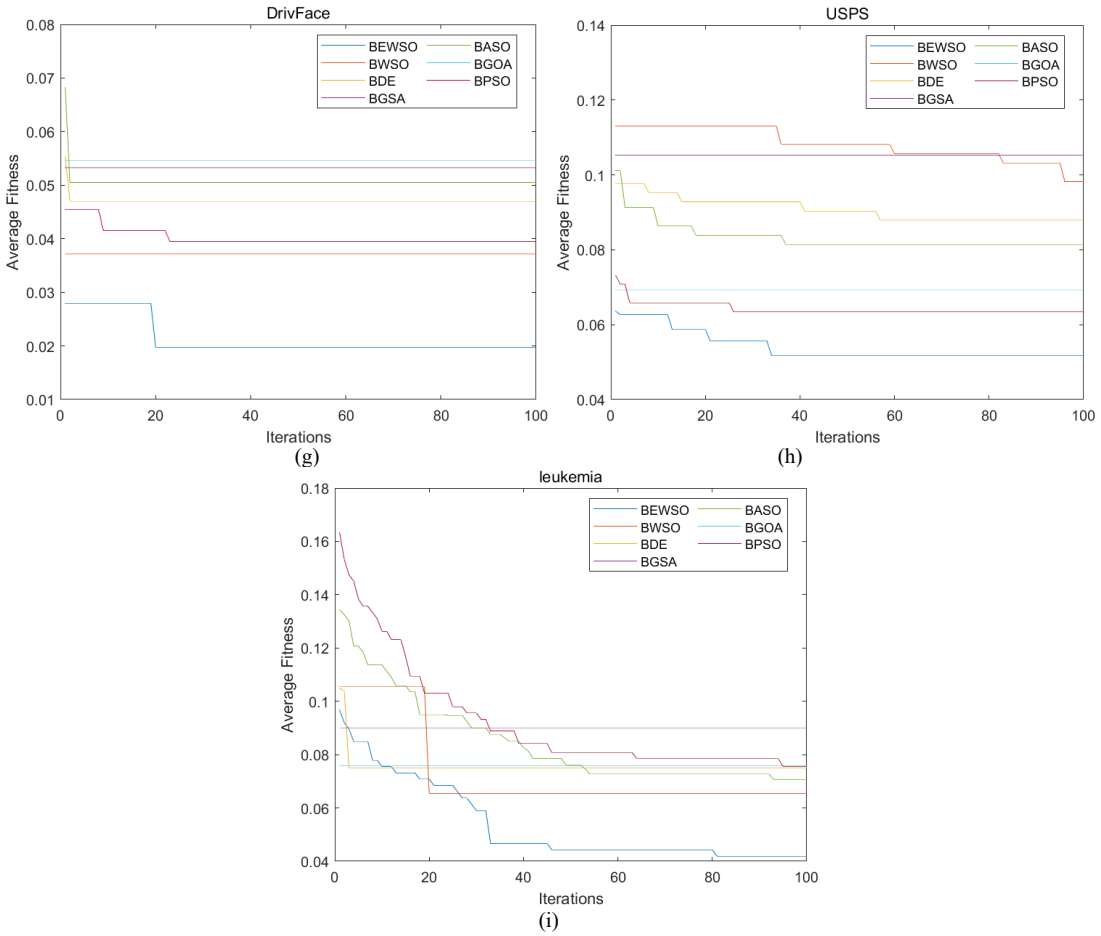


Fig. 3. The average convergence curve of the algorithms under data with 60% unlabeled.

From a practical perspective, the time taken for the algorithm to finish its calculations is crucial. The execution times for the suggested and other competing approaches have been measured and are presented in Table X. BGSA has the lowest overall average running time, with the proposed algorithm coming in second. Specifically, BGSA records the fastest execution time across seven datasets, while the proposed method obtains three sub-optimal results.

Furthermore, Table XI displays the computation time results for seven different algorithms using 60% unlabeled datasets. BGSA again has the shortest overall average running time, followed by the proposed algorithm. BGSA achieves seven optimal results and one suboptimal result, while the proposed method secures four suboptimal results.

TABLE X. THE AVERAGE TIME OF THE ALGORITHMS UNDER DATA WITH 80% UNLABELED

Dataset	BEWSO	BWSO	BDE	BGSA	BASO	BGOA	BPSO
1	40.70	45.79	73.63	21.04	63.00	**20.65**	74.50
2	36.49	37.86	53.39	**23.70**	50.05	53.74	27.32
3	48.15	39.00	68.96	**11.58**	53.47	69.33	51.00
4	28.00	**21.35**	26.91	21.60	25.60	23.00	29.00
5	13.00	19.00	14.00	**11.00**	15.52	24.55	14.84
6	19.16	17.20	15.77	**15.00**	**15.00**	16.00	**15.00**
7	63.00	64.00	75.00	**23.00**	91.09	70.00	85.00
8	32.00	38.00	65.00	**27.00**	56.80	60.00	67.00
9	38.00	37.60	50.45	**17.13**	38.23	50.00	34.00
Total average	35.39	35.53	49.23	**19.01**	45.42	43.03	44.18

TABLE XI. THE AVERAGE TIME OF THE ALGORITHMS UNDER DATA WITH 60% UNLABELED

Dataset	BEWSO	BWSO	BDE	BGSA	BASO	BGOA	BPSO
1	34.00	36.00	56.00	**13.31**	26.00	19.10	38.00
2	28.10	31.58	36.00	**13.50**	45.51	46.00	48.00
3	39.57	32.00	47.44	**12.67**	53.00	35.00	35.01
4	14.90	15.00	25.00	**10.63**	33.00	29.10	27.66
5	6.00	6.05	15.48	**5.00**	13.60	23.53	13.00
6	12.00	**11.24**	20.00	13.00	17.00	23.00	33.05
7	35.00	49.00	30.00	**13.67**	52.00	55.00	46.00
8	25.00	23.00	34.13	18.00	45.20	**17.00**	23.00
9	24.00	24.00	18.00	**13.00**	35.10	24.10	25.00
Total average	24.29	25.32	31.34	**12.53**	35.60	30.20	32.08

V. CONCLUSION

This study introduces an Enhanced White Shark Optimization algorithm with the dynamic dual-elite strategy and the sinusoidal mutation method, aimed at increasing the diversity of offspring and enhancing the potential optimal solutions. Furthermore, it introduces a binary transfer function and further investigates its binary application in semi-supervised feature selection. The proposed algorithm is evaluated on nine high-dimensional datasets and is compared against six other meta-heuristic methods for feature selection. The results indicate that when 20% and 40% of the training samples are randomly labeled, the proposed algorithm achieves the best overall mean accuracy; its lead rate is 5.03% and 3.64%, respectively, while also decreasing the number of features chosen. Specifically, with 20% labeled data, it surpasses other developed algorithms in average accuracy of 9 datasets and the number of selected features of 2 datasets. With 40% labeled data, it again outperforms all other developed algorithms in average accuracy of 9 datasets and number of selected features of 3 datasets. The experimental findings show the good performance of the Binary Enhanced White Shark Optimization method. This method could be applied to study high-dimensional gene selection in the future.

ACKNOWLEDGMENT

This work was supported by the National Natural Science Foundation of China (Grant No. 72403137) and the Natural Science Foundation of Shandong Province (Grant No. ZR2024QG024).

REFERENCES

[1] Wu, W. T., Li, Y. J., Feng, A. Z., Li, L., Huang, T., Xu, A. D., Lyu, J.. Data mining in clinical big data: the frequently used databases, steps, and methodological models. Military Medical Research, 2021, 8(1): 44.

[2] Cai, J., Luo, J. W., Wang, S. L., Yang, S.. Feature selection in machine learning: A new perspective. Neurocomputing, 2018, 300: 70-79.

[3] Camargo, G., Bugatti, P. H., Saito, P. T. M.. Active semi-supervised learning for biological data classification. Public Library of Science, 2020, 15(8): e0237428.

[4] Hodashinsky, I., Sarin, K., Shelupanov, A., Slezkin, A.. Feature Selection Based on Swallow Swarm Optimization for Fuzzy Classification. Symmetry-basel, 2019, 11(11): 1423.

[5] Braik, M., Hammouri, A., Atwan, J.. White Shark Optimizer: A novel bio-inspired meta-heuristic algorithm for global optimization problems. Knowledge-Based Systems, 2022, 243: 108457.

[6] Zhao, J. D., Lu, K., He, X. F.. Locality sensitive semi-supervised feature selection. Neurocomputing, 2008, 71(10-12): 1842-1849.

[7] Cheng, H. R., Deng, W., Fu, C., Wang, Y., Qin, Z. G.. Graph-based semi-supervised feature selection with application to automatic spam image identification. International Workshop on Computer Science for Environmental Engineering and EcoInformatics. 2011, 259-264.

[8] Doquire, G., Verleysen, M.. A graph Laplacian based approach to semi-supervised feature selection for regression problems. Neurocomputing, 2013, 121: 5-13.

[9] Benabdeslem, K., Hindawi, M.. Constrained Laplacian score for semi-supervised feature selection. Joint European Conference on Machine Learning and Knowledge Discovery, 2011: 204-218.

[10] Kalakech, M., Biela, P., Macaire, L., Hamad, D.. Constraint scores for semi-supervised feature selection: a comparative study. Pattern Recognition Letters, 2011, 32(5): 656-665.

[11] Sheikhpour, R., Sarram, M. A., Gharaghani, S.. A survey on semi-supervised feature selection methods. Pattern Recognition, 2017, 64: 141-158.

[12] Sunzhong, L. V., Jiang, H., Zhao, L.. Manifold based fisher method for semi-supervised feature selection. International Conference on Fuzzy Systems and Knowledge Discovery, 2013, 664-668.

[13] Han, Y. H., Yang, Y., Yan, Y.. Semisupervised feature selection via spline regression for video semantic recognition. IEEE Transactions on Neural Networks and Learning Systems, 2015, 26(2): 252-264.

[14] Zeng, Z. Q., Wang, X. D., Zhang, J., Wu, Q.. Semi-supervised feature selection based on local discriminative information. Neurocomputing, 2016, 173: 102-109.

[15] Ren, J., Qiu, Z., Fan, W.. Forward semi-supervised feature selection. Conference on knowledge discovery and data mining, 2008: 970-976.

[16] Song, X. F., Zhang, Y., Guo, Y. N., Sun, X. Y., Wang, Y. L.. Variable-size cooperative coevolutionary particle swarm optimization for feature selection on high-dimensional data. IEEE Transactions on Evolutionary Computation, 2020, 24(5): 882-895.

[17] Bellal, F., Elghazel, H., Aussem, A.. A semi-supervised feature ranking method with ensemble learning. Pattern Recognition Letters, 2012, 33(10): 1426-1433.

[18] Barkia, H., Elghazel, H., Aussem A.. Semi-supervised feature importance evaluation with ensemble learning. IEEE International Conference on Data Mining, 2011:31-40.

[19] Han, Y. K., Park, K. S., Lee, Y. K.. Confident wrapper-type semi-supervised feature selection using an ensemble classifier. International Conference on Artificial Intelligence Management Science and Electronic Commerce, 2011: 4581-4586.

[20] Ma, Z. G., Nie, F. P., Yang, Y.. Discriminating joint feature analysis for multimedia data understanding. IEEE Transactions on Multimedia, 2012, 14(6): 1662-1672.

[21] Shi, C. J., Ruan, Q. Q., An, G.. Sparse feature selection based on graph Laplacian for web image annotation. Image and Vision Computing, 2014, 32(3): 189-201.

[22] Xu, Z. L., King, I., Lyu, M. R. T., Jin, R.. Discriminative semi-supervised feature selection via manifold regularization. IEEE Transactions on Neural networks, 2010, 21(7): 1033-1047.

[23] Feofanov, V., Devijver, E., Amini, M. R.. Semi-supervised wrapper feature selection by modeling imperfect labels. arXiv, 2019.

[24] Lin, Y. J., Hu, Q., Zhang, J., Wu, X. D.. Multi-label feature selection with streaming labels. Information Sciences, 2016, 372: 256-275.

[25] Peng, Y., Chen, G., Xu, M., Wang, C. J., Xie, J. Y.. Multi-label learning by exploiting label correlations with LDA. International Conference on Tools with Artificial Intelligence, 2017: 168-174.

[26] Chen, L., Ma, L. Y., Li, L.. Enhancing sine cosine algorithm based on social learning and elite opposition-based learning. Computing, 2024, 106(5): 1475-1517.

[27] Amer, D. A., Attiya, G., Zeidan, I., Nasr, A. A.. Elite learning Harris Hawks Optimizer for multi-objective task scheduling in cloud computing. Journal of Supercomputing, 2022, 78(2): 2793-2818.

[28] Draa, A., Chettah, K., Talbi, H.. A Compound Sinusoidal Differential Evolution algorithm for continuous optimization Check. Swarm and Evolutionary Computation, 2019, 50, 100450.

[29] Khalilpourazari, S., Pasandideh, S. H. R.. Sine-cosine crow search algorithm: theory and applications. Neural Computing and Applications, 2020, 32(12): 7725-7742.

[30] Agrawal, P., Ganesh, T., Oliva, D., Mohamed, A. W.. S-shaped and V-shaped gaining-sharing knowledge-based algorithm for feature selection. Journal information, 2022, 52(1): 81-112.

[31] Dheeru, D., Karra, T. E.. UCI Machine Learning Repository, University of California, Irvine, School of Information and Computer Sciences, 2007.

[32] Tawhid, M. A., Ibrahim, A. M.. Feature selection based on rough set approach, wrapper approach, and binary whale optimization algorithm. International Journal of Machine Learning and Cybernetics, 2020,11(3): 573–602.

[33] Hammouri, A. I., Braik, M. S., Al-hiary, H. H., Abdeen, R. A.. A binary hybrid sine cosine white shark optimizer for feature selection. Cluster Computing-The Journal of Networks Software Tools and Applications, 2024, 27(6): 7825-7867.

[34] Pampara, G., Engelbrecht, A. P., Franken, N.. Binary differential evolution. IEEE International Conference on Evolutionary Computation, 2006, 873–1879.

[35] Rashedi, E., Nezamabadi-pour, H., Saryazdi, S.. BGSA: binary gravitational search algorithm. Natural Computing 2010, 9(3): 727-745.

[36] Too, J., Abdullah, A. R.. Binary atom search optimization approaches for feature selection. Connection Science, 2020, 32(4): 406-430.

[37] Mafarja, M., Aljarah, I., Faris, H., Hammouri, A. I., Mirjalili, S.. Binary grasshopper optimisation algorithm approaches for feature selection problems. Expert Systems with Applications, 2018, 117: 267-286.

[38] Chuang, L. Y., Chang, H. W., Tu, C. J., Yang, C. H.. Improved binary PSO for feature selection using gene expression data. Computational Biology and Chemistry, 2008, 32(1): 29–38.