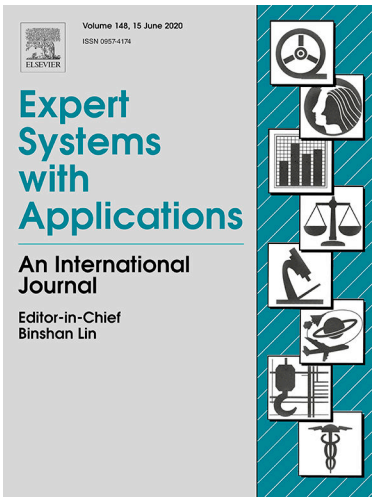
Journal Pre-proofs



A Hybrid Feature Selection Approach based on Information Theory and Dynamic Butterfly Optimization Algorithm for Data Classification

Anurag Tiwari, Amrita Chaturvedi

PII: S0957-4174(22)00111-7
DOI: https://doi.org/10.1016/j.eswa.2022.116621
Reference: ESWA 116621

To appear in: *Expert Systems with Applications*

Received Date: 9 April 2021
Revised Date: 10 November 2021
Accepted Date: 27 January 2022

Please cite this article as: Tiwari, A., Chaturvedi, A., A Hybrid Feature Selection Approach based on Information Theory and Dynamic Butterfly Optimization Algorithm for Data Classification, *Expert Systems with Applications* (2022), doi: https://doi.org/10.1016/j.eswa.2022.116621

A Hybrid Feature Selection Approach based on Information Theory and Dynamic Butterfly Optimization Algorithm for Data Classification

Anurag Tiwari[1*], Amrita Chaturvedi[2]

[1,2]Indian Institute of Technology (BHU), Varanasi, 221005, India
anuragtiwari.rs.cse17@itbhu.ac.in
amrita.cse@iitbhu.ac.in
[1,2]Department of Computer Science and Engineering, Indian Institute of Technology (BHU)

Corresponding Author: Anurag Tiwari
Mail id: anuragtiwari.rs.cse17@itbhu.ac.in
Institute Name: Indian Institute of Technology (BHU), Varanasi, India, 221005
Contact: +91-8318604979

ABSTRACT

The ubiquitous usage of feature selection in search space optimization, information retrieval, data mining, signal processing, software fault prediction, and bioinformatics is paramount to expert and intelligent systems. Most of the conventional feature selection methods implemented are based on filter and wrapper approaches that suffer from poor classification accuracy, high computational cost, and selection of irrelevant and redundant features. This is due to the limitations of the employed objective functions leading to overestimation of the feature significance. On the contrary, hybrid feature selection methods formulated from information theory and nature-inspired metaheuristic algorithms are preferred because of their high computational efficiency, scalability in avoiding redundant and less informative features, and independence from the classifier. However, these methods have three common drawbacks: (1) poor trade-off between exploration and exploitation phase, (2) getting stuck into an optimal local solution, and (3) avoiding irrelevancy and redundancy of selected features. The first and the second drawback is related to metaheuristic algorithm implementation, while the third is concerned with applied information-theoretic paradigms. To address the aforementioned problems, we developed a new hybrid feature selection method, namely, the Iterative Feature Selection using Dynamic Butterfly Optimization Algorithm based Interaction Maximization (IFS-DBOIM) that combines Dynamic Butterfly Optimization Algorithm (DBOA) with a mutual information-based Feature Interaction Maximization (FIM) scheme for selecting the optimal feature subset. There is evidence that DBOA performs better in exploration, exploitation, and avoidance of local optima entrapment, and FIM comparatively scores the maximum relevancy with minimum redundancy of the new features with previously selected ones. The performance of the proposed method is compared using twenty publicly available datasets with ten baseline feature selection approaches. The results revealed that IFS-DBOIM outperforms other approaches on most datasets, maximizing the percent classification accuracy with the least number of features. The nonparametric Wilcoxon rank test confirms the statistical significance of these outcomes. Moreover, this method realizes the best trade-off between accuracy and stability.

