

RESEARCH ARTICLE

A Hybrid Feature Selection Method Using an Improved Binary Butterfly Optimization Algorithm and Adaptive β –Hill Climbing

ANURAG TIWARI 

Thapar Institute of Engineering and Technology, Patiala, Punjab 147004, India

e-mail: anuragtiwari.rs.cse17@itbhu.ac.in


This work was supported by the Department of Computer Science and Engineering, Thapar Institute of Engineering and Technology, Patiala, India.

ABSTRACT The Butterfly Optimization Algorithm (BOA) is a recently proposed nature-inspired meta-heuristic algorithm mimicking the food-foraging behavior of butterflies. Its abilities include simplicity, good convergence rate towards local optima, and avoiding the local optima stagnation problem to some extent. In earlier studies, the performance of Binary BOA (BBOA) is shown to be superior to various state-of-the-art methods in different optimization issues, such as search space reduction and solving classical engineering problems. Here, BBOA expands the original search space with all possibilities (Exploration) and seeks to determine the best one from all the produced solutions (Exploitation). Generally, the global performance of BBOA depends on the tradeoff between the Exploration and Exploitation phase and hence, produces quality solutions when a suitable tradeoff is maintained. This study introduces an improved and computationally effective variant of conventional BBOA by improving the local search ability of the Butterfly Optimization Algorithm. Initially, twelve binary variants were produced using three different transfer functions (S, U, V-shaped), and solution quality is evaluated in terms of respective fitness function scores. Next, we explored the local search ability of BOA by another recently developed optimization technique, namely, Adaptive β –Hill Climbing, to compute quality solutions. This optimization process employed two stochastic operators: N -operator (Neighborhood operator) and β -operator (Mutation operator) to generate improved offspring compared to parent solutions. This phase is iteratively implemented until the desired level of binary pattern with suitable classification accuracy is obtained. We validated the proposed approach on twenty datasets with eleven state-of-the-art feature selection algorithms. The overall results suggest that the proposed improvements increase the classification accuracy with fewer features on most datasets. In addition, the proposed approach's time complexity was significantly reduced on eighteenth out of twenty datasets. Moreover, the proposed method effectively balances space exploration and solution exploitation in feature selection problems.

INDEX TERMS Butterfly optimization algorithm, classification accuracy, convergence rate, feature selection, local optima, transfer function.

I. INTRODUCTION

With the rapid growth of science and technology, a huge amount of data is produced in different data mining applications, such as telecommunication, the banking sector, and biological data analysis. Most often, this data is inbuilt with

The associate editor coordinating the review of this manuscript and approving it for publication was Donato Impedovo .

abundant noisy, irrelevant, and redundant features or dimensions that reduce the performance of applied data analysis techniques in terms of inaccurate results and high computational complexity. Moreover, the conventional machine learning approaches fail to deal with high dimensionality because when the dimensionality increases, the space volume increases so fast that the available data becomes sparse [1]. Therefore, it is required to determine the limited but

informative set of features that can improve the overall performance of the machine learning model with relatively lower complexity and higher classification accuracy.

In many research areas, feature selection methods are vital in improving classification accuracy with minimum training time by selecting only a significant set of features from the original set. Based on the information processing standards, feature selection approaches can be broadly categorized into three classes: (1) Filter, (2) Wrapper, and (3) Hybrid methods [2]. Filter methods employ various statistics and information-theoretic concepts such as mutual information, variance, information gain, and correlation to determine the relevance of features. These methods are computationally fast but limited by poor classification accuracy. In addition, filter methods ignore the relevance of the dimensions as a set while selecting a new feature. On the other hand, wrapper methods use a classification approach to evaluate the significance of the selected feature subset. The performance of these methods depends on the discrimination criteria and working of the applied classifier, making them slow and computationally expensive but more effective than filter approaches.

Ultimately, hybrid methods enjoy the merits of both filter and wrapper classes in two ways: (1) they involve the interaction between selected features and classifier, and (2) they are capable of determining dependencies with a lower computational cost than the wrapper methods since they do not require to evaluate the optimal feature set iteratively. However, finding two or more mutually compatible and effective optimization schemes for hybridization is still a major challenge.

Recently, multiple researchers use different approximation algorithms to seek only optimal solutions without evaluating the remaining alternatives. Metaheuristic techniques such as Genetic Algorithm (GA) [3], Particle Swarm Optimization (PSO) [4], Ant Colony Optimization (ACO) [5], Monarch Butterfly Optimization (MBO) [6], and Grey Wolf Optimizer (GWO) [7] have been already introduced in wrapper methods to maximize classification accuracy with the minimum number of features. In addition, these algorithms provide classification accuracy bounds by computing minimum and maximum performances of all the feasible solutions. However, these algorithms suffer from three key problems: (1) Computationally expensive because of the involvement of classifier and multiple optimization operators, (2) Poor solution quality because of an inappropriate tradeoff between the exploration and exploitation phase, and (3) Weak convergence rate. These limitations affect the applied classification approach's overall performance and minimize the respective algorithm's scope in similar domains.

In recent studies, various new and improved versions of metaheuristics have been demonstrated to solve feature selection problems with a high feature reduction rate. For example, Zhang et al. [8] designed a two-archive-guided multiobjective Artificial Bee Colony (ABC) to improve the original

ABC algorithm's convergence rate and exploration abilities. Here, two new operators: (1) Convergence-guided search for employed bees and (2) Diversity-guided search operators, were used to finding a set of non-dominating feature subsets. In addition, two archives, *i.e.*, the leader and the external archive, are employed to enhance the search ability of different kinds of bees. The improvements obtained better classification accuracy with a higher convergence rate and fewer attributes. Al-Betar et al. [9] amalgamated a local search-based β -Hill climbing optimizer with an S-shaped transfer function to compute pleasing solutions in the feature selection problem. The performance of the proposed optimization algorithm was compared with three local search methods and ten metaheuristic algorithms. The obtained results indicate that the developed binary optimizer outperforms other comparative local search methods in terms of classification accuracy on 16 out of 22 datasets.

Dhiman et al. [10] developed Binary variants of Emperor Penguin Optimizer (BEPO) using S- and V-shaped transfer functions to feature selection problems. They compared their results with Binary Spotted Hyena Optimizer (BSHO) [11], Binary Whale Optimizer (BWO) [12], Binary Dragonfly Optimizer (BDO) [13], Binary Bat Algorithm (BBA) [14], Binary Grey Wolf Optimizer (BGWO) [15], Binary Particle Swarm Optimizer (BPSO) [16], and Binary Gravitational Search Algorithm (BGSA) [17] on twelve benchmark datasets. The overall results concluded the superiority of the proposed BEPO over other competitive algorithms.

Recently, a new metaheuristic algorithm, the Butterfly Optimization Algorithm (BOA), mimics butterflies' food search and mating behavior to solve global optimization problems [18]. This optimizer is mainly based on the foraging strategy of butterflies, which utilize their sense of smell to determine the location of their mating partner. The performance of the BOA is shown to be superior compared to Artificial Bee Colony (ABC) [19], Cuckoo Search (CS) [20], Differential Evolution (DE) [21], Firefly Algorithm (FA) [22], Genetic Algorithm (GA) [3], Particle Swarm Optimization (PSO) [4] and Monarch Butterfly Optimization (MBO) [6] for various engineering problems. Despite various advantages, BOA also suffers from drawbacks such as diminished population diversity and the tendency to get trapped in a local optimum. However, similar to other metaheuristics, conventional BOA was limited only to continuous search space-based problems. Earlier, various Binary variants of BOA (BBOA) have been developed to solve discrete real-time problems such as feature selection [50] and multidimensional knapsack [51]. In such problems, two transfer functions: (1) Sigmoid and (2) V-shaped were used to convert continuous search space into a binary one. Moreover, the selection process of the transfer functions was random, and therefore no correlation could be developed between the performance of both methods. These works mainly focused on instance state transformation from continuous to binary search space and thus lacked providing high-quality solutions during problem

optimization. These limitations inspired us to develop a novel hybrid variant of BOA that maximizes the metaheuristic's overall performance in terms of higher solution quality, better convergence rate, local optima avoidance, and reduced computational and temporal complexity.

The proposed work amalgamates the local and global search strategy of the BOA with a recently introduced, Adaptive β -Hill Climbing ($A - \beta HC$) [23]. The $A - \beta HC$ optimizer is an improved version of the old βHC that iteratively produces an improved solution based on two operators: (1) N -operator (Neighborhood operator) and β -operator (mutation operator). Further, three transfer functions (S-, V-, and U-shape) [24] are used to create twelve binary variants of BOA, and the proposed improvement was merged to obtain optimal solutions. We validated our proposed feature selection method on twenty publicly open datasets and three classifiers: (1) Support Vector Machine (SVM), (2) Naïve Bayes (NB) algorithm, and (3) Decision Tree (DT). Also, a five-fold cross-validation approach [25] is used to quantify classification results statistically to split the global population into training and evaluation sets.

The main objectives of the proposed research study can be recapitulated as follows:

- I. Understanding the effect of BOA's local/global search ability on the exploration-exploitation tradeoff in feature selection.
- II. We introduced three binary variants of conventional BOA using transfer functions (S-, V-, and U-shape) and evaluated the relationship between a set of three and the $A - \beta HC$ optimizer.
- III. Proposing a novel approach to improve the performance of conventional b-BOA while boosting offspring quality in optimal feature subset selection.
- IV. Estimating the compatibility of the proposed technique and three different classification schemes.
- V. Comparing the performance of our feature selection and classification model with newly published state-of-the-art algorithms.

The remaining structure of the paper is as follows: Section II describes the working of conventional BOA, transfer functions, and $A - \beta HC$ technique. In Section III, the proposed improvement strategy is explained. The experimental results and discussion are provided in Section IV. Finally, conclusions and the future scope of the work are presented in Section V.

II. PRELIMINARIES

This section describes three important concepts used in our study, namely, Butterfly Optimization Algorithm (BOA), transfer functions, and $A - \beta HC$ optimizer scheme. In the first subsection, the motivation and update strategy of BOA is discussed. Similarly, three mathematical functions for transforming a continuous problem into a binary one are covered in the second subsection. The third subsection describes the mathematical representation and working of the Adaptive β -Hill Climbing scheme.

A. BUTTERFLY OPTIMIZATION ALGORITHM

Conventional Butterfly Optimization Algorithm (BOA) is a recently proposed nature-inspired metaheuristic scheme that mimics butterflies' food foraging and mating behavior. In BOA, the functioning of butterflies can be described as follows:

- 1) Each butterfly emits some fragrance to attract other butterflies towards each other.
- 2) Butterflies forage randomly or move towards the butterfly with the most fragrance value.
- 3) The entire concept of sensing and food search processing depends on the produced fragrance (f) and three essential factors, namely, sensory modality (c), stimulus intensity (I), and power exponent (a). The relationship among these factors is given in Equation 1.

$$f = c * I^a \quad (1)$$

The values of constants c and a lie between 0 and 1. For $a=1$, the maximum fitness value or stimulus intensity is obtained, while $a=0$ shows the minimum fragrance value that any butterfly cannot sense. Therefore, parameter a controls the nature of the butterfly optimization algorithm. Another parameter is c which determines the convergence speed of the algorithm.

In BOA, two phases, global and local search schemes, are used to determine the food location. During the global search, the movement toward the best butterfly position (g^*) is based on the fitness value of the objective function score computed by the butterfly according to Equation (2)

$$x_i^{t+1} = x_i^t + (r^2 \times g^* - x_i^t) \times f_i \quad (2)$$

where x_i^t denotes the position vector of i^{th} butterfly at the time t , r is a random number distributed in the range [1,0] and f_i is the fragrance emitted by i^{th} butterfly. In the second scheme, local search is defined as:

$$x_i^{t+1} = x_i^t + (r^2 \times x_j^t - x_k^t) \times f_i \quad (3)$$

where x_j^t and x_k^t are instances of j^{th} and k^{th} butterflies from the same solution space created by g^* . In this study, x_j^t and x_k^t instances are the second and third nearest best solutions to g^* . The selection strategy of global or local search policy depends on the quality of the obtained best solution. This process continues until the algorithm achieves maximum performance or matches the stopping criteria. The pseudocode of conventional BOA is given in Algorithm 1.

B. TRANSFER FUNCTION

In general, conventional metaheuristics are developed only for continuous optimization problems. In order to explore the scope of metaheuristics in a discrete domain, a set of mathematical formulas is used. In machine learning, these formulas are termed transfer functions and provide the probability that the problem instance can take any discrete value from a given range. Since feature selection is a binary optimization

Algorithm 1 General Pseudocode of BOA

- Initialize n butterflies population positions x_i ($i = 1, 2, \dots, n$)
 - Set the initial value of parameters (switching probability ρ , sensory modality c , power exponent a , and the number of iterations N)
1. **while** not reach N **do**
 2. **for** each butterfly bf in the population **do**
 3. Compute the fragrance value f for each bf using Eq. 1
 4. **end for**
 5. Find the best butterfly bf
 6. Assign the best butterfly to g^*
 7. **for** each butterfly bf in the population **do**
 8. Generate a random value r over the interval $[1,0]$
 9. **if** ($r < \rho$)
 10. Update bf position by using Eq. 2 (Exploration phase)
 11. **else**
 12. Update bf position by using Eq. 3 (Exploitation phase)
 13. **end if** Evaluate the new butterfly If the new butterfly is better, update it in the population
 14. **end for**
 15. Update the value of the power exponent, and variable c
 16. Update the best global solution if find the better solution
 17. **end while**
 18. Return the best solution found by the BOA

problem, transfer functions can be used to identify the most dominating feature subset from the original ones. In the BOA, transfer functions compute the probability of changing the position vector from 0 to 1 and vice versa at a given instance. In our work, three functions convert continuous search space into a binary, namely Sigmoid, V-shape, and U-shape transfer functions. The detailed working of all three transfer functions is discussed in the following subsections:

1) SIGMOID VERSION OF BUTTERFLY OPTIMIZATION ALGORITHM

As discussed above, the butterflies' new position after a global or local search provides continuous search space. Therefore, this space must be transformed into corresponding binary ones. This transformation is performed using a spiral-shaped mechanism provided by the sigmoid or S-shape function defined in Equation 4.

$$S(F_i^k(t)) = \frac{1}{1 + e^{-F_i^k(t)}} \quad (4)$$

where $F_i^k(t)$ is the continuous value of the fragrance of the i^{th} butterfly in the k^{th} direction at instance t . The output of the sigmoid function is still a continuous value; therefore, a threshold is fixed to compute the binary value

corresponding to $S(F_i^k(t))$. The S-shape transfer function transforms an infinite input space into finite output. It should be known that the probability of changing the position vectors increases as the slope of the sigmoid function curve rises. The key point about the S-shaped transfer function is its ability to restrict butterfly movement within the range of $[1,0]$, which makes it simple to implement and easy to transform the infinite continuous positions into respective binary ones. The common stochastic threshold used to obtain a binary solution using the S-shape transfer function is given in Equation 5. The graphical representation of the S-shaped transfer function with four different slopes and output variation is given in Fig. 1. (A).

$$F_i^k(t+1) = \begin{cases} 0 & \text{if } rand < S(F_i^k(t)) \\ 1 & \text{otherwise} \end{cases} \quad (5)$$

2) V-SHAPED VARIANT OF BUTTERFLY OPTIMIZATION ALGORITHM

V-shape transfer functions are alternatives to Sigmoid functions and require different rules for updating the positions. For these transfer functions, the following rules are used to update the positions of butterflies in the continuous search space:

$$V(F_i^k(t+1)) = \begin{cases} (f_i^k(t))^{-1} & \text{if } rand < V(F_i^k(t)) \\ f_i^k(t) & \text{otherwise} \end{cases} \quad (6)$$

where $(f_i^k(t))^{-1}$ indicates the complement of $f_i^k(t)$ and the remaining symbols are as same as defined in the previous subsection. The benefit of the V-shaped transfer function over the S-shaped is that these functions avoid locating the butterfly position within the range of $[1,0]$. In other words, they encourage butterflies to roam freely according to the update rule (Equation 6) without restricting their movements. Based on their shape, these rules are named V-shaped transfer functions, and the group is termed the "V-shaped" family of transfer functions. The graphical representation of the V-shaped transfer function with four different slopes and output variation is given in Fig. 1. (B).

3) QUADRATIC TRANSFER FUNCTION

Compared to S- and V-shaped transfer functions, the Quadratic transfer function (Q-shaped) is relatively new and has been successfully implemented to increase the exploration ability of Particle Swarm Optimization (PSO) and Equilibrium Optimizer [26]. Because of its shape, it is also termed a U-shaped function. Similar to the S-shaped, the U-shaped transfer function ensures that computed output will remain in the specific range during the execution of the binarization process. The mathematical formula of the U-shaped

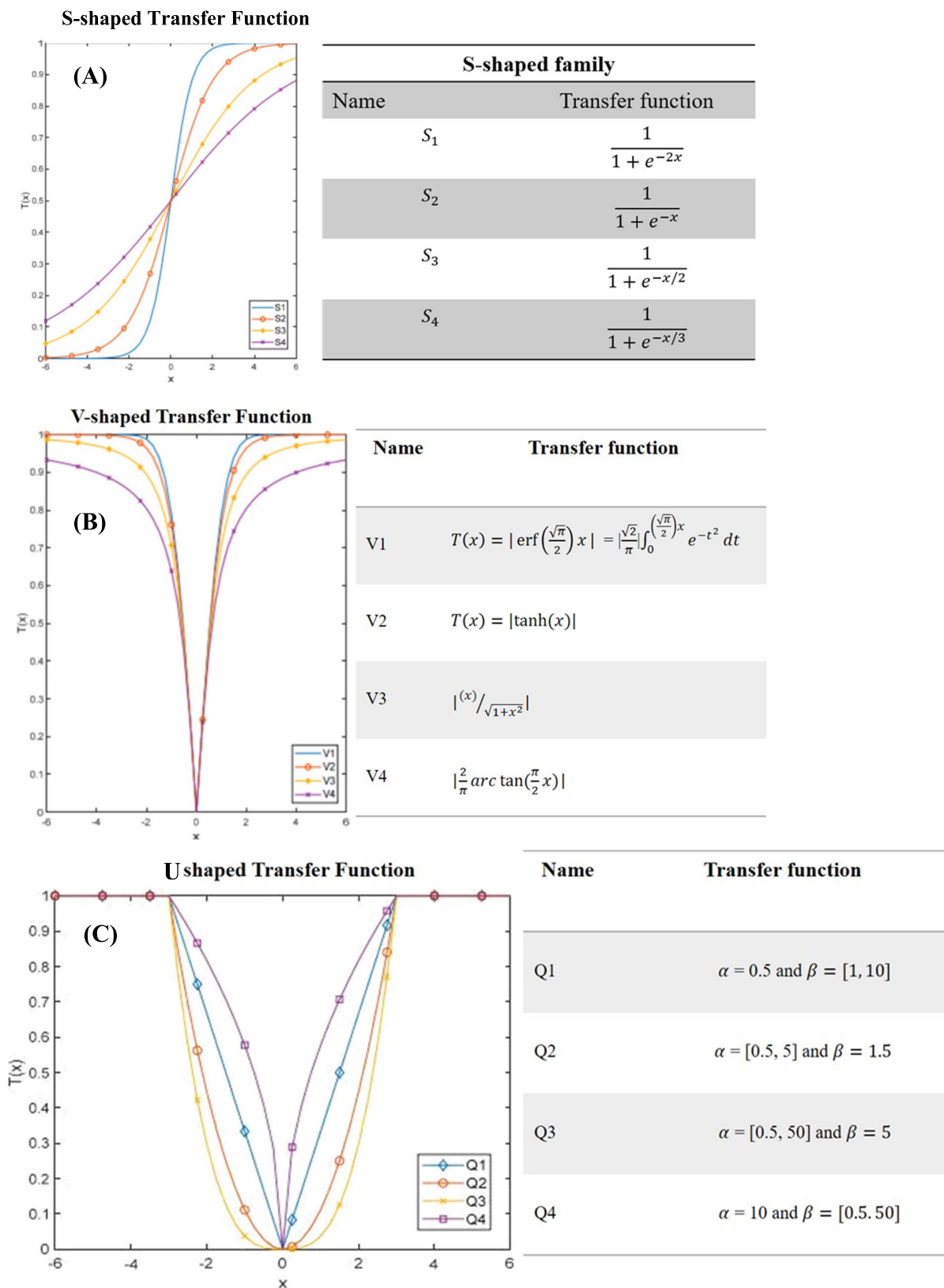


FIGURE 1. The pictorial representation of (A) The S-shaped, (B) V-shaped, and (C) U-shaped transfer function with four different slopes and respective Variations in output.

transfer function is given in Equation 7.

$$U(x) = \alpha \cdot |x|^\beta \tag{7}$$

$$T(U_x(t+1)) = \begin{cases} 1 & \text{if } U(x) \geq rand \\ 0 & \text{otherwise} \end{cases} \tag{8}$$

where α and θ are two control parameters that define the slope and width of the U-shaped transfer function. The range of this function always lies in [1,0]. If the produced transfer function output is greater than or equal to generated random number, then the binary state will be 1 (accepted); otherwise, it will be 0 (rejected). Here, the accepted and rejected terms represent the inclusion and exclusion of features in a given iteration while reducing irrelevant and redundant dimensions. It should be noted that the performance of the U-shape transfer function is not used to monitor the performance of the binary butterfly algorithm to date. The graphical representation of the U-shaped transfer function with four different slopes and output variation is given in Fig. 1. (C).

C. ADAPTIVE β -HILL CLIMBING SCHEME

In metaheuristics applications, it is crucial to improve solution quality without altering the exploration and exploitation tradeoff. Adaptive β -Hill Climbing ($A\beta HC$) is a recently proposed optimization scheme, an adaptive version of the conventional βHC and HC algorithms. Hill Climbing is a simple local search method that seeks a better solution than the previous one. But it often gets stuck in local optima. To resolve this limitation, βHC iteratively produces improved solutions using two operators: (1) N - operator (Neighborhood operator) and (2) β -operator. The N - operator randomly selects a neighborhood solution using Eq. 9.

$$r'_i = r_i \pm U(0, 1) * N \exists i \in [1, D] \tag{9}$$

where i is randomly selected in the range [1, D], where D refers to the dimensions of the problem and N denotes the highest possible distance between the current solution and its neighbor solution. β -operator is motivated by the uniform mutation operator of the Genetic Algorithm. Here, the new solution is assigned values either from the current solution or randomly from the corresponding range with a probability value $\beta \in [0, 1]$.

$$r''_i = \begin{cases} r_r \text{ if } rnd \leq \beta \\ r'_i \text{ else} \end{cases} \tag{10}$$

where rnd is a random number generated in the range [1,0] and r_r is another random number within the range of that particular dimension of the problem in consideration. It is clear that the final output of this optimization process mainly depends on mentioned parameters β and N . The optimal value of both parameters requires exhaustive experiments, which are computationally expensive and time-consuming. To resolve this overhead, $A\beta HC$ expresses β and N as a function of iteration number and given by

$$N(t) = 1 - \frac{t^{\frac{1}{k}}}{(\text{total number of iteration})^{\frac{1}{k}}} \tag{11}$$

where k is a constant and $N(t)$ denotes the value of N at iteration t . Similarly, parameter β is computed in terms of a specific range $[\beta_{min}, \beta_{max}]$ using Eq. 12.

$$\beta(t) = \beta_{min} + (\beta_{max} - \beta_{min}) * \frac{t}{\text{total number of iterations}} \tag{12}$$

where $\beta(t)$ denotes the value of β at iteration t . Now, if the new solution r''_i (Equation 10) shows better solution quality in terms of minimum fitness score (Equation 13), it replaces the older one r'_i ((Equation 9) otherwise, no change occurs. The pseudocode of the $A\beta HC$ optimization is given in Algorithm 2.

Algorithm 2 General Pseudocode of Adaptive β -Hill Climbing algorithm

- 1: Initialize β_{min} , β_{max} , and K
- 2: $x_i = LB_i + (UB_i - LB_i) \times U(0, 1), \forall i = 1, 2, \dots, N$
- 3: Calculate $f(x)$
- 4: $t = 0$
- 5: **while** ($t \leq \text{Max}_t$) **do**
- 6: $x' = x$
- 7: $C_t = \frac{1}{t^{\frac{1}{k}}}$
- 8: $N_t = 1 - C_t^{\frac{\text{Max}_t}{1 - C_t^{\frac{1}{k}}}}$ { Adaptive N }
- 9: $RndIndex \in (1, N)$
- 10: $x'_{RndIndex} = x'_{RndIndex} \pm N_t$
- 11: $x'' = x'$
- 12: $\beta_t = \beta_{min} + t \times \frac{\beta_{max} - \beta_{min}}{\text{Max}_t - t}$ { Adaptive β }
- 13: **for** $i = 1, \dots, N$ **do**
- 14: **if** ($ra \leq \beta_t$) **then**
- 15: $x''_i = x_k$
- 16: **end if** { $ra \in [0, 1]$ }
- 17: **end for**
- 18: **if** ($f(x'') \leq f(x)$) **then**
- 19: $x = x''$
- 20: $f(x) = f(x'')$
- 21: **end if**
- 22: $t = t + 1$
- 23: **end while**

III. PROPOSED METHODOLOGY

A. ENHANCED BINARY BUTTERFLY OPTIMIZATION ALGORITHM

In this section, the proposed feature selection model is implemented in two steps: In the first phase, we merged the $A\beta HC$ algorithm with three different binary variants of BBOA to maintain the required balance between exploration and exploitation states. This step helped to obtain better solutions in the form of a reduced feature set. While in the second phase, a set of classifiers was applied to evaluate the effectiveness of selected features in data classification. The detailed work of the proposed improvement strategy is illustrated in Fig. 2.

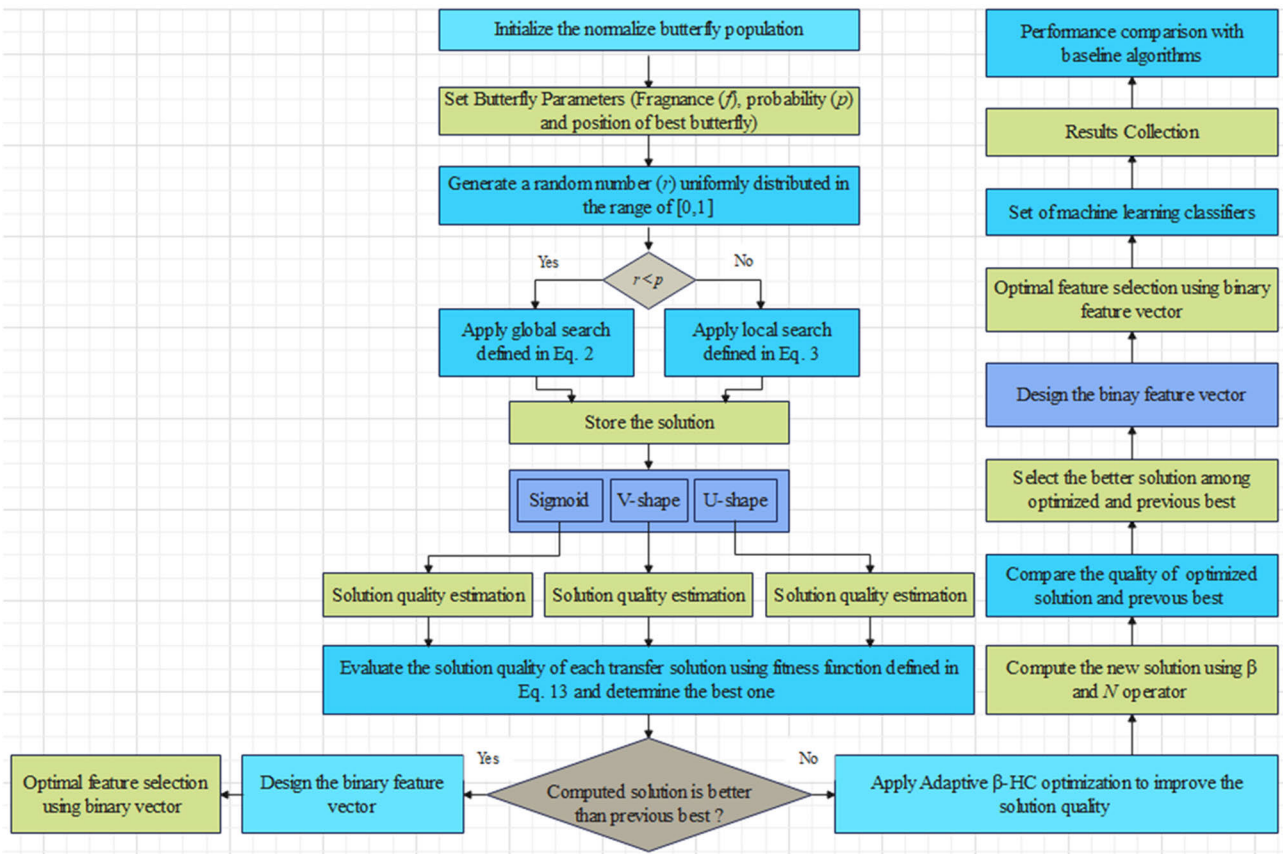


FIGURE 2. Working on the proposed feature selection model.

In our study, a linear weighted fitness function with two objectives: (1) classification error rate and (2) the relative number of selected relevant features is used to evaluate the quality of the computed solution. Here, the relative number of selected features is derived by dividing the number of selected features by the total number of available features. The formula of the fitness function is given in Equation 13. A lower fitness value reflects a better quality solution than a solution with a higher fitness value.

$$Fitnessfunction = w_1 * O_1 + w_2 * O_2 \quad (13)$$

where O_1 and O_2 indicate the classification error rate and the relative number of selected features, respectively. In addition, the sum of w_1 and w_2 is always equal to 1. Our improvement algorithm is described below:

1) Initially, BOA-specific parameters (fragrance (f), sensor quality (c), power exponent (a)) are fixed using a set of experiments. Initially, we used a grid search pattern to compute the best parameter pair (a, c) for three high-dimensional datasets (Penglungew, TOX-17, Yale: Table 1) based on maximum classification accuracy achieved within 100 iterations. The details of computed parameters are listed in Table 2. Here, the best position of a butterfly is considered by transforming it into a binary feature vector using the

sigmoid transfer function and minimizing the respective classification accuracy error. A butterfly with minimum features and maximum classification accuracy is considered the best among its neighbors.

- 2) The proposed algorithm executes the global and local search ((Equation 2 and 3, respectively) using a randomization procedure and switching probability (0.5). The switching probability provides equal chances to compute global and local solutions for the given number of iterations.
- 3) Three transfer functions (S, V, and Q-shape) are applied in each iteration to realize the best binary equivalent feature vector and respective average classification accuracy is recorded. If the performance is better than state-of-the-art algorithms, then the BOA switches to the next iteration; otherwise, the β HHC optimization is applied to find an improved solution. Here, the quality of an improved and previous solution is expressed in terms of the fitness value computed in Equation 13.
- 4) The improved solution is converted into an equivalent binary feature vector, and the classification accuracy is computed using various classification techniques. The pseudocode of the proposed methodology is given in Algorithm 3.

TABLE 1. Statistical description of twenty UCI datasets.

| Index | Dataset | Category | Number of features (D) | Number of samples (S) | Number of classes |
|-------|-------------|--------------------|------------------------|-----------------------|-------------------|
| 1 | Australian | Financial | 14 | 690 | 2 |
| 2 | Credit | Financial | 20 | 1000 | 2 |
| 3 | CTG | Life or Biological | 22 | 2126 | 3 |
| 4 | Exactly | Life or Biological | 13 | 1000 | 2 |
| 5 | Diabetic | Life or Biological | 20 | 1151 | 2 |
| 6 | Hill Vally | Graphical | 100 | 606 | 2 |
| 7 | Ionosphere | Physical | 34 | 351 | 2 |
| 8 | Libras | Biological | 90 | 360 | 15 |
| 9 | M-of-N | NA | 13 | 1000 | 2 |
| 10 | OBS-Network | Network | 21 | 1075 | 4 |
| 11 | Penglungew | Life or Biological | 325 | 73 | 2 |
| 12 | QSAR | Biological | 41 | 1055 | 2 |
| 13 | Sonar | Physical | 60 | 208 | 2 |
| 14 | Spambase | Computer | 57 | 4601 | 2 |
| 15 | Spect | Life or Biological | 22 | 267 | 2 |
| 16 | TOX-171 | Life or Biological | 5748 | 171 | 4 |
| 17 | Vote | Social | 16 | 300 | 2 |
| 18 | Vowel | Computer vision | 13 | 990 | 10 |
| 19 | Waveform | Physics | 21 | 5000 | 3 |
| 20 | Yale | Computer vision | 1024 | 165 | 15 |
| Mean | - | - | 383.70 | - | - |

B. CLASSIFICATION PERFORMANCE

To evaluate the performance of the proposed improvement strategy, three classifiers: (1) Support Vector Machine, (2) Naïve Bayes, and (3) Majority voting-based ensemble techniques are used, and their results are recorded. A short introduction to applied classification techniques is given below.

1) SUPPORT VECTOR MACHINE

Support Vector Machine (SVM) [27] is a popular supervised machine learning algorithm for classification, regression, and outlier detection. It aims to find a maximum marginal hyperplane in N -dimensional feature space that distinctly discriminates the input data points. Here, the term maximum marginal indicates the maximum distance between data points of both classes. Maximizing the distance provides an extent of enforcement so that new data can be effectively placed in the appropriate class. During distance maximization, SVM follows the “Structural Risk Minimization” principle [28], where a unique hyperplane is selected based on its resistive behavior against the overfitting issue. Therefore, it minimizes the probability of placing new or unseen data points into the wrong class. Unlike other popular classification methods, such as Artificial Neural Networks (ANNs), SVM does not require large training data for learning purposes.

Also, an SVM model effectively deals with high-dimensional datasets without increasing spatial complexity.

2) NaïVE BAYES

Naïve Bayes (NB) classifiers belong to a family of probabilistic classifiers that work on the Bayes Theorem [29]. In simple terms, a Naïve Bayes classifier assumes strong independence between different features available in the dataset. When these assumptions truly hold, the Naïve Bayes classifier achieves better classification accuracy with few training data than other models such as SVM and ANNs. Naïve Bayes classifiers are fast, memory efficient, and immune to overfitting, making them a robust classification approach for noisy data samples.

3) MAJORITY VOTING-BASED ENSEMBLE METHODS

Ensemble methods are classification techniques combining base models to design one optimal predictive model [30]. In the voting mechanism, each classification approach predicts the class of new data observation in the form of either vote or probability. In the case of votes, the true class label is declared with the most votes given by all available base models. Contrary to the voting system, the ensemble estimator involves summing the predicted probabilities (or probability-like scores) for each class label and predict-

TABLE 2. Statistical details about parameters used in all the state-of-the-art algorithms and proposed method.

| Algorithm | Parameter Setting |
|---------------|--|
| GA | Crossover_ratio = 0.9, Mutation_ratio = 0.1, M (number of runs) = 30, N (number of iterations) = 100 |
| GOA | c_Max = 1, c_Min = 0.0004, M (number of runs) = 30, N (number of iterations) = 100 |
| PSO | Acceleration_constants ($C1 = 2, C2 = 2$), M (number of runs) = 30, N (number of iterations) = 100 |
| ALO | $I = 1$ set as in the original article [32]. |
| SCA | a - Power exponent = 2, as in the original article [34]. |
| BOA | a - Power exponent = 0.1, as in the original article [18]. |
| CBOA | Control parameter (P) = 0.5, Chaotic numbers $\in (0,1)$, Constant (b) = 0.2. These parameters are listed in the original article [35]. |
| DBOA | a - Power exponent = 0.1 as in [18], $nm = 20$, $mu = 0.1$ [36]. |
| OEbBOA | M (number of runs) = 20, P (number of search agents) = 7, N (number of iterations) = 100, Search domains = [0,1], a - Power exponent = 0.1, c - Sensor modality = [0.01-0.25], τ_{max} (upper bound of shape tune parameter) = 4, τ_{min} (lower bound of shape tune parameter) = 0.01, F (scaling parameter) = [0,1], Crossover_ratio = 0.7, P_r (random variation paramter) = 0.7. These parameters are listed in the original article [37]. |
| S-bBOA | K for cross-validation = 5, M (number of runs) = 20, P (number of search agents) = 7, N (number of iterations) = 100, Search domains = [0,1], Crossover_ratio = 0.9, Mutation_ratio = 0.1, a - Power exponent = 0.1, c - Sensor modality [min, max] = [0.01-0.25]. |
| IFS-DBOIM | K for cross-validation = 5, M (number of runs) = 30, N (number of iterations) = 100, a - Power exponent = 0.5, c - Sensor modality [min, max] = [0.01-0.50]. SVM parameters ($C = 0.01, \gamma = 100$) [1]. |
| Enhanced BBOA | K for cross-validation = 5, M (number of runs) = 30, N (number of iterations) = 100, a - Power exponent = 0.5, switching probability (p) = 0.5, c - Sensor modality [min, max] = [0.01-0.50]. SVM parameters ($C = 0.01, \gamma = 100$) (Current work). |

ing the class label with the largest probability. Ensemble methods have two main advantages over conventional classification models. They are (1) Performance: An ensemble can make better predictions and achieve better performance than any single contributing model. Robustness: An ensemble reduces the spread or dispersion of the predictions and model performance.

C. COMPUTATIONAL COMPLEXITY

The performance of the proposed model depends on three major steps: (1) Raw solution computation, (2) Quality improvement (hybridization with $A\beta HC$ optimization), and (3) classification. In the first step, three substeps: (1) BOA execution, (2) position alteration using transfer function, and (3) fitness function comparison are involved. In the second

Algorithm 3 General Pseudocode of Proposed Feature Selection Algorithm

- Initialize n butterflies population positions x_i ($i = 1, 2, \dots, n$)
 - Set the initial value of parameters (switching probability ρ , sensory modality c , power exponent a , and the number of iterations N)
1. **while** not reach N **do**
 2. **for** each butterfly bf in the population **do**
 3. Compute the fragrance value f for each bf using Eq. 1
 4. **end for**
 5. Find the best butterfly bf
 6. Assign the best butterfly to g^*
 7. **for** each butterfly bf in the population **do**
 8. Generate a random value r over the interval [1,0]
 9. **if** ($r < \rho$)
 10. Update bf position by using Eq. 2– (Exploration)
 11. Else
 12. Update bf position by using Eq. 3– (Exploitation)
 13. **end if**
 14. Transform the updated position into respective binary vectors using S-, V-, and Q-shaped transfer functions.
 15. Compute the quality of binary feature vectors using Eq. (13) and select the best one as $Updated_{best}$.
 16. **if** ($g^* < Updated_{best}$)
 17. $g^* = Updated_{best}$
 18. Else
 19. Apply $A\beta HC$ optimization using N and β -operator to improve the solution quality
 20. **GOTO** Line 14
 21. **if** ($N = Current\ iteration$)
 22. Compute the mean classification accuracy using mentioned classifiers
 23. Compare the results with benchmark solutions.

and third phases, $A\beta HC$ techniques and classification techniques are applied to determine the true class label. In BOA, a dataset representing N butterflies and K dimensions updates its positions in $O(N * K)$ time because each butterfly updates its location in terms of all dimensions. Similar revisions are performed in the position alteration phase, where three transfer functions were applied. The complexity of this phase will be the same as BOA execution and equal to $O(N * K)$. For N butterflies or solutions, the fitness function comparison and best solution estimation will take $O(N)$ time.

In the second phase, the quality of N solutions is improved using the $A\beta HC$ method. The analysis of this shows that the worst-case complexity of the hybridization process is $O(\text{Number of iterations} * (N * t_{fitness} + K))$ where $t_{fitness}$ is the time required for calculating the fitness value using a given classifier. So, the overall complexity of the proposed model can be represented as the sum of all the

mentioned steps, which can be given as $O[(N * K) + (N) + (Number\ of\ iterations * (N * t_{fitness} + K))]$.

IV. EXPERIMENTAL RESULTS AND ANALYSIS

In this section, we evaluate the performance of the proposed model on twenty standard datasets taken from the University of California Irvin (UCI) repository [31] and compare it with eleven state-of-the-art feature selection algorithms. It should be known that results corresponding to all baseline methods are directly taken from a recently published article [1]. This section is structured as follows: experimental setup and dataset description are given in subsections IV.A and IV.B, respectively. Subsection IV.C consists of measuring criteria to evaluate the performance of the proposed model. Finally, the result analysis and their comparison with the baseline feature selection approaches are described in subsection IV.D.

A. DATASETS DESCRIPTION

Twenty high-dimensional datasets from different research domains are used to validate the proposed enhanced bBOA algorithm. The details of selected datasets are given in Table 1 as the number of classes, categories, samples, and features. Each dataset contains various characteristics in the context of attributes and sample size. For example, Penglungew, TOX-171, and Yale are high-dimensional (>300) datasets with fewer samples. Therefore, a proper cross-validation scheme is used to avoid the overfitting issue caused by the three datasets mentioned above. Similarly, CTG, Libras, OBS-Network, TOX-171, Vowel, Waveform, and Yale are multiclass (>2) datasets. All datasets are normalized before applying the IFS-DBOIM method. The experimental results are computed on Matlab 2019b on a laptop with an Intel®Core™i3 Processor, 4.2 GHz CPU frequency, 8 GB memory, and 1 TB secondary storage with a Windows 10 operating system.

B. EXPERIMENTAL SETUP

In our work, a five-fold cross-validation scheme is applied to each dataset to test the effectiveness of the proposed methodology and avoid overfitting issues. In other words, the datasets are divided into training and testing data samples in the following manner. In the first iteration, 80% of feature vectors are used for training, and the remaining 20% are employed for testing purposes. In the next, another 20% of feature vectors are used for testing, and the rest of the 80% are employed for the training set. This process is repeated until all the feature vectors are used for testing the proposed algorithm. All the data instances are normalized in intervals of 0 and 1. To quantify results statistically, each fold is repeated 30 times, and every experiment is performed 100 times, giving a total of 15,000 runs for each dataset. Next, the predictive classification model is developed on the training data and validated on the testing data, and the results are computed. Finally, the results are averaged over all the folds and compared with state-of-the-art methods. All parameter settings for each of the baseline and proposed algorithms are

given in Table 2. The computed results are compared with two groups of baseline feature selection algorithms.

The computed results are compared with two groups of baseline feature selection algorithms. In the first group, five naïve evolutionary algorithms, namely, (1) Ant Lion Optimization (ALO) [32], (2) Genetic Algorithm (GA) [3], (3) Grasshopper Optimization Algorithm (GOA) [33], (4) Particle Swarm Optimization (PSO) [4], and (4) Sine-Cosine Algorithm (SCA) [34] are used to compare the results. In the second group, conventional BOA with six different variants: (1) Butterfly Optimization Algorithm [18], (2) Chaotic Butterfly Optimization Algorithm (CBOA) [35], (3) Dynamic Butterfly Optimization Algorithm (DBOA) [36], (4) Iterative Feature Selection using Dynamic Butterfly Optimization-based Interaction Maximization (IFS-DBOIM) [1], (4) Optimization and Extension of binary Butterfly Optimization Algorithm (OEBBOA) [37], (6) S-shaped binary Butterfly Optimization Algorithm (S-bBOA) [38] are employed in results comparison. In addition, the number of fitness evaluations is used to determine the total number of iterations and fair comparison between state-of-the-art and proposed algorithms. The primary objective of using fitness functions as a performance measure is inspired by its ability to explore new information about solution quality computed in each iteration. Thus, limiting the number of fitness evaluations shows the total amount of information that our algorithm can obtain from a given dataset.

C. PERFORMANCE MEASURES USED IN THE STUDY

Performance evaluation is one of the crucial steps to showing the effectiveness of any proposed algorithm. It is advised to explore multiple performance measures rather than only one because a single performance metric may become biased toward a random dataset. To avoid this issue, a set of five performance measures: (1) Classification Accuracy, (2) Feature Reduction Rate, (3) Fitness Function, (4) Sensitivity, and (4) Specificity are used in the performance comparison. In addition, a nonparametric Wilcoxon signed-rank test [38] is also used to compare the significance of the computed results with baseline models. In this test, the results of two algorithms (Baseline methods and proposed approach) are computed, and their relevance is computed in the form of *p*value. If the computed *p*value is less than 0.05, then the results are statistically significant, and implementation of the proposed scheme is advisable otherwise may be ignored. The details of all the mentioned performance metrics are given below.

1) CLASSIFICATION ACCURACY

It is one of the important measures to reflect the discrimination ability of a classifier, as the number of correct predictions is divided by the total number of predictions. In an iterative procedure, Average Classification Accuracy (ACA) is computed over the total number of performed trials. The

mathematical representation of ACA can be defined as:

$$ACA = \frac{1}{M} \sum_{i=1}^M \frac{1}{N} \sum_{j=1}^N \text{match}(C_j, L_j) \quad (14)$$

where M is the total number of iterations the algorithm has executed, N represents the total number of observations in the test dataset, C_j and L_j indicate predicted and true class labels, respectively, and match is a comparison function that provides output 1 when both labels are the same and 0 when different.

2) FEATURE REDUCTION RATE

Feature Reduction Rate (FRR) shows how effectively irrelevant and redundant features are eliminated from the original feature set without compromising the global classification accuracy. It aims to minimize any feature selection method's computational and temporal complexity by effectively selecting only a significant set of features. In feature selection problems, a high FRR score is always desirable because of its positive influence on classification accuracy maximization. This performance measure can be computed as:

$$FRR = 1 - \frac{\text{number of selected features}}{\text{total number of original features}} \quad (15)$$

3) CONVERGENCE RATE

The convergence rate is an important measure that shows how rapidly an algorithm achieves a steady solution. A high convergence rate is always suitable for determining a design solution in fewer iterations. Otherwise, the algorithm is slow and requires more time to produce a steady solution.

4) SENSITIVITY

Refers to the ratio between actual positive or true cases that are predicted as positive (or true positive). This implies that another proportion of actual positive cases will get predicted wrongly or negatively (false negative). The formula of sensitivity is defined in Eq. 16.

$$\text{Sensitivity} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Negative}} = \frac{TP}{TP + FN} \quad (16)$$

5) SPECIFICITY

It is defined as the ratio of actual negative instances, which got computed as the true negatives. It indicates that there will be few other actual instances, which got computed as positive and could be called false positive instances. Specificity can be calculated as:

$$\text{Specificity} = \frac{TN}{TN + FP} \quad (17)$$

where TP is True_Positive, TN is True_negative, and FN is False_negative samples detected during classification.

D. RESULTS COMPARISON AND DISCUSSION

In this subsection, we perform two experiments to evaluate the robust performance of the proposed enhancement technique. In the first experiment, we compare our results with

six algorithms of group 1, while in the second, the remaining five competitive feature selection algorithms are used. The comparative results with all six algorithms are discussed in the subsequent subsections.

1) PERFORMANCE COMPARISON IN THE FIRST GROUP OF EXPERIMENTS

Table 3 summarizes the results of all six algorithms and compares them with our proposed methodology. Here, two measures: (1) Average Classification Accuracy (ACA) and (2) Feature Reduction Rate (P), is used to demonstrate the performance of all six algorithms over 30 iterations. In addition, the mean, standard deviation (S.D), and rank corresponding to classification accuracy are also provided. Here, the approach with a lower rank is considered more effective than the higher one. It is clear that the mean classification accuracy (91.67%) achieved by improved BOA is the maximum among all baseline feature selection methods. It realized the best classification accuracies on twelve datasets when used with SVM and ensemble classifiers. Also, the proposed approach obtained maximum classification accuracy with the Naïve Bayes approach only on one dataset. It can be concluded that our approach is more compatible with SVM and ensemble methods rather than the NB classifier.

It is interesting to discuss that our method consistently achieves the best results for high-dimensional multiclass datasets (Libras, Vowel, and Yale). They consist of more classes (≥ 10) than the remaining datasets. Henceforth, our method can be used to solve various real-time problems, such as language translation in social media platforms, neural state identification in cognitive science, and geographical data classification. It must be mentioned here that the number of labels in such problems is in the thousands, but the mapping function for the feature to true class label is only one; therefore, it is required to select an effective procedure to determine the corresponding class accurately. In addition to high ACA, our method employs fewer features (143.774) than other baseline methods. Compared to the average number of features (383.70) in Table 1, it discards approximately 62.66% of insignificant features and gains the best feature reduction rate.

In individual analysis, it effectively reduces the size of the feature set on eight datasets by eliminating redundant and irrelevant attributes. Although it selects more features in five datasets (Australian, Credit, Exactly, M-of-N, and Spect) compared to IFS-DBOIM (second-best method) but obtained maximum classification accuracy. It proves that our method can properly balance relevancy and redundancy while designing an optimal feature subset. However, the mean FRR score of our method is the best among all the methods. IFS-DBOIM is the second-best method that has shown competitive results with the current approach. One of the possible reasons may be the dynamic behavior of BOA and its hybridization with a feature interaction maximization scheme that simultaneously increases the solution quality and relevance of the selected features. In conventional evolutionary algorithms,

TABLE 3. Performance comparison between state-of-the-art evolutionary algorithms in group 1 and the proposed Enhanced BBOA algorithm in terms of average classification accuracy rate (in %) and the number of selected features (P) on twenty UCI dataset.

| No. | ALO | | GA | | GOA | | PSO | | SCA | | IFS-DBOIM | | Enhanced BBOA algorithm | | | |
|------|-------|---------|-------|---------|-------|---------|--------------|---------|-------|---------|--------------|--------------|-------------------------|--------------|--------------|----------------|
| | ACA | P | ACA | P | ACA | P | ACA | P | ACA | P | ACA | P | ACA | | | P |
| | | | | | | | | | | | | | SVM | NB | Ensemble | |
| 1 | 79.17 | 06.61 | 84.10 | 05.63 | 79.54 | 06.42 | 85.04 | 06.39 | 78.91 | 06.43 | 82.10 | 04.18 | 86.93 | 77.52 | 88.18 | 5.20 |
| 2 | 71.03 | 09.44 | 75.05 | 09.03 | 72.51 | 09.93 | 76.23 | 09.97 | 72.68 | 12.36 | 83.68 | 07.30 | 78.12 | 83.18 | 94.70 | 9.33 |
| 3 | 93.40 | 11.11 | 94.52 | 09.33 | 93.05 | 10.97 | 95.69 | 10.18 | 93.49 | 12.66 | 98.46 | 04.20 | 94.57 | 66.92 | 90.12 | 7.18 |
| 4 | 70.40 | 07.08 | 81.70 | 08.03 | 72.30 | 06.69 | 87.46 | 06.73 | 87.46 | 10.90 | 83.20 | 05.78 | 89.60 | 67.30 | 93.12 | 9.42 |
| 5 | 68.34 | 09.71 | 70.63 | 08.36 | 68.54 | 09.46 | 71.70 | 09.01 | 68.50 | 11.60 | 80.02 | 06.02 | 93.18 | 81.98 | 71.28 | 5.04 |
| 6 | 55.23 | 49.19 | 60.02 | 46.33 | 56.03 | 49.38 | 61.48 | 48.57 | 55.73 | 57.06 | 74.66 | 39.60 | 77.48 | 89.54 | 84.16 | 34.62 |
| 7 | 88.04 | 15.79 | 91.41 | 13.80 | 88.85 | 16.64 | 93.07 | 15.80 | 88.09 | 17.76 | 98.33 | 08.38 | 91.02 | 74.15 | 88.72 | 11.08 |
| 8 | 75.27 | 45.08 | 80.13 | 40.16 | 75.64 | 44.37 | 80.97 | 42.33 | 75.73 | 48.23 | 96.18 | 29.02 | 98.60 | 78.90 | 94.66 | 27.12 |
| 9 | 84.20 | 07.18 | 91.15 | 08.50 | 85.40 | 07.22 | 95.96 | 06.97 | 92.48 | 10.30 | 77.80 | 06.42 | 96.80 | 92.26 | 97.33 | 8.52 |
| 10 | 93.78 | 09.36 | 94.45 | 05.36 | 93.56 | 09.87 | 95.00 | 08.74 | 93.89 | 07.83 | 99.34 | 02.73 | 90.16 | 85.80 | 94.20 | 8.14 |
| 11 | 90.81 | 160.86 | 93.77 | 135.96 | 91.24 | 161.40 | 93.94 | 151.74 | 91.86 | 178.46 | 99.20 | 98.62 | 93.30 | 68.04 | 84.98 | 83.33 |
| 12 | 84.58 | 20.24 | 87.86 | 19.20 | 85.13 | 20.24 | 88.81 | 19.81 | 85.11 | 26.23 | 92.18 | 14.20 | 97.54 | 88.40 | 96.14 | 12.88 |
| 13 | 83.47 | 28.97 | 89.57 | 26.56 | 84.77 | 29.25 | 91.80 | 28.59 | 85.32 | 35.83 | 98.50 | 19.40 | 89.12 | 72.54 | 85.04 | 27.33 |
| 14 | 89.58 | 28.16 | 91.71 | 28.80 | 90.19 | 28.88 | 93.02 | 29.00 | 90.62 | 41.03 | 96.12 | 21.68 | 99.54 | 78.12 | 77.10 | 18.16 |
| 15 | 75.09 | 10.82 | 80.67 | 11.03 | 75.65 | 10.86 | 81.73 | 10.50 | 77.90 | 14.06 | 88.20 | 07.25 | 90.72 | 66.30 | 92.40 | 21.33 |
| 16 | 74.51 | 2875.79 | 82.14 | 2825.90 | 78.53 | 2878.48 | 85.36 | 2846.23 | 76.26 | 3268.73 | 97.02 | 2403.74 | 95.58 | 92.16 | 80.28 | 2208.14 |
| 17 | 94.66 | 07.85 | 95.94 | 07.16 | 94.77 | 07.93 | 96.38 | 07.58 | 95.05 | 08.73 | 99.33 | 03.82 | 88.50 | 90.00 | 94.33 | 7.34 |
| 18 | 90.11 | 07.31 | 93.24 | 08.86 | 90.57 | 07.25 | 93.50 | 07.05 | 92.02 | 10.63 | 87.20 | 07.92 | 97.52 | 92.18 | 83.92 | 9.76 |
| 19 | 79.96 | 11.32 | 82.54 | 13.93 | 80.64 | 11.79 | 83.17 | 11.47 | 83.09 | 19.00 | 89.22 | 12.33 | 93.40 | 81.36 | 88.52 | 9.16 |
| 20 | 62.57 | 509.28 | 68.15 | 492.96 | 64.08 | 509.80 | 72.70 | 501.21 | 64.17 | 563.23 | 87.92 | 383.16 | 91.76 | 69.14 | 93.42 | 352.40 |
| Mean | 80.21 | 191.55 | 84.43 | 186.24 | 81.04 | 191.84 | 86.15 | 188.89 | 82.41 | 218.05 | 90.43 | 154.28 | 91.67 | 79.78 | 88.60 | 143.774 |
| S.D | 10.75 | | 09.67 | | 10.40 | | 09.41 | | 10.69 | | 07.09 | | 5.78 | 9.05 | 6.71 | |
| Rank | 7 | | 4 | | 6 | | 3 | | 5 | | 2 | | 1 | | | |

only particle swarm optimization has shown some good results compared to both IFS-DBOIM and our approach. For large biological and vision datasets (Penglungew, TOX-171, and Yale), our method effectively eliminated approximately 75%, 61%, and 65% of insignificant attributes and achieved good classification accuracy. Compared to the IFS-DBOIM method, it reduces more features (10%) and filters only relevant and discriminable features for classification.

In addition to classification accuracy and feature reduction rate, Table 3 ranks each feature selection algorithm by considering the mean classification accuracies achieved on all twenty datasets. Since the current approach achieves the best mean classification accuracy of 91.67% with the SVM classifier, which is the maximum among all the methods, it ranks first, followed by IFS-DBOIM (90.43%), PSO (86.15%), GA (84.43%), SCA (82.41%), GOA (81.04%), and ALO (80.21%). This improvement is because of producing high-quality intermediate solutions corresponding to the suitable operator and transfer functions. Although confirming the goodness of produced numbers is time-consuming, it ensures high average classification accuracy when summarized over the total number of iterations. Also, the mutation and neighborhood operation between the previous best and current solution helps select the best outcome from the large pool of solutions.

In the solution improvement step, our scheme is restricted to producing two superior offspring than the previous best

solution, ensuring a good convergence mechanism to achieve maximum classification accuracy compared to other methods. It helps avoid the local optima, thereby replacing the worst solution with a better one. Indirectly, it boosts the exploitation ability of the BOA algorithm to find upgraded solutions based on the current ones populating the search space. Compared to our method, the remaining six algorithms had shown limited exploration ability because they were forced to deal with only generic solutions when no optimization scheme was implemented. Also, these methods evaluate intermediate solutions at a given instance without comparing their quality with existing outputs. It limits the generation of similar but high-quality solutions that can strongly minimize the probability of getting trapped in local optima. The key advantage of all the upgraded solutions over the existing ones is their dense distribution around the mean classification accuracy (in the case of SVM and ensemble) because our method has the least Standard Deviation (S.D.) among all the baseline methods.

Fig. 3 shows the relationship between classification accuracies and the number of selected features on all 20 experimental datasets. Here, two high-dimensional datasets (TOX-171 and Yale) are selected as standards to explain the variations in the classification accuracy concerning changes in the size of the optimal feature subset. These two datasets are good examples of high-dimensional space because they have many features/ attributes compared to the number of

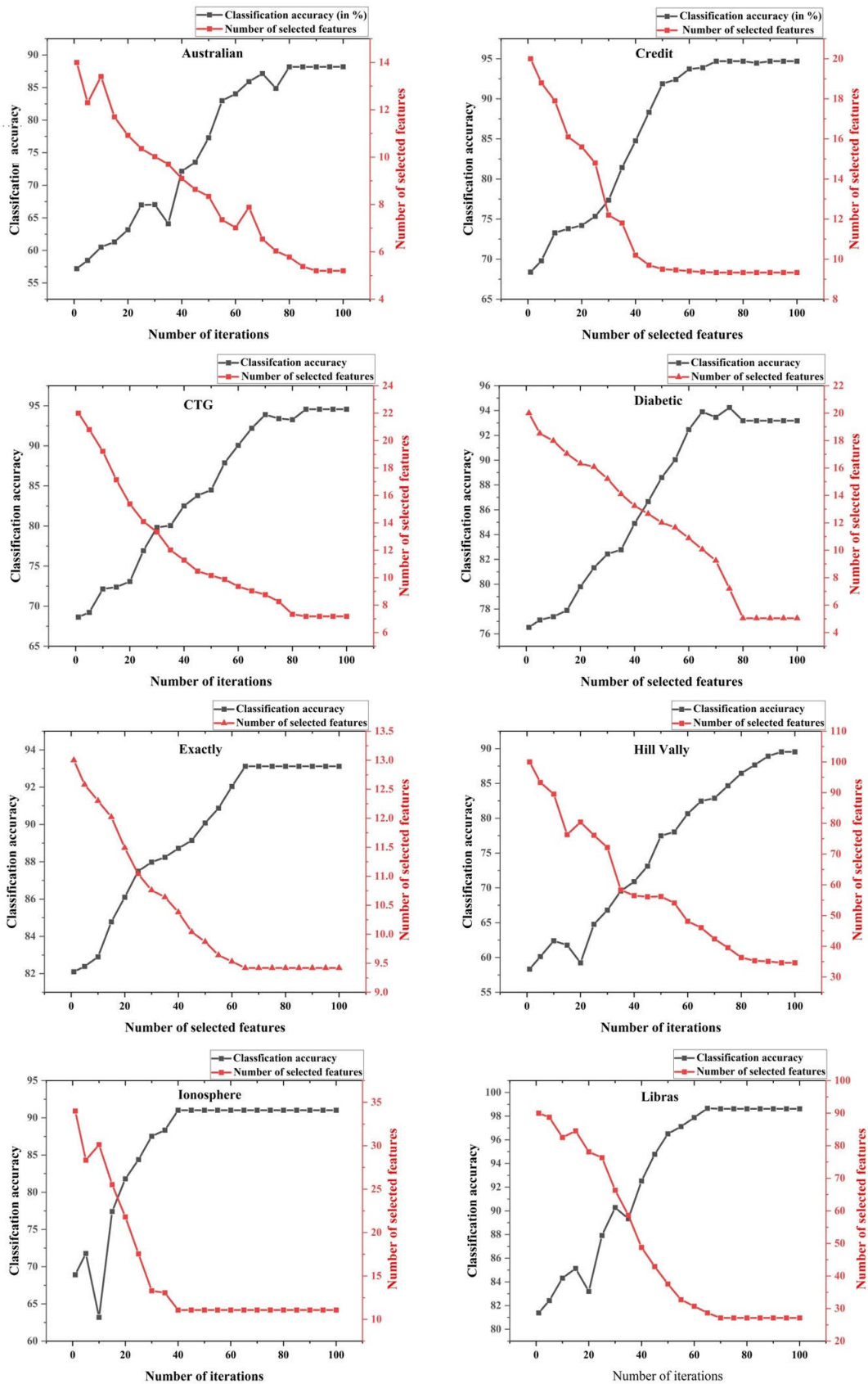


FIGURE 3. Iteration-wise relationship between classification accuracy and selection features for twenty datasets.

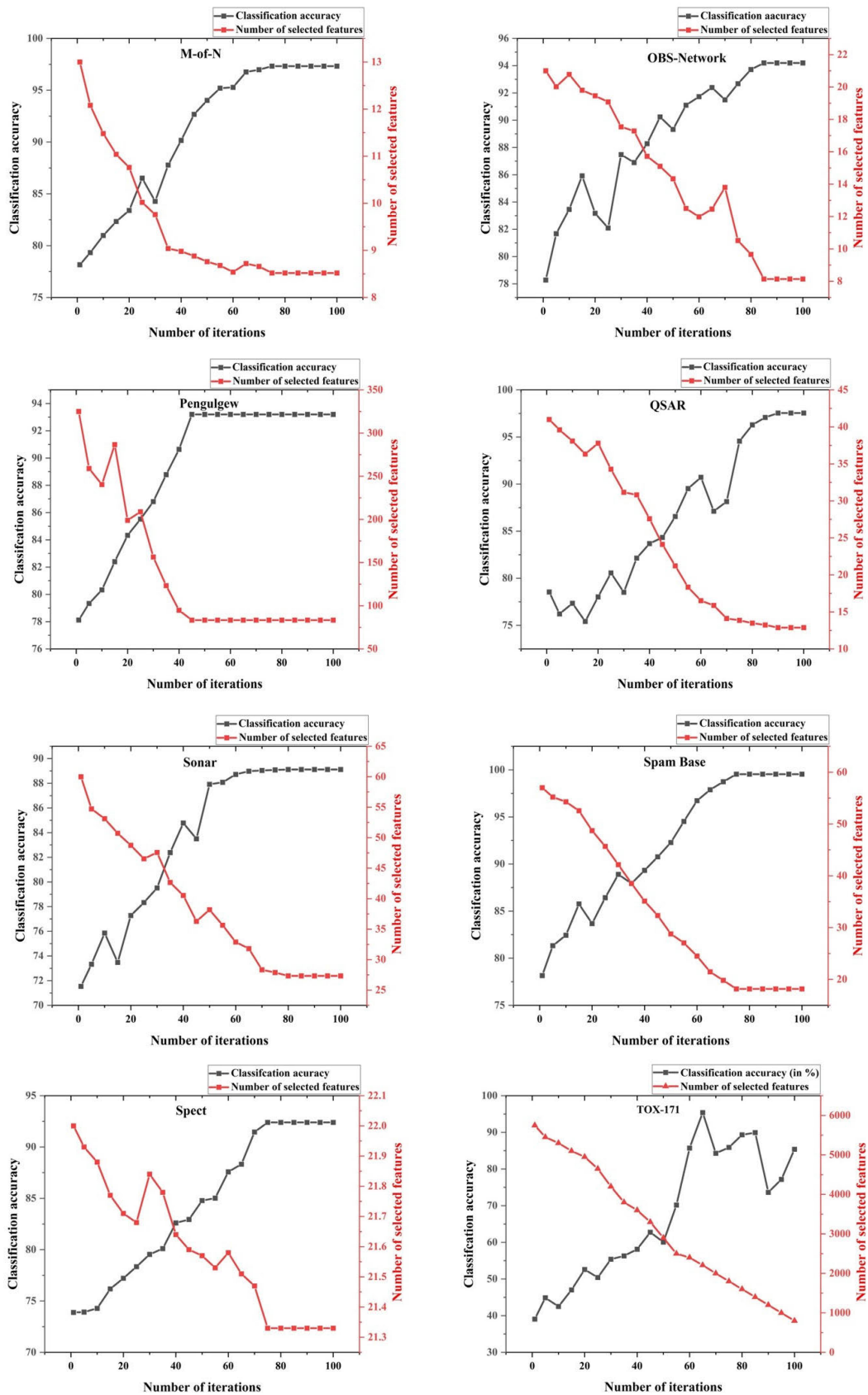


FIGURE 3. (Continued.) Iteration-wise relationship between classification accuracy and selection features for twenty datasets.

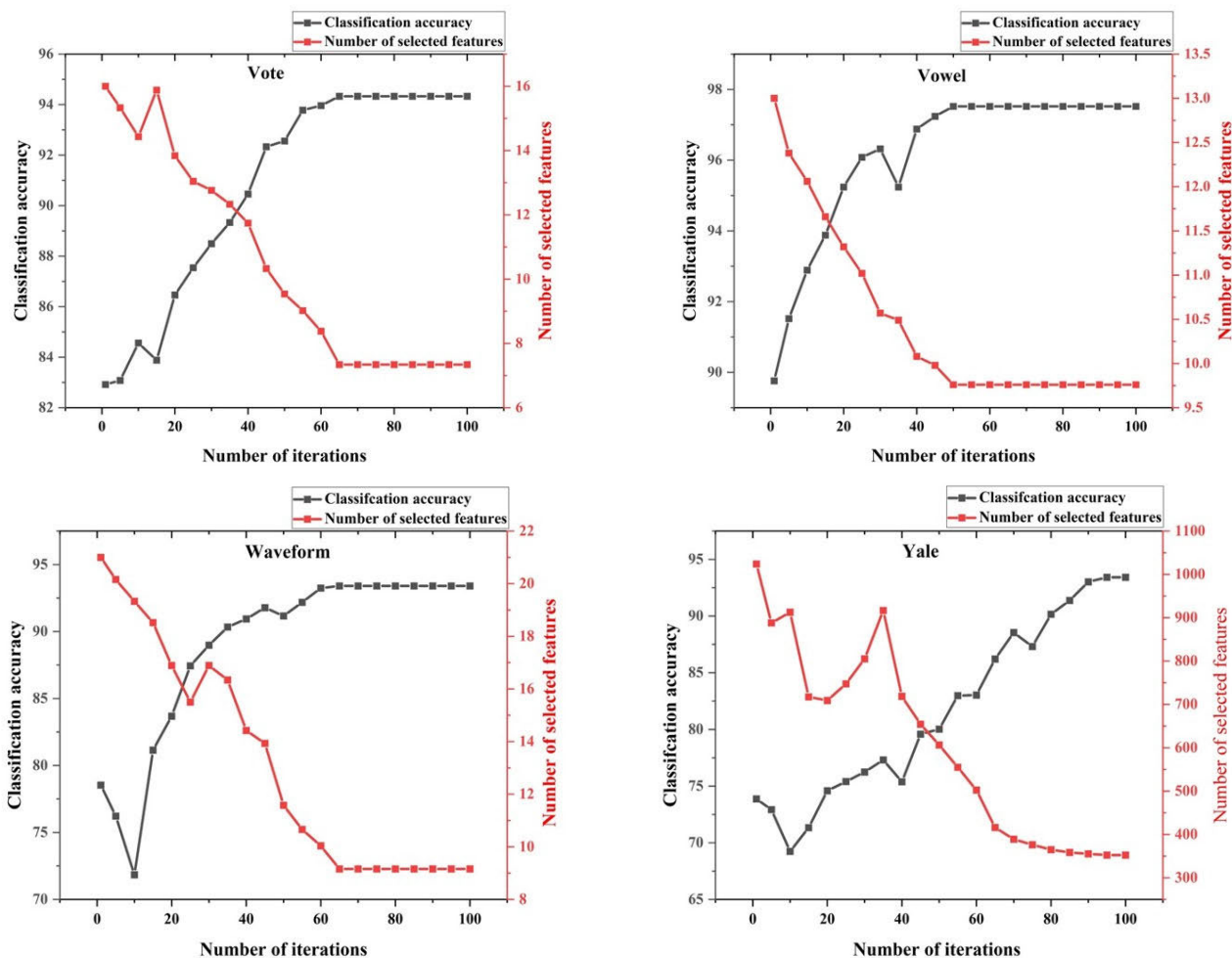


FIGURE 3. (Continued.) Iteration-wise relationship between classification accuracy and selection features for twenty datasets.

rows/observations. On the TOX-171 dataset, the proposed improvement mechanism realizes maximum classification accuracy (95.38%) using 2208 features within the maximum number of iterations.

In the early phase, the classification accuracy varies with the number of selected features and converges when 30% of the original attributes are used in the classification. It continuously evaluated the relevance and role of newly selected features in the accurate truth label prediction and achieved the best accuracy when approximately 62% of raw features were eliminated. On the contrary, the second-best method (IFS-DBOIM) consistently improves its classification accuracy with iterations but uses more features. Interestingly, both methods obtain almost equal classification accuracy when almost 75% of the features are diminished. However, in the second case, our method outperforms IFS-DBOIM from the early phase of iterations and gains the best classification accuracy and feature reduction rate with both SVM and ensemble classification approaches.

The simulation results of both datasets confirm that the classification accuracy rate need not compromise by many

features. Therefore, if an appropriate search scheme is applied in the early execution phase, higher classification accuracy can be realized in both datasets. However, it should be noticed that the proposed method achieves a set of stable solutions (high classification accuracy, high feature reduction rate) after completing 50 iterations on both TOX-171 and Yale datasets. Conversely, all remaining datasets except QSAR achieve a set of good responses in the very early phase of execution (≤ 40 iterations). It may be because of some dataset-specific properties such as data distribution, associations between variables and observations, and their support to applied metaheuristic algorithms. However, the shape of the curve for each dataset mainly depends on the performance of the substrate layer of all three transfer functions because they decide the intermediate feature set while improving the global classification accuracy.

2) PERFORMANCE COMPARISON IN THE SECOND GROUP OF EXPERIMENTS

In Table 4, the performance of the proposed model is compared in terms of ACA and FRR with five different variants

TABLE 4. Performance of the Enhanced BBOA in terms of classification accuracy rates (%) and the number of selected features for UCI datasets in the second group of experiments. Here, CA represents average classification accuracy (%), and P indicates the average number of selected features.

| No. | CBOA | | DBOA | | OebBOA | | S-bBOA | | Enhanced BBOA | | | |
|------|--------------|---------|--------------|--------------|--------------|--------------|--------|--------------|---------------|--------------|--------------|----------------|
| | CA | P | CA | P | CA | P | CA | P | SVM | NB | Ensemble | P |
| 1 | 79.77 | 06.12 | 88.04 | 04.53 | 83.22 | 07.18 | 79.12 | 09.12 | 86.93 | 77.52 | 88.18 | 5.20 |
| 2 | 74.35 | 10.08 | 77.71 | 08.70 | 78.11 | 13.55 | 74.42 | 12.52 | 78.12 | 83.18 | 94.70 | 9.33 |
| 3 | 94.21 | 07.12 | 98.46 | 04.53 | 97.15 | 06.20 | 96.58 | 05.44 | 94.57 | 66.92 | 90.12 | 7.18 |
| 4 | 82.31 | 08.30 | 99.85 | 06.50 | 89.25 | 08.92 | 97.24 | 10.16 | 89.60 | 67.30 | 93.12 | 9.42 |
| 5 | 68.77 | 07.22 | 73.15 | 06.56 | 71.33 | 06.46 | 68.11 | 07.34 | 93.18 | 81.98 | 71.28 | 5.04 |
| 6 | 56.11 | 38.72 | 63.98 | 43.20 | 61.44 | 52.40 | 57.88 | 44.12 | 77.48 | 89.54 | 84.16 | 34.62 |
| 7 | 97.70 | 11.33 | 95.48 | 09.16 | 96.65 | 10.44 | 90.70 | 08.40 | 91.02 | 74.15 | 88.72 | 11.08 |
| 8 | 79.44 | 38.42 | 83.75 | 33.93 | 81.00 | 41.22 | 76.61 | 36.14 | 98.60 | 78.90 | 94.66 | 27.12 |
| 9 | 92.66 | 07.30 | 99.76 | 06.76 | 96.99 | 06.98 | 97.20 | 08.26 | 96.80 | 92.26 | 97.33 | 8.52 |
| 10 | 93.77 | 04.36 | 97.50 | 02.73 | 95.11 | 03.33 | 94.76 | 05.25 | 90.16 | 85.80 | 94.20 | 8.14 |
| 11 | 91.20 | 164.48 | 96.74 | 104.63 | 92.91 | 142.23 | 87.75 | 192.41 | 93.30 | 68.04 | 84.98 | 83.33 |
| 12 | 87.55 | 23.50 | 90.28 | 18.20 | 91.03 | 24.75 | 86.22 | 27.62 | 97.54 | 88.40 | 96.14 | 12.88 |
| 13 | 94.20 | 18.24 | 96.13 | 21.40 | 95.14 | 25.28 | 93.62 | 23.68 | 89.12 | 72.54 | 85.04 | 27.33 |
| 14 | 90.91 | 31.20 | 94.13 | 33.00 | 91.40 | 29.68 | 91.11 | 27.58 | 99.54 | 78.12 | 77.10 | 18.16 |
| 15 | 82.90 | 11.78 | 86.54 | 09.70 | 85.16 | 13.21 | 84.63 | 12.44 | 90.72 | 66.30 | 92.40 | 21.33 |
| 16 | 77.12 | 3320.11 | 89.07 | 2743.86 | 83.07 | 2907.46 | 76.11 | 3120.12 | 95.58 | 92.16 | 80.28 | 2208.14 |
| 17 | 96.44 | 04.98 | 98.16 | 04.70 | 98.44 | 05.76 | 96.53 | 07.98 | 88.50 | 90.00 | 94.33 | 7.34 |
| 18 | 91.77 | 14.86 | 94.73 | 08.46 | 92.12 | 11.16 | 95.13 | 10.28 | 97.52 | 92.18 | 83.92 | 9.76 |
| 19 | 80.30 | 19.30 | 84.42 | 15.26 | 83.10 | 18.40 | 74.29 | 21.14 | 93.40 | 81.36 | 88.52 | 9.16 |
| 20 | 67.22 | 612.33 | 75.18 | 466.96 | 70.15 | 524.42 | 64.54 | 532.16 | 91.76 | 69.14 | 93.42 | 352.40 |
| Mean | 83.93 | 217.98 | 89.15 | 177.63 | 86.63 | 192.95 | 84.12 | 206.10 | 91.67 | 79.78 | 88.60 | 143.774 |
| S.D. | 10.91 | 723.90 | 09.85 | 597.17 | 10.01 | 632.92 | 11.72 | 678.69 | 5.78 | 9.05 | 6.71 | |
| Rank | 5 | | 2 | | 3 | | 4 | | 1 | | | |

TABLE 5. The average fitness values of all competing algorithms over 30 runs.

| No. | CBOA | | DBOA | | OebBOA | | S-bBOA | | Enhanced BBOA | | | |
|------|--------------|---------|--------------|--------------|--------------|--------------|--------|--------------|---------------|--------------|--------------|----------------|
| | CA | P | CA | P | CA | P | CA | P | SVM | NB | Ensemble | P |
| 1 | 79.77 | 06.12 | 88.04 | 04.53 | 83.22 | 07.18 | 79.12 | 09.12 | 86.93 | 77.52 | 88.18 | 5.20 |
| 2 | 74.35 | 10.08 | 77.71 | 08.70 | 78.11 | 13.55 | 74.42 | 12.52 | 78.12 | 83.18 | 94.70 | 9.33 |
| 3 | 94.21 | 07.12 | 98.46 | 04.53 | 97.15 | 06.20 | 96.58 | 05.44 | 94.57 | 66.92 | 90.12 | 7.18 |
| 4 | 82.31 | 08.30 | 99.85 | 06.50 | 89.25 | 08.92 | 97.24 | 10.16 | 89.60 | 67.30 | 93.12 | 9.42 |
| 5 | 68.77 | 07.22 | 73.15 | 06.56 | 71.33 | 06.46 | 68.11 | 07.34 | 93.18 | 81.98 | 71.28 | 5.04 |
| 6 | 56.11 | 38.72 | 63.98 | 43.20 | 61.44 | 52.40 | 57.88 | 44.12 | 77.48 | 89.54 | 84.16 | 34.62 |
| 7 | 97.70 | 11.33 | 95.48 | 09.16 | 96.65 | 10.44 | 90.70 | 08.40 | 91.02 | 74.15 | 88.72 | 11.08 |
| 8 | 79.44 | 38.42 | 83.75 | 33.93 | 81.00 | 41.22 | 76.61 | 36.14 | 98.60 | 78.90 | 94.66 | 27.12 |
| 9 | 92.66 | 07.30 | 99.76 | 06.76 | 96.99 | 06.98 | 97.20 | 08.26 | 96.80 | 92.26 | 97.33 | 8.52 |
| 10 | 93.77 | 04.36 | 97.50 | 02.73 | 95.11 | 03.33 | 94.76 | 05.25 | 90.16 | 85.80 | 94.20 | 8.14 |
| 11 | 91.20 | 164.48 | 96.74 | 104.63 | 92.91 | 142.23 | 87.75 | 192.41 | 93.30 | 68.04 | 84.98 | 83.33 |
| 12 | 87.55 | 23.50 | 90.28 | 18.20 | 91.03 | 24.75 | 86.22 | 27.62 | 97.54 | 88.40 | 96.14 | 12.88 |
| 13 | 94.20 | 18.24 | 96.13 | 21.40 | 95.14 | 25.28 | 93.62 | 23.68 | 89.12 | 72.54 | 85.04 | 27.33 |
| 14 | 90.91 | 31.20 | 94.13 | 33.00 | 91.40 | 29.68 | 91.11 | 27.58 | 99.54 | 78.12 | 77.10 | 18.16 |
| 15 | 82.90 | 11.78 | 86.54 | 09.70 | 85.16 | 13.21 | 84.63 | 12.44 | 90.72 | 66.30 | 92.40 | 21.33 |
| 16 | 77.12 | 3320.11 | 89.07 | 2743.86 | 83.07 | 2907.46 | 76.11 | 3120.12 | 95.58 | 92.16 | 80.28 | 2208.14 |
| 17 | 96.44 | 04.98 | 98.16 | 04.70 | 98.44 | 05.76 | 96.53 | 07.98 | 88.50 | 90.00 | 94.33 | 7.34 |
| 18 | 91.77 | 14.86 | 94.73 | 08.46 | 92.12 | 11.16 | 95.13 | 10.28 | 97.52 | 92.18 | 83.92 | 9.76 |
| 19 | 80.30 | 19.30 | 84.42 | 15.26 | 83.10 | 18.40 | 74.29 | 21.14 | 93.40 | 81.36 | 88.52 | 9.16 |
| 20 | 67.22 | 612.33 | 75.18 | 466.96 | 70.15 | 524.42 | 64.54 | 532.16 | 91.76 | 69.14 | 93.42 | 352.40 |
| Mean | 83.93 | 217.98 | 89.15 | 177.63 | 86.63 | 192.95 | 84.12 | 206.10 | 91.67 | 79.78 | 88.60 | 143.774 |
| S.D. | 10.91 | 723.90 | 09.85 | 597.17 | 10.01 | 632.92 | 11.72 | 678.69 | 5.78 | 9.05 | 6.71 | |
| Rank | 5 | | 2 | | 3 | | 4 | | 1 | | | |

of BOA. These variants are application-specific and employ different optimization techniques while curtailing insignificant features from the given datasets. For example, conventional BOA uses only food foraging of butterflies to obtain better solutions, while chaotic BOA explores different chaotic

mapping functions to improve the performance of BOA in terms of both local optima avoidance and convergence speed. DBOA is another popular variant that uses the Local Search Algorithm Based on Mutation (LSAM) operator to improve the solution quality by avoiding local optima problems and

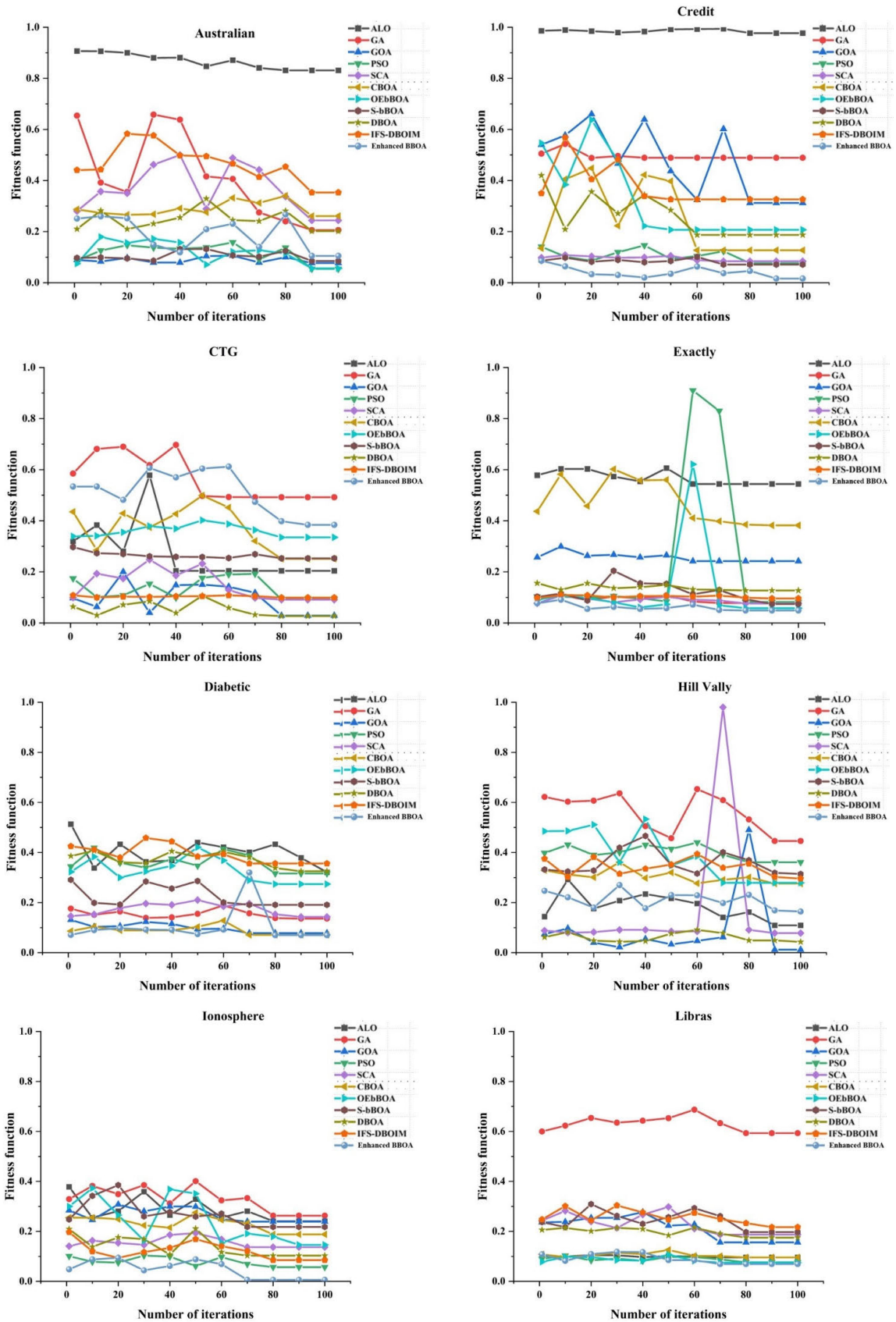


FIGURE 4. Graphical representation of fitness function variations corresponding to the number of iteration.

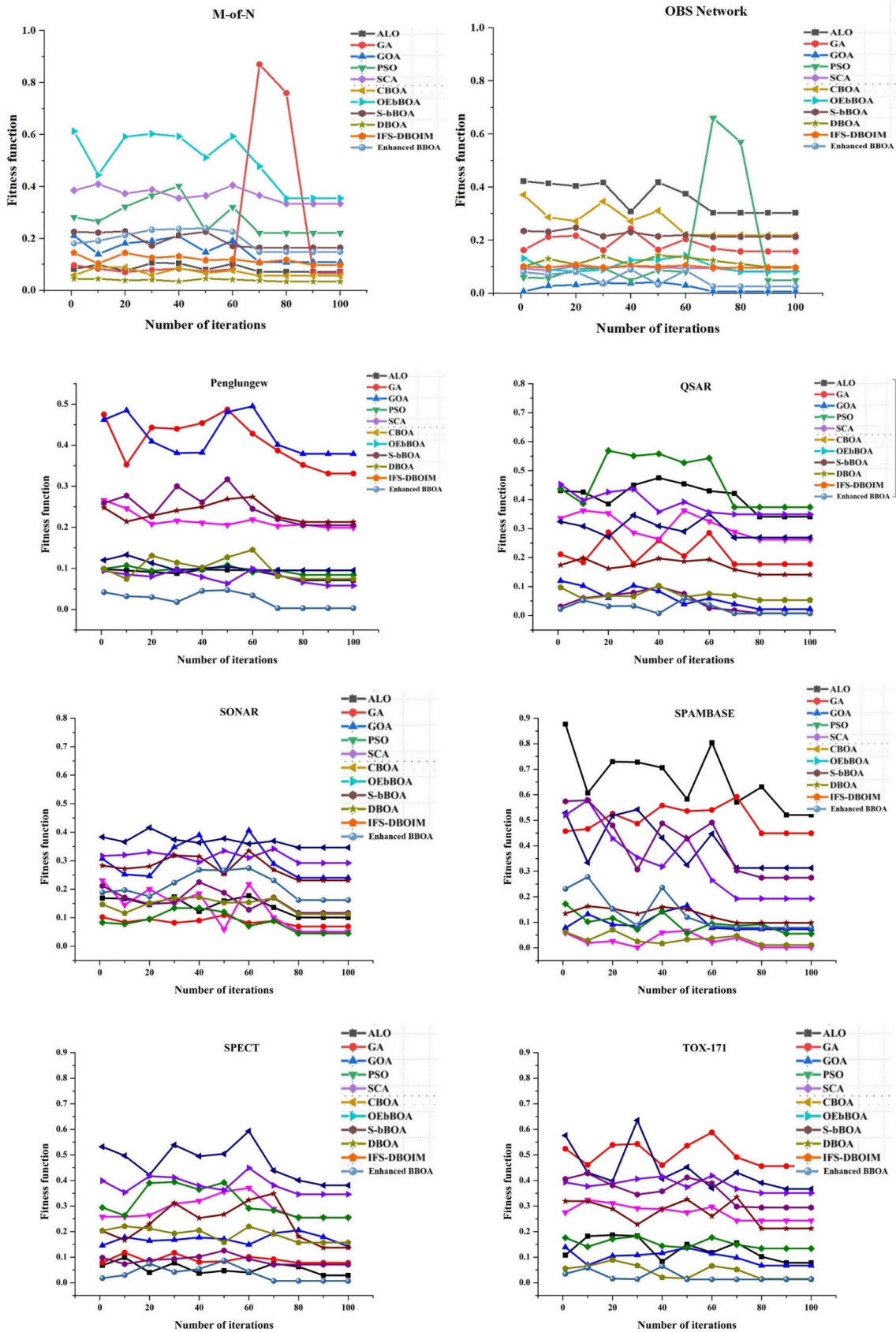


FIGURE 4. (Continued.) Graphical representation of fitness function variations corresponding to the number of iteration.

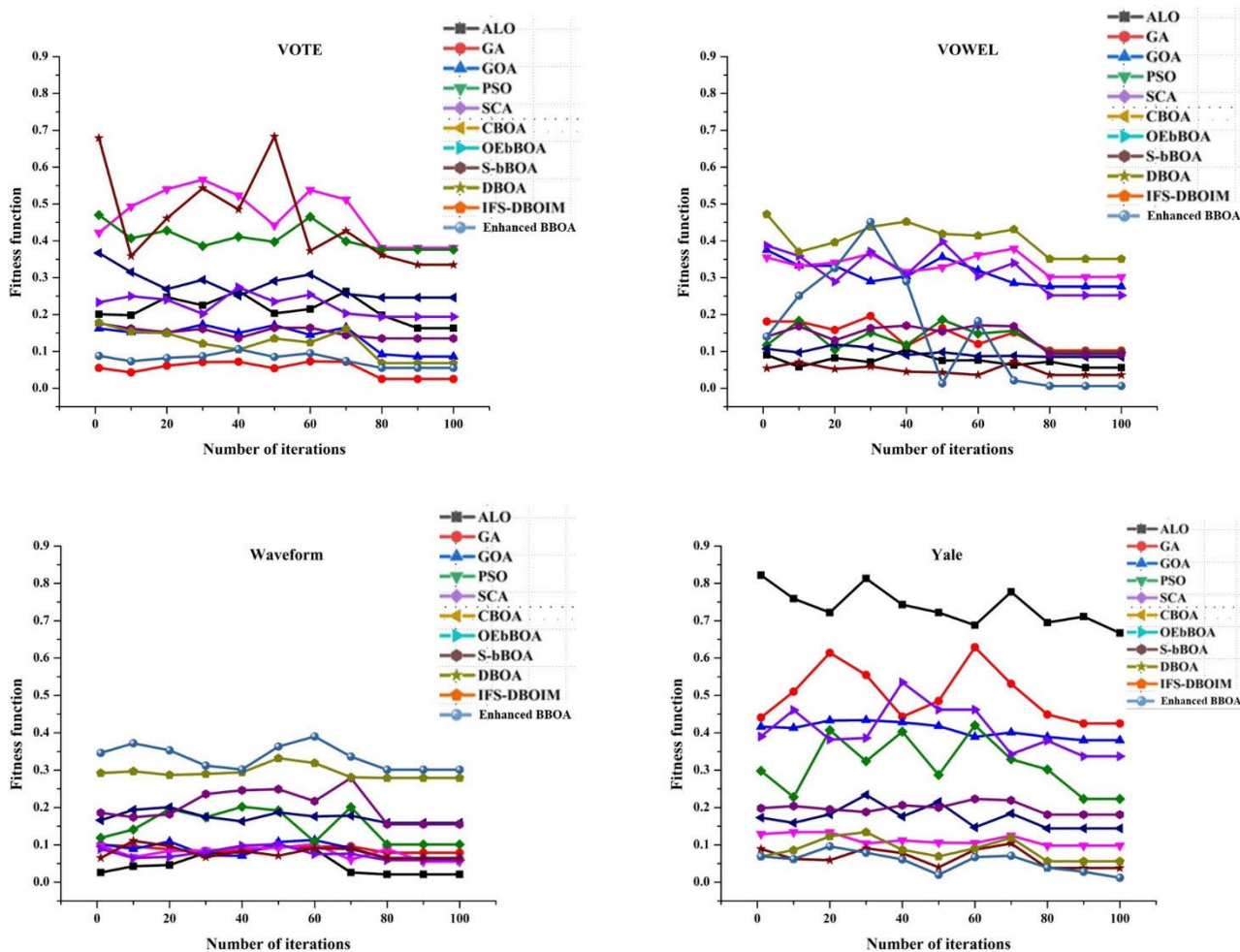


FIGURE 4. (Continued.) Graphical representation of fitness function variations corresponding to the number of iteration.

improving BOA solutions diversity. In the OebBOA, a new initialization strategy using the evolution population dynamics (EPD) mechanism is employed to increase the adaptive behavior of BOA and maintain the tradeoff between explorations and exploitations. Finally, S-shaped BOA refers to a binary variant of BOA that ensures the movement of all the butterflies within an interval of [1,0]. It can be observed that the introduced approach significantly outperforms thirteen out of twenty datasets in terms of classification accuracy and feature reduction rate. In two datasets with the largest dimensions, our method significantly reduces approximately 25% more features than DBOA and increases classification accuracy by 5% and 15%, respectively. It shows that our improved binary variant has better feature space optimization scope than mutation-based local search methods. In addition, it obtains more than 90% classification accuracies on all the datasets except the Australian, Hill Vally, and Sonar datasets. DBOA was another improved algorithm that shows competitive results along with our method. The main reason for these superior results may be the scalable nature of both N and β -operators and their ability to maintain the desired balance

between exploration and exploitation. Moreover, determining the suitable set of parameters in the early phase of execution may be another reason for performance enhancement.

The FRR of our method is also better than the other competitive methods. It uses fewer features than other BOA versions on eleven of twenty datasets. Overall, it has the best global feature reduction rate because it uses an average of 144 features on each UCI dataset, which is the least among all the methods. The order of the algorithms in terms of feature reduction rate is Enhanced BBOA > DBOA > OEBBOA > S-bBOA > CBOA.

3) FITNESS FUNCTION COMPARISON

Table 5 reports the fitness function score computed by all the above-mentioned eleven algorithms on twenty datasets. We obtained minimum fitness scores on twelve out of twenty datasets using the proposed optimization scheme, including the TOX-171 and YALE. Since a lower fitness value reflects the higher quality of the solutions, PSO and DBOA perform equally on four datasets (Australian, Spambase, CTG, and M-of-N). Moreover, each of the four algorithms (ALO, GA,

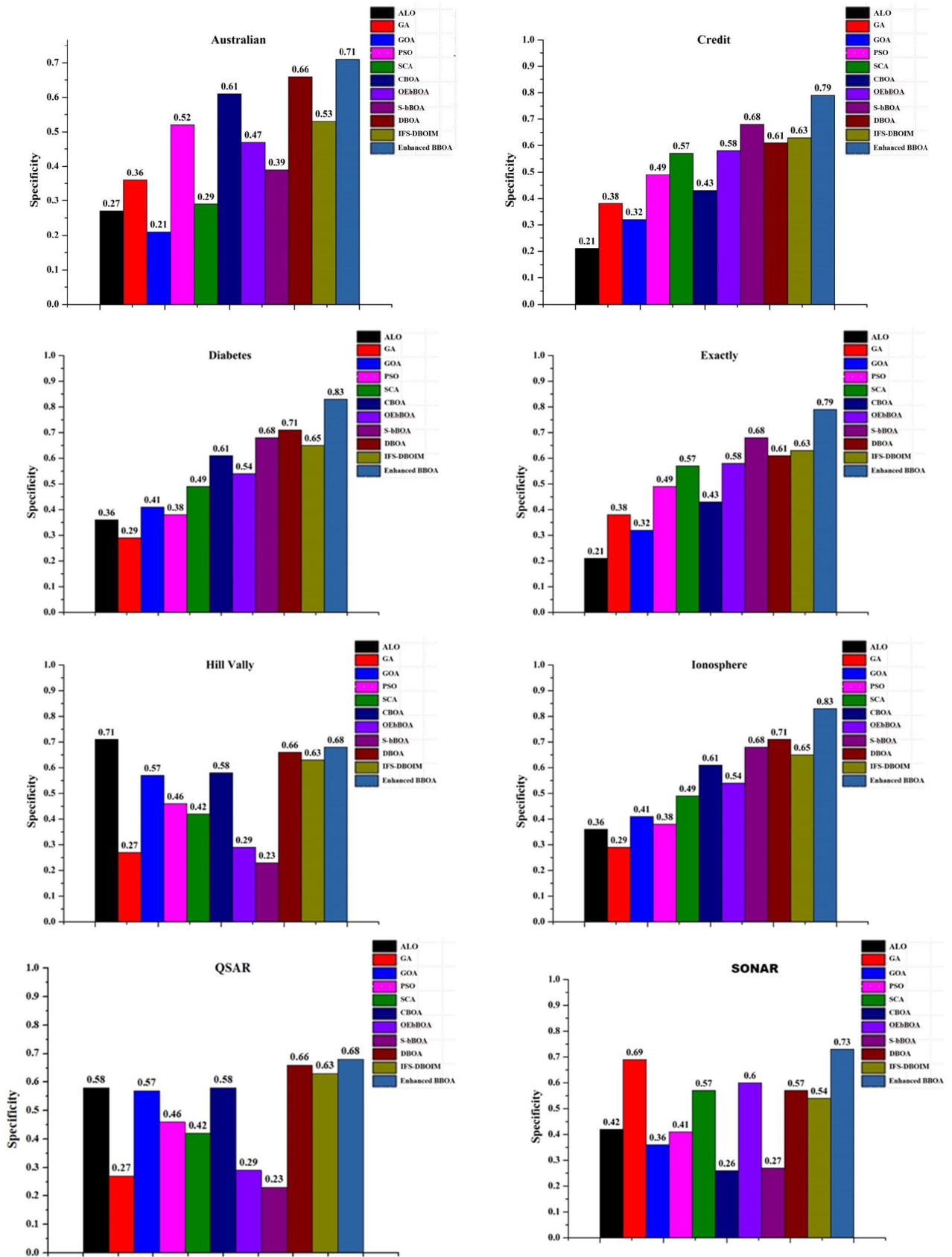


FIGURE 5. Specificity score for all thirteen binary datasets.

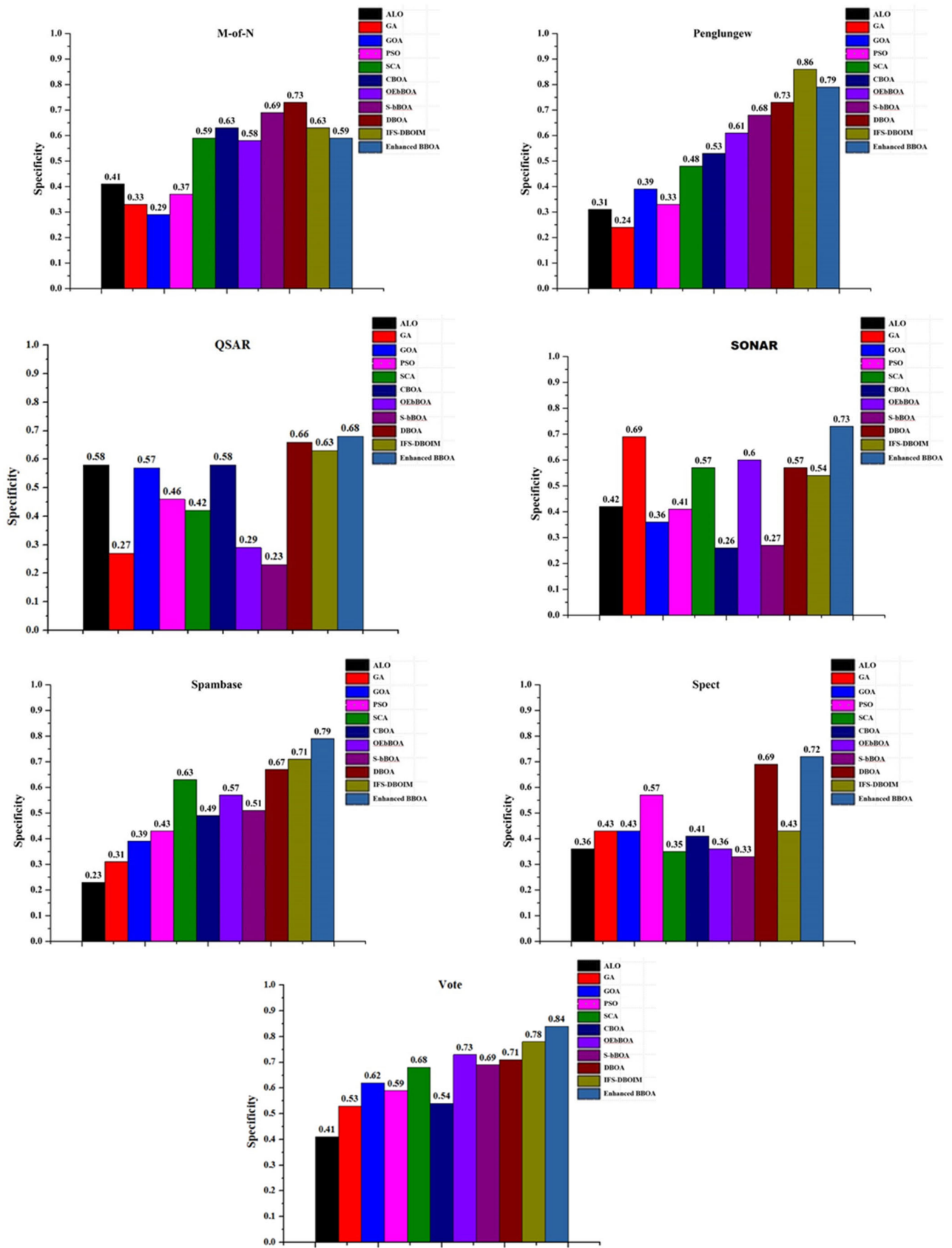


FIGURE 5. (Continued.) Specificity score for all thirteen binary datasets.

GOA, and SCA) computes the minimum score for individual datasets. Our method achieves the least (≈ 0) fitness score on five datasets (Ionosphere, Pengulgew, QSAR, Spect, and Vowel) while estimating the relevance of a selected attribute in the optimal feature subset. Here, three out of five (Pengulgew, QSAR, Spect) are biological datasets with a moderate number of dimensions. In contrast, the remaining two datasets (Ionosphere and Vowel) are related to the environment and computer vision domains.

The reported results confirm that our method strongly computes the relationship between classification accuracy and the number of features in similar datasets. In addition, our method realizes an almost equal fitness score on the top two high-dimensional datasets (TOX-171 and Yale) with an almost equal number of observations. Due to this, the proposed approach realizes maximum classification accuracy while using the minimum number of features. In all the baseline algorithms, ALO performed worst on eight datasets (Australian, Credit, Exactly, Diabetic, OBS-network, QSAR, SPAMBASE, Yale) because it achieved a maximum fitness score of more than 50% of the iterations. It may be because of ALO's unsuitable solution initialization strategy, improper parameter tuning, and poor tendency to shift towards better solutions. Fig.4 shows the relationship between the fitness score and the number of iterations for all the datasets.

4) SENSITIVITY AND SPECIFICITY PERFORMANCE

Specificity and sensitivity are other important performance parameters concerned with the performed experiment's accuracy relative to a given standard result. Sensitivity is the ratio of correctly classified positive instances to the total number of positive samples, and specificity is the ratio of correctly classified negative instances to the total number of negative samples. It should be noted that sensitivity and specificity are applicable only to binary classification problems. Therefore, we have shown sensitivity and specificity metrics only for thirteen datasets in Fig. 5 and 6, respectively. It can be noticed that enhanced bBOA obtained a better specificity score on eleven binary datasets, excluding M-of-N and Penglungew. In both cases, DBOA and IFS-DBOIM achieved maximum specificity scores compared to others. DBOA is another dynamic variant of BOA that obtains the second-best rank in specificity computation of all the binary datasets.

Similar to specificity, our proposed method also outperforms other sensitivity measurement methods. In Fig. 6, the proposed method achieved the best sensitivity scores on ten of thirteen datasets. On one dataset (Penglungew), the developed improvement scheme performs equally to the IFS-DBOIM approach with equal sensitivity values. Experiments confirm that the IFS-DBOIM-based feature selection method obtains second-best sensitivity scores for eight UCI datasets. This improved performance may be correlated with an inbuilt local search-based mutation scheme in the IFS-DBOIM.

5) TIME COMPLEXITY ANALYSIS

The time complexity of an algorithm refers to the total execution time that an algorithm takes to determine the desirable solution. Table 6 presents the total running time of all eleven baseline algorithms with the Enhanced BBOA over 30 independent iterations. Since the proposed method is an iterative procedure that generates quality solutions using the $A\beta HC$ optimization when required. It is a time-consuming step because it explores two different mathematical operators, and sometimes it may take a lot of time.

On the contrary, the existing feature selection methods rely on local solutions to determine the true class label corresponding to a given observation. However, the state-of-the-art models compromise with average classification accuracy because the generated solutions may result in poor discrimination ability of the applied classification scheme. It can be cross-verified from Table 6, where the proposed algorithm realizes the least execution time only on five out of twenty datasets, whereas the PSO outperformed on eleven datasets. DBOA was the third-best algorithm that achieved the least execution time on three datasets, while GOA performed the best runnable time only on one dataset.. Based on the mean computation time (on 20 datasets), PSO won the speedup race with rank one, while the proposed method was the slowest among all the methods. Overall, the order of execution rate is $PSO > ALO > DBOA > GOA > CBOA > IFS-DBOIM > S-bBOA > SCA > OEBBOA > GA > BOA > Enhanced BBOA$.

6) SOLUTION SIGNIFICANCE ANALYSIS

Significance analysis is an important measure to determine the obtained solutions' relevance. In order to determine the significance of classification results over 30 runs, a non-parametric Wilcoxon signed-rank test is performed on all twenty datasets with twelve algorithms. This test compares the results of two matched samples and returns a p -score. The results are considered significant if the computed p -value is less than 0.05, whereas a greater p -value indicates otherwise. Table 7 reports all the p -values considering the Enhanced BBOA as a benchmark solution. The relevant results are marked with dark color in Table 7. Our results are highly significant on at least ten datasets compared with IFS-DBOIM, DBOA, CBOA, GOA, and ALO because the corresponding p -values are less than the 0.05 threshold. Therefore, the computed results can be used as an improvement over the outcomes of all the mentioned five baseline algorithms. Our algorithm can be recommended for the remaining datasets in the place of BOA, GA, PSO, SCA, and OEBBOA approaches. The global order of result significance can be listed as $Enhanced BBOA > IFS-DBOIM > CBOA > DBOA > ALO > GOA > OEBBOA > PSO \approx BOA > SCA \approx GA$.

7) LIMITATIONS OF THE STUDY

In this study, we conduct multiple experiments to maximize the global performance of Enhanced BBOA in feature

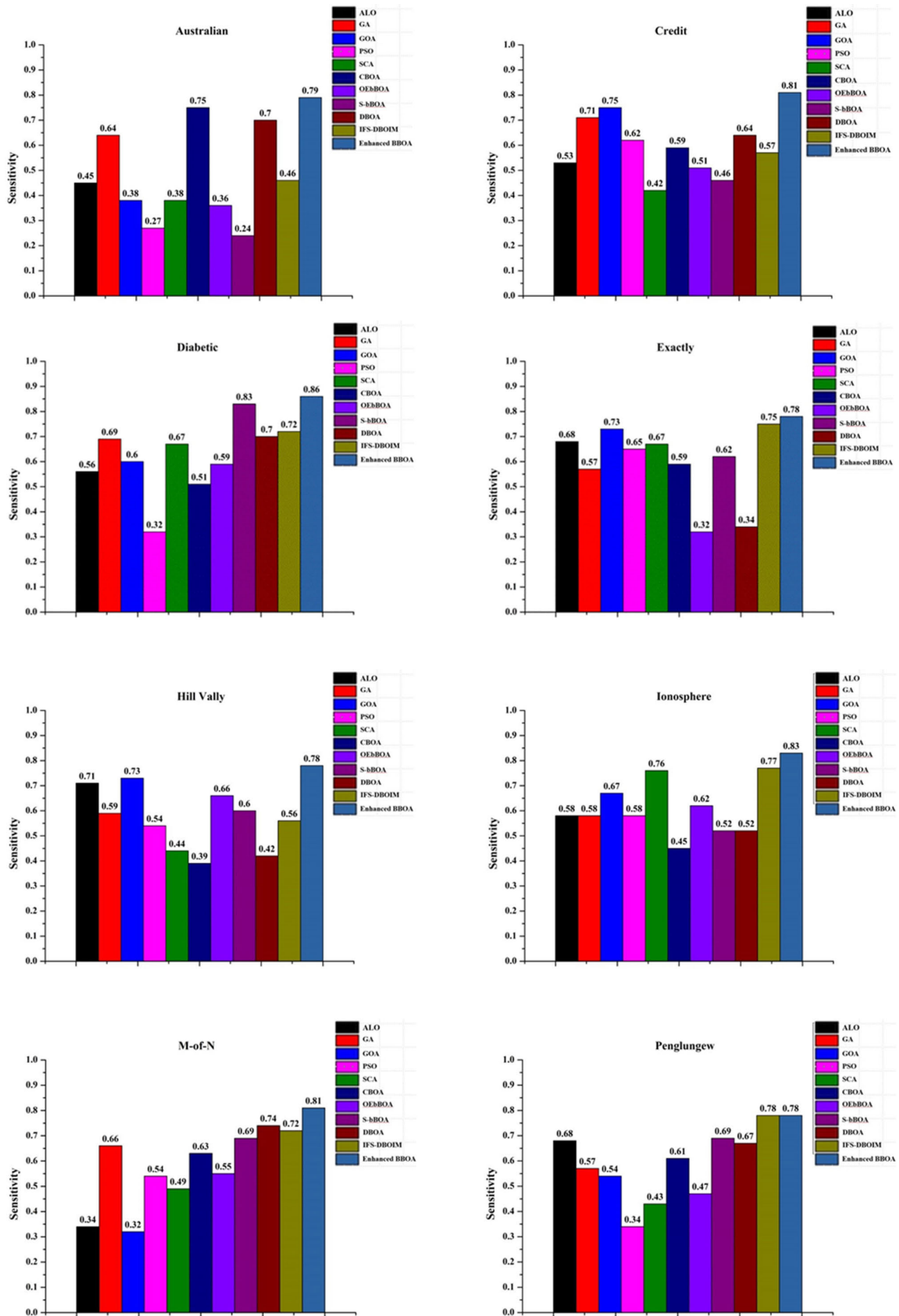


FIGURE 6. Sensitivity score for all thirteen binary datasets.

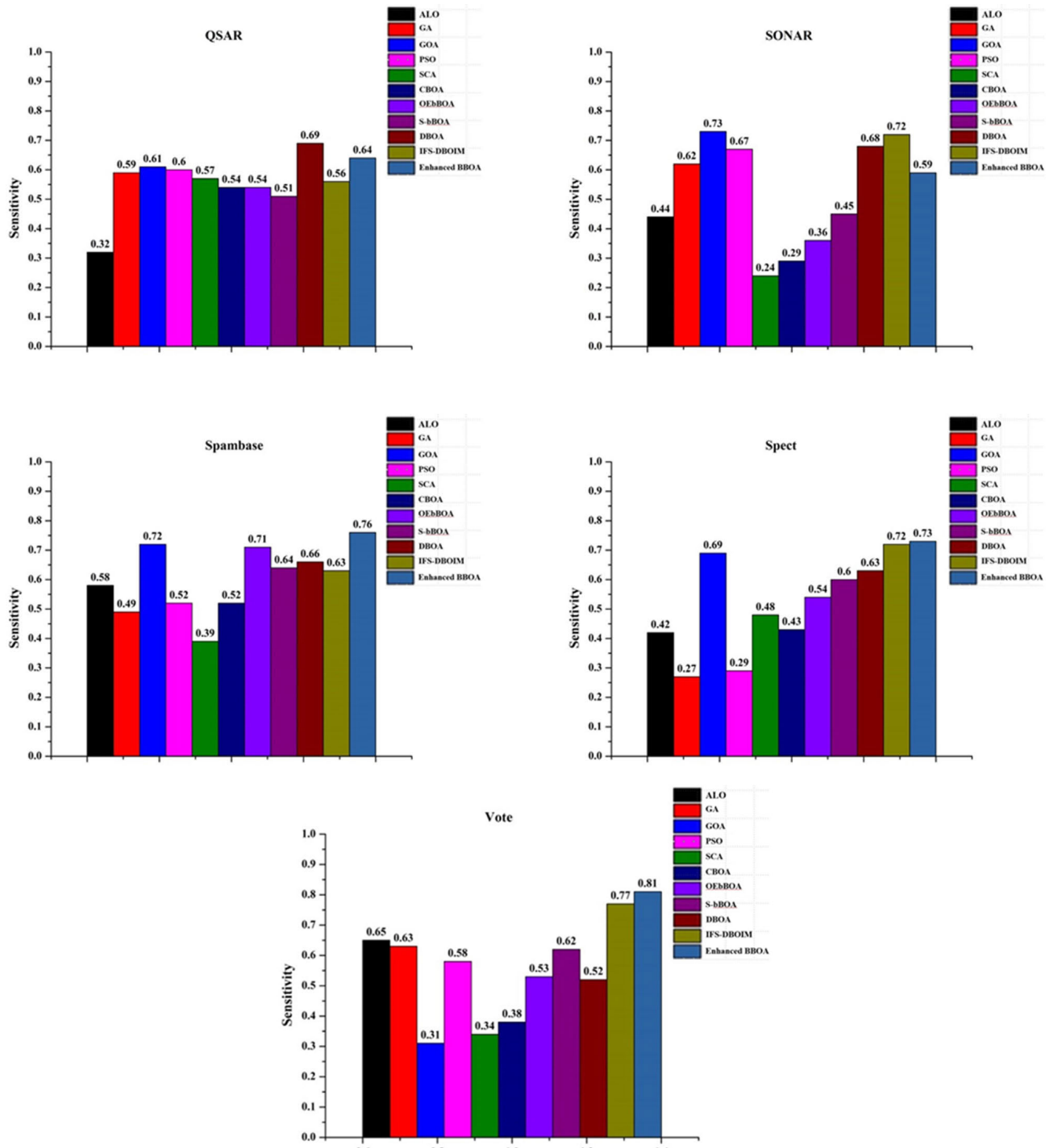


FIGURE 6. (Continued.) Sensitivity score for all thirteen binary datasets.

selection problems. Moreover, all the results are computed on twenty open-source UCI datasets and compared with eleven recently published state-of-the-art methods. The comparative analysis concludes the supremacy of our method over other

feature selection algorithms because of its ability to produce better solutions in each iteration. However, it is important to discuss the limitations of the performed work in the context of research findings, interpret the validity of the scientific

TABLE 6. Execution time comparison between state-of-the-art algorithms and the proposed Enhanced BBOA method.

| No. | ALO | BOA | GA | GOA | PSO | SCA | CBOA | S-bBOA | OEbBOA | DBOA | IFS-DBOIM | Enhanced BBOA |
|------|-------|-------|-------|-------------|-------------|-------|-------|--------|--------|-------------|-----------|---------------|
| 1 | 16.41 | 10.84 | 17.16 | 12.28 | 9.55 | 10.24 | 10.11 | 14.11 | 17.74 | 16.93 | 12.61 | 26.47 |
| 2 | 13.16 | 17.21 | 6.05 | 14.7 | 4.07 | 9.15 | 13.36 | 12.99 | 14.78 | 6.9 | 4.46 | 15.56 |
| 3 | 8.95 | 6.13 | 19.38 | 12.63 | 10.82 | 14.26 | 7.84 | 8.36 | 6.98 | 7.07 | 14.94 | 5.87 |
| 4 | 8.31 | 17.16 | 16.76 | 11.01 | 11.28 | 8.29 | 6.11 | 12.02 | 6.99 | 5.31 | 17.74 | 30.78 |
| 5 | 9.44 | 11.41 | 10.58 | 7.25 | 6.27 | 11.24 | 13.64 | 11.84 | 12.04 | 7.28 | 12.44 | 31.01 |
| 6 | 12.11 | 17.16 | 7.22 | 11.39 | 6.82 | 10.52 | 10.71 | 10.7 | 16.29 | 11.81 | 13.82 | 17.69 |
| 7 | 5.55 | 14.54 | 15.6 | 9.29 | 5.03 | 10.15 | 6.12 | 9.86 | 10.59 | 11.24 | 7.77 | 14.61 |
| 8 | 8.6 | 15.09 | 14.32 | 11.2 | 10.82 | 9.27 | 14.93 | 7.19 | 9.79 | 10.81 | 16.48 | 7.08 |
| 9 | 11.89 | 12.13 | 8.19 | 11.84 | 12.54 | 14.15 | 8.63 | 12.23 | 15.3 | 7.69 | 8.93 | 32.3 |
| 10 | 14.03 | 16.53 | 15.01 | 13.08 | 12.03 | 8.74 | 14.5 | 13.42 | 12.29 | 12.26 | 8.56 | 7.48 |
| 11 | 8.37 | 11.57 | 10.46 | 9.59 | 8.06 | 10.15 | 9.49 | 12.61 | 8.08 | 11.35 | 11.3 | 22.18 |
| 12 | 7.81 | 10 | 7.61 | 13.8 | 14.54 | 12.18 | 8.49 | 11.84 | 8.82 | 7.19 | 16.25 | 10.48 |
| 13 | 6.12 | 10.57 | 16.36 | 10.52 | 6.06 | 12.24 | 12.98 | 14.01 | 15.5 | 16.84 | 7.67 | 18.06 |
| 14 | 10.25 | 12.36 | 12.38 | 6.42 | 11.52 | 11.16 | 13.6 | 11.31 | 15.38 | 8.85 | 14.56 | 28.76 |
| 15 | 7.5 | 15.39 | 16.48 | 6.25 | 5.51 | 14.37 | 11.01 | 9.47 | 16.86 | 8.12 | 9.24 | 12.35 |
| 16 | 12.27 | 10.38 | 17.09 | 9.01 | 4.5 | 14.07 | 9.55 | 12.07 | 17.37 | 8.31 | 6.5 | 31.51 |
| 17 | 11.63 | 10.64 | 11.66 | 8.99 | 6.31 | 13.04 | 13.46 | 9.11 | 13.25 | 14.95 | 6.61 | 22.48 |
| 18 | 8.25 | 15.62 | 6.54 | 9.63 | 9.46 | 10.45 | 12.44 | 10.64 | 7.21 | 16.4 | 14.64 | 6.41 |
| 19 | 13.11 | 17.11 | 6.82 | 14.55 | 6.58 | 14.16 | 7.27 | 14.62 | 11.19 | 8.14 | 12.2 | 15.44 |
| 20 | 10.29 | 16.81 | 17.36 | 11.11 | 12.58 | 16.72 | 17.05 | 12.05 | 14.96 | 13.25 | 10.65 | 9.33 |
| Mean | 10.20 | 13.43 | 12.65 | 10.72 | 8.71 | 11.72 | 11.06 | 11.52 | 12.57 | 10.53 | 11.36 | 18.29 |
| Rank | 2 | 11 | 10 | 4 | 1 | 8 | 5 | 7 | 9 | 3 | 6 | 12 |

TABLE 7. P-value comparison between state-of-the-art algorithms and the proposed Enhanced BBOA method.

| No. | ALO | BOA | GA | GOA | PSO | SCA | CBOA | S-bBOA | OEbBOA | DBOA | IFS-DBOIM |
|------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|
| 1 | 0.03407 | 0.01374 | 0.0706 | 0.03727 | 0.06952 | 0.07719 | 0.07973 | 0.06684 | 0.08018 | 0.03856 | 0.04401 |
| 2 | 0.04533 | 0.0484 | 0.04039 | 0.04674 | 0.01153 | 0.08279 | 0.0214 | 0.06406 | 0.01693 | 0.03323 | 0.07904 |
| 3 | 0.07494 | 0.01878 | 0.06584 | 0.06865 | 0.05372 | 0.0192 | 0.03265 | 0.06958 | 0.06834 | 0.04009 | 0.02906 |
| 4 | 0.02914 | 0.07319 | 0.03246 | 0.04704 | 0.04487 | 0.05341 | 0.0232 | 0.00871 | 0.05872 | 0.01892 | 0.08275 |
| 5 | 0.01726 | 0.07813 | 0.04339 | 0.02153 | 0.03789 | 0.06924 | 0.07392 | 0.05034 | 0.07031 | 0.08496 | 0.03198 |
| 6 | 0.03528 | 0.05304 | 0.03969 | 0.0756 | 0.07322 | 0.02593 | 0.03358 | 0.05636 | 0.01675 | 0.08787 | 0.01969 |
| 7 | 0.04547 | 0.06629 | 0.0536 | 0.08893 | 0.02298 | 0.06801 | 0.02764 | 0.03318 | 0.01279 | 0.03686 | 0.05804 |
| 8 | 0.07148 | 0.07382 | 0.04746 | 0.02438 | 0.06183 | 0.08903 | 0.04719 | 0.03688 | 0.07718 | 0.03031 | 0.02048 |
| 9 | 0.04061 | 0.07043 | 0.06212 | 0.04272 | 0.05055 | 0.02487 | 0.01369 | 0.08127 | 0.07589 | 0.02181 | 0.02799 |
| 10 | 0.08415 | 0.03809 | 0.02783 | 0.08111 | 0.05956 | 0.08582 | 0.04948 | 0.05141 | 0.01992 | 0.01396 | 0.08071 |
| 11 | 0.08118 | 0.06146 | 0.01089 | 0.07555 | 0.04608 | 0.06548 | 0.07752 | 0.04867 | 0.04896 | 0.05434 | 0.08861 |
| 12 | 0.08714 | 0.03345 | 0.06337 | 0.06185 | 0.05738 | 0.04098 | 0.0351 | 0.00899 | 0.066 | 0.05564 | 0.01327 |
| 13 | 0.02916 | 0.02563 | 0.06209 | 0.0585 | 0.08036 | 0.08047 | 0.05901 | 0.01796 | 0.04633 | 0.01367 | 0.0714 |
| 14 | 0.08884 | 0.08542 | 0.02373 | 0.05964 | 0.04356 | 0.08448 | 0.05506 | 0.00824 | 0.08513 | 0.05007 | 0.02242 |
| 15 | 0.03733 | 0.07211 | 0.03164 | 0.04591 | 0.03941 | 0.04691 | 0.02053 | 0.07049 | 0.03134 | 0.0303 | 0.03104 |
| 16 | 0.07232 | 0.07239 | 0.07758 | 0.03983 | 0.08139 | 0.06722 | 0.02806 | 0.0553 | 0.08933 | 0.06438 | 0.00999 |
| 17 | 0.01883 | 0.00834 | 0.0195 | 0.06902 | 0.07315 | 0.02196 | 0.01995 | 0.07433 | 0.0542 | 0.05683 | 0.00826 |
| 18 | 0.07853 | 0.0462 | 0.06861 | 0.0324 | 0.01204 | 0.0663 | 0.06887 | 0.01105 | 0.02252 | 0.05762 | 0.02842 |
| 19 | 0.04274 | 0.06242 | 0.05768 | 0.05106 | 0.03697 | 0.03118 | 0.08533 | 0.06999 | 0.04848 | 0.02539 | 0.03847 |
| 20 | 0.03319 | 0.04764 | 0.08857 | 0.04516 | 0.06958 | 0.05269 | 0.04582 | 0.07976 | 0.06623 | 0.02672 | 0.03652 |
| Mean | 0.0523 | 0.0524 | 0.0493 | 0.0536 | 0.0512 | 0.0576 | 0.0448 | 0.0481 | 0.0527 | 0.0420 | 0.0411 |
| Rank | 7 | 8 | 5 | 10 | 6 | 11 | 3 | 4 | 9 | 2 | 1 |

work, and ascribe a credibility level to the conclusions of published research. Therefore, despite providing very robust results, we list two limitations related to our study.

1) Our method is computationally expensive because it always depends on the execution of two mathematical operators (N and β) that can maximize global classification accuracy with fewer features. It involves a

complex procedure to validate the significance of produced feature subset and corresponding classification accuracy.

2) The introduced optimization scheme lacks to determine inter-relevance between the newly selected and priorly filtered features while designing an optimal feature subset. In other words, the proposed Enhanced BBOA

may fail to maintain a tradeoff between relevancy and redundancy.

V. CONCLUSION AND FUTURE SCOPE

In this study, we proposed a hybrid variant of the BBOA by amalgamating an $A\beta HC$ -based search scheme with the exploration-exploitation mechanism of conventional BOA. It is known that the performance of conventional BBOA relies on random numbers and may result in good solution quality enhancement if a suitable butterfly position is generated. Therefore, we concentrated on improving the search strategy of BOA by iteratively producing improved solutions using: (1) N -operator (Neighborhood operator) and (2) β -operator. The significance of the generated solutions is computed in terms of producing offspring that are better than their parents (previous best solution, new solution). Here, a bi-objective fitness function is used to determine the quality of each solution after applying the $A\beta HC$ technique.

The mentioned improvement strategy is amalgamated with three popular binary variants (S-, V-, and Q-shape) of the Butterfly Optimization Algorithm (BOA), and the best possible solution is computed. The performance of our methodology is validated on twenty high-dimensional UCI datasets and compared with eleven state-of-the-art algorithms.

A comparative study shows that our methodology can be effectively used in search space optimization without compromising the major quality measures such as classification accuracy, feature reduction rate, specificity, and sensitivity.

In the future, our proposed method can be used to improve the performance of various interdisciplinary applications such as design pattern detection [40], channel selection [41], and cognitive imaging [42]. In addition, Rough Set Theory (RST) [43] can be used to investigate positive boundary regions, which may help select more relevant features from high-dimensional datasets. Recently introduced clustering-based metaheuristic algorithms such as Jellyfish Search Optimizer (JSO) [44], Red Deer Algorithm (RDA) [45], and Human Mental Search (HMS) [46] can be used with new crossover and mutation techniques like Order crossover operator (OX1), Order-based crossover operator (OX2) [47] and Position-based crossover operator (POS) [48] can also be used to obtain more robust classification results than the proposed optimization technique. A new transfer function, an X-shaped variant [49] with a parameter-independent metaheuristic algorithm such as JAYA optimization [52], can also be used to improve the performance of the proposed model.

REFERENCES

- [1] A. Tiwari and A. Chaturvedi, "A hybrid feature selection approach based on information theory and dynamic butterfly optimization algorithm for data classification," *Expert Syst. Appl.*, vol. 196, Jun. 2022, Art. no. 116621.
- [2] X.-Y. Liu, Y. Liang, S. Wang, Z.-Y. Yang, and H.-S. Ye, "A hybrid genetic algorithm with wrapper-embedded approaches for feature selection," *IEEE Access*, vol. 6, pp. 22863–22874, 2018.
- [3] J. H. Holland, "Genetic algorithms," *Sci. Amer.*, vol. 267, no. 1, pp. 66–72, Jul. 1992.
- [4] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proc. Int. Conf. Neural Netw. (ICNN)*, vol. 4, Nov. 1995, pp. 1942–1948.
- [5] C. Blum, "Ant colony optimization: Introduction and recent trends," *Phys. Life Rev.*, vol. 2, no. 4, pp. 353–373, Dec. 2005.
- [6] G.-G. Wang, S. Deb, and Z. Cui, "Monarch butterfly optimization," *Neural Comput. Appl.*, vol. 31, no. 7, pp. 1995–2014, 2019.
- [7] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey wolf optimizer," *Adv. Eng. Softw.*, vol. 69, pp. 46–61, Mar. 2014.
- [8] Y. Zhang, S. Cheng, Y. Shi, D.-W. Gong, and X. Zhao, "Cost-sensitive feature selection using two-archive multi-objective artificial bee colony algorithm," *Expert Syst. Appl.*, vol. 137, pp. 46–58, Dec. 2019.
- [9] M. A. Al-Betar, A. I. Hammouri, M. A. Awadallah, and I. A. Doush, "Binary β -hill climbing optimizer with S-shape transfer function for feature selection," *J. Ambient Intell. Hum. Comput.*, vol. 12, no. 7, pp. 7637–7665, 2021.
- [10] G. Dhiman, D. Oliva, A. Kaur, K. K. Singh, S. Vimal, A. Sharma, and K. Cengiz, "BEPO: A novel binary emperor penguin optimizer for automatic feature selection," *Knowl.-Based Syst.*, vol. 211, Jan. 2021, Art. no. 106560.
- [11] V. Kumar and A. Kaur, "Binary spotted hyena optimizer and its application to feature selection," *J. Ambient Intell. Humanized Comput.*, vol. 11, no. 7, pp. 2625–2645, Jul. 2020.
- [12] H. F. Eid, "Binary whale optimisation: An effective swarm algorithm for feature selection," *Int. J. Metaheuristics*, vol. 7, no. 1, pp. 67–79, 2018.
- [13] H. Hicheam, M. Elkamel, M. Rafik, M. T. Mesaoud, and C. Ouahiba, "A new binary grasshopper optimization algorithm for feature selection problem," *J. King Saud Univ. Comput. Inf. Sci.*, vol. 34, no. 2, pp. 316–328, Feb. 2022.
- [14] S. Mirjalili, S. M. Mirjalili, and X.-S. Yang, "Binary bat algorithm," *Neural Comput. Appl.*, vol. 25, nos. 3–4, pp. 663–681, Sep. 2014.
- [15] E. Emary, H. M. Zawbaa, and A. E. Hassanien, "Binary grey wolf optimization approaches for feature selection," *Neurocomputing*, vol. 172, pp. 371–381, Jan. 2016.
- [16] M. A. Khanesar, M. Teshnehlab, and M. A. Shoorehdeli, "A novel binary particle swarm optimization," in *Proc. Medit. Conf. Control Autom.*, Jun. 2007, pp. 1–6.
- [17] E. Rashedi, H. Nezamabadi-Pour, and S. Saryazdi, "BGSA: Binary gravitational search algorithm," *Natural Comput.*, vol. 9, no. 3, pp. 727–745, Sep. 2010.
- [18] S. Arora and S. Singh, "Butterfly optimization algorithm: A novel approach for global optimization," *Soft Comput.*, vol. 23, no. 3, pp. 715–734, Feb. 2019.
- [19] D. Karaboga and B. Basturk, "On the performance of artificial bee colony (ABC) algorithm," *Appl. Soft Comput.*, vol. 8, no. 1, pp. 687–697, Jan. 2008.
- [20] X.-S. Yang and S. Deb, "Cuckoo search: Recent advances and applications," *Neural Comput. Appl.*, vol. 24, no. 1, pp. 169–174, Jan. 2014.
- [21] S. Das and P. N. Suganthan, "Differential evolution: A survey of the state-of-the-art," *IEEE Trans. Evol. Comput.*, vol. 15, no. 1, pp. 4–31, Feb. 2011.
- [22] X.-S. Yang, "Firefly algorithm, stochastic test functions and design optimisation," *Int. J. Bio Inspired Comput.*, vol. 2, no. 2, pp. 78–84, 2010.
- [23] M. A. Al-Betar, I. Aljarah, M. A. Awadallah, H. Faris, and S. Mirjalili, "Adaptive β -hill climbing for optimization," *Soft Comput.*, vol. 23, no. 24, pp. 13489–13512, 2019.
- [24] M. A. Awadallah, A. I. Hammouri, M. A. Al-Betar, M. S. Braik, and M. A. Elaziz, "Binary horse herd optimization algorithm with crossover operators for feature selection," *Comput. Biol. Med.*, vol. 141, Feb. 2022, Art. no. 105152.
- [25] T. Fushiki, "Estimation of prediction error by using K -fold cross-validation," *Statist. Comput.*, vol. 21, no. 2, pp. 137–146, Apr. 2011.
- [26] A. Faramarzi, M. Heidarinejad, B. Stephens, and S. Mirjalili, "Equilibrium optimizer: A novel optimization algorithm," *Knowl.-Based Syst.*, vol. 191, Mar. 2020, Art. no. 105190.
- [27] C. Cortes and V. Vapnik, "Support-vector networks," *Mach. Learn.*, vol. 20, no. 3, pp. 273–297, 1995.
- [28] V. Vapnik, "Principles of risk minimization for learning theory," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 4, 1991, pp. 1–8.
- [29] R. Swinburne, "Bayes' theorem," *Revue Philosophique de la France et de l.*, vol. 194, no. 2, pp. 250–251, 2004.
- [30] J. Zhou, X. Li, and X. Shi, "Long-term prediction model of rockburst in underground openings using heuristic algorithms and support vector machines," *Saf. Sci.*, vol. 50, no. 4, pp. 629–644, Apr. 2012.

- [31] P. M. Murphy "UCI repository of machine learning databases," Dept. Inf. Comput. Sci., Univ. California Irvine, Irvine, CA, USA, 1992. [Online]. Available: <http://www.ics.uci.edu/~mllearn/MLRepository.html>
- [32] S. Mirjalili, "The Ant Lion Optimizer," *Adv. Eng. Softw.*, vol. 83, pp. 80–98, May 2015.
- [33] S. Z. Mirjalili, S. Mirjalili, S. Saremi, H. Faris, and I. Aljarah, "Grasshopper optimization algorithm for multi-objective optimization problems," *Int. J. Speech Technol.*, vol. 48, no. 4, pp. 805–820, Apr. 2018.
- [34] S. Mirjalili, "SCA: A sine cosine algorithm for solving optimization problems," *Knowl.-Based Syst.*, vol. 96, pp. 120–133, Mar. 2016.
- [35] S. Arora and S. Singh, "An improved butterfly optimization algorithm with chaos," *J. Intell. Fuzzy Syst.*, vol. 32, no. 1, pp. 1079–1088, Jan. 2017.
- [36] M. Tubishat, M. Alswaitti, S. Mirjalili, M. A. Al-Garadi, M. T. Alrashdan, and T. A. Rana, "Dynamic butterfly optimization algorithm for feature selection," *IEEE Access*, vol. 8, pp. 194303–194314, 2020.
- [37] B. Zhang, X. Yang, B. Hu, Z. Liu, and Z. Li, "OEBBOA: A novel improved binary butterfly optimization approaches with various strategies for feature selection," *IEEE Access*, vol. 8, pp. 67799–67812, 2020.
- [38] Z. Sadeghian, E. Akbari, and H. Nematzadeh, "A hybrid feature selection method based on information theory and binary butterfly optimization algorithm," *Eng. Appl. Artif. Intell.*, vol. 97, Jan. 2021, Art. no. 104079.
- [39] R. F. Woolson, "Wilcoxon signed-rank test," in *Wiley Encyclopedia of Clinical Trials*. Hoboken, NJ, USA: Wiley, 2007, pp. 1–3.
- [40] S. Chaturvedi, A. Chaturvedi, A. Tiwari, and S. Agarwal, "Design pattern detection using machine learning techniques," in *Proc. 7th Int. Conf. Rel., INFOCOM Technol. Optim. Trends Future Directions (ICRITO)*, Aug. 2018, pp. 1–6.
- [41] A. Tiwari and A. Chaturvedi, "A novel channel selection method for BCI classification using dynamic channel relevance," *IEEE Access*, vol. 9, pp. 126698–126716, 2021.
- [42] A. Tiwari, "Wilson's disease classification using higher-order Gabor tensors and various classifiers on a small and imbalanced brain MRI dataset," *Multimedia Tool. Appl.*, pp. 1–27, 2023.
- [43] Z. Pawlak, "Rough set theory and its applications to data analysis," *Cybern. Syst.*, vol. 29, no. 7, pp. 661–688, Oct. 1998.
- [44] J.-S. Chou and D.-N. Truong, "A novel metaheuristic optimizer inspired by behavior of jellyfish in ocean," *Appl. Math. Comput.*, vol. 389, Jan. 2021, Art. no. 125535.
- [45] A. M. Fathollahi-Fard, M. Hajiaghayi-Keshteli, and R. Tavakkoli-Moghaddam, "Red deer algorithm (RDA): A new nature-inspired meta-heuristic," *Soft Comput.*, vol. 24, no. 19, pp. 14637–14665, Oct. 2020.
- [46] S. J. Mousavirad and H. Ebrahimpour-Komleh, "Human mental search: A new population-based metaheuristic optimization algorithm," *Int. J. Speech Technol.*, vol. 47, no. 3, pp. 850–887, Oct. 2017.
- [47] K. Deep and H. Mebrahtu, "New variations of order crossover for traveling salesman problem," *Int. J. Comb. Optim. Probl. Inform.*, vol. 2, no. 1, pp. 2–13, 2011.
- [48] V. A. Cicirello, "Non-wrapping order crossover: An order preserving crossover operator that respects absolute position," in *Proc. 8th Annu. Conf. Genetic Evol. Comput.*, Jul. 2006, pp. 1125–1132.
- [49] A. Tiwari and A. Chaturvedi, "Automatic channel selection using multi-objective X-shaped binary butterfly algorithm for motor imagery classification," *Expert Syst. Appl.*, vol. 206, Nov. 2022, Art. no. 117757.
- [50] S. Arora and P. Anand, "Binary butterfly optimization approaches for feature selection," *Expert Syst. Appl.*, vol. 116, pp. 147–160, Feb. 2019.
- [51] A. Shahbandegan and M. Naderi, "A binary butterfly optimization algorithm for the multidimensional knapsack problem," in *Proc. 6th Iranian Conf. Signal Process. Intell. Syst. (ICSPIS)*, Dec. 2020, pp. 1–5.
- [52] A. Tiwari, "A logistic binary Jaya optimization-based channel selection scheme for motor-imagery classification in brain-computer interface," *Expert Syst. Appl.*, vol. 223, Aug. 2023, Art. no. 119921.



ANURAG TIWARI received the Ph.D. degree in computer science and engineering from the Indian Institute of Technology (BHU), Varanasi, India. He is currently an Assistant Professor with the Thapar Institute of Engineering and Technology (TIET), Patiala, Punjab, India. His research interests include computer vision, machine learning, and physiological signal processing. He has served as a reviewer for several peer-reviewed journals and conferences.

• • •