Keywords: Feature selection; Dynamic butterfly optimization algorithm; Feature interaction maximization; Mutual information; Classification accuracy

1. Introduction

In the growing era of the information industry, data size and features have enormously increased, whereas the patterns are relatively few, which results in relevant, redundant, and irrelevant information of the features. Such redundant and irrelevant features increase the computational burden and bring the "curse of dimensionality" (Verleysen et al., 2005), which refers to those phenomena that arise during data analysis in high dimensional space. Dimensionally cursed phenomena occur in numerical analysis, sampling, combinatorics, machine learning, data mining, and databases. The common theme of these problems is that when the dimensionality increases, the volume of the space increases so fast that the available data becomes sparse. The main objective for reducing the dimensionality of the data and selecting the informative features is to alleviate the training time and improve the classification accuracy of the algorithm. In many research areas, feature selection techniques play an important role in eliminating irrelevant and redundant features from the original set. Therefore, efficient feature selection methods can effectively project the high-dimensional data into relatively lower-dimensional space.

Feature selection methods can be broadly categorized into three main classes: (1) filter, (2) wrapper, (3) hybrid or embedded methods (Zhang et al., 2019). The filter methods employ statistical data-dependent approaches to determine an optimal feature set for classification (Ghosh et al., 2019). These methods are independent of the classifier type and are computationally fast; however, they ignore the relevance of the various dimensions while determining the optimal feature subset (Bommert et al., 2020). They utilize various information-theoretic concepts such as Mutual Information (MI), entropy, Information Gain (IG), and trace rate. Also, they explore different distance and correlation measures to find the relationship between the dimensions. On the other hand, the wrapper methods employ machine learning algorithms to search for the most optimal solution from a set of feasible solutions. These methods rely on the predictive ability of the applied classifier to estimate the quality of the selected features. Wrapper methods are more effective, but they are more computationally expensive than filter methods because of the involvement of a classification process (Hu et al., 2015). Finally, the hybrid methods inherit the merits of both filter and wrapper methods in two ways- (1) they involve the interaction between selected features and a learning algorithm, and (2) they are capable of determining dependencies with a lower computational cost than the wrapper methods

as they do not evaluate the optimal feature set iteratively. These methods are implemented by algorithms with built-in feature selection methods (Çavuşoğlu et al., 2019).

A large family of hybrid feature selection algorithms contains information-theoretic paradigms to measure the importance of the features. A popular information theory measure, Mutual Information (MI), has effectively evaluated the statistical dependence between two variables in recent studies. For example, Mutual Information-based Feature Selection (MIFS) was introduced by Battiti (1994) for considering the relevancy of features. Peng et al. (2005) proposed the Minimal-Redundancy Maximal-Relevance (MRMR) criterion that determines the relationship between the feature redundancy parameter and the number of selected features. The MRMR and MIFS effectively compute the redundancy between the features but are limited by estimating too much redundancy. Bennasar et al. (2013) introduced a new information-theoretic concept, namely, Feature Interaction Maximization (FIM), which employed three-way interaction information to find the feature redundancy. Validation results on three different datasets proved its superiority over Information Gain (IG) (Thangaiah et al., 2009), Minimal-Redundancy Maximal-Relevance (MRMR), and Interaction Gain based Feature Selection (IGFS) (El Akadi et al., 2008) methods.

Grasshopper optimization algorithm (GOA) is a bio-inspired swarm intelligence algorithm that imitates the social interaction behavior of the grasshopper (Mirjalili et al., 2018). GOA has produced encouraging results in solving different optimization problems, such as feature selection (Aljarah et al., 2018), financial stress prediction (Luo et al., 2018), and task scheduling (Xu et al., 2020). Wang et al. (2019) developed an efficient and fast feature selection model using the migratory behavior of a special kind of butterfly that is native to North America, termed as Monarch Butterfly Optimization (MBO) algorithm and wrapper method that uses K-nearest neighbors (K-NN) classifier. The MBO algorithm achieved an average of 93% classification accuracy, superior to the Genetic Algorithm (Holland, 1992) and Particle Swarm Optimization (Kennedy et al., 1995). In a similar work, Arora and Singh (2019) developed a new nature-inspired metaheuristic algorithm for global optimization named Butterfly Optimization algorithm (BOA), which is inspired by the foraging behavior of the butterflies. The behavior of butterflies can be described as their cooperative movement toward the food source position. Simulation results on benchmark functions indicate that the global search ability of BOA is better than MBO in terms of optimization accuracy and convergence speed (Arora and Singh, 2019).

In this paper, we chose BOA over MBO for two main reasons: (1) compared to BOA, the MBO depends on two migratory operators to determine the exact direction of butterflies. In order to maintain a unique migration strategy, MBO adjusts the ratio between both operators and then fixes the pathway of butterflies. This process increases the overall complexity of the MBO algorithm and reduces the convergence speed of the feature selection algorithm, and (2) the search strategy of the original MBO easily falls into local optima; therefore, its ability to find the best solution in large search space is questionable. In addition, BOA has demonstrated its ability to solve various engineering problems by constraint handling effectively, such as spring design, welded beam, and gear train design, with competitive results compared to other optimization algorithms. However, the original BOA algorithm has the problem of lacking solution diversity during the optimization process.

Recently, numerous hybrid learning approaches have been introduced to improve the performance of different metaheuristic algorithms. For example, Chen et al. (2018) proposed a metaheuristic scheme based on an Adaptive Large Neighborhood Search (ALNS) algorithm to solve the Dynamic Vehicle Routing Problem (DVRP) with the limited number of vehicles and

hard-time windows. This approach involves ad hoc destroy/repair heuristics and a periodic perturbation procedure. Arora and Anand (2019) employed S and V-shaped transfer functions to improve the solution diversity of traditional BOA. The experimental results confirmed the effectiveness of the proposed approach in improving the solution diversity and classification accuracy of the original BOA. Zhang et al. (2019) presented a two-phase feature selection method by combining mutual information-based search strategy and Particle Swarm Optimization (PSO). In the first phase of this work, an average information-based filtering method was applied to reduce the search space, while a local search strategy enhanced the swarm's exploitation capability in the second phase.

Tubishat et al. (2020) proposed a Dynamic Butterfly Optimization Algorithm (DBOA) as an improved variant of the BOA. They introduced a Local Search Algorithm based on Mutation (LSAM) scheme to improve solution diversity. The results demonstrated that DBOA significantly improved the solution diversity compared to the S-shaped variant of BOA. Wang et al. (2021) improved the monitoring efficiency of mobile wireless nodes using Distributed Guidance Antiflocking Algorithm (DGAA) (Ganganath et al., 2015). This approach mimics the predatory behavior of solitary animals to find the redundant areas using multiple wireless sensors. Simulation results showed that the improved DGAA has better dynamic inclusion in monitoring the continuous movement of sensors and a better convergence rate than the original DGAA and random search algorithm (Bergstra et al., 2012). Li et al. (2021) proposed a novel dynamic learning strategy to maintain the diversity and convergence of the distribution of solutions over the Pareto Front (PF). This method continuously changes the direction of learning: convergence and diversity with each iteration by applying crossover and mutation operations. The experimental results showed that the proposed learning scheme effectively improved the quality of approximations on the PF obtained from multiobjective optimization algorithms.

Zhang et al. (2021) proposed a stochastic multiobjective Nondominated Sorting Genetic Algorithm (NSGA)-III as a dynamic variant of the conventional NSGA-III algorithm (Yi et al., 2018). In this algorithm, two approaches, second-order difference, and random search strategy, were applied with the NSGA-III scheme, which obtained a good convergence rate and maintained the populations' diversity. Li et al. (2021) developed a hybrid optimization algorithm by combining a new coevolutionary scheme with the dynamic learning strategy. The coevolutionary scheme improved the quality of individuals in a population through cooperation between multiple populations (or subpopulations). Further, the improved solutions were used to detect the dynamic behavior of each evolutionary state when used with the dynamic learning strategy. Li and Wang (2021) combined the dynamic topology and the Biogeography Based Optimization (BBO) (Simon et al., 2008) with Elephant Herdling Optimization (EHO) (Wang et al., 2015). They named it the Biogeography-based Learning Elephant Herdling Optimization (BLEHO). In this hybrid approach, they changed the topological structure of the population by dynamically changing the number of instances available in the original population. For updating each instance, they used the revised operator based on either the BBO or the EHO. Finally, through elitism strategy, a certain number of better individuals were preserved directly for the next generation without being processed, thus ensuring a better convergence for the population. Sadeghian et al. (2021) developed a feature ranking model by combining S-shaped BOA with information gain. The proposed hybrid model achieved competitive classification results compared to ReliefF (Kira and Rendell, 1992), Fisher Score (Gu et al., 2012), Minimal Redundancy-Maximal New Classification Information (MR-MNCI) algorithm (Gao et al., 2018). However, this approach suffers from a poor convergence rate because the gradient for input far from the origin is near zero, thus slowing gradient-based learning for saturated data instances.

In this study, we extend the work of Sadeghian et al. (2021) by combining Dynamic Butterfly Optimization Algorithm (DBOA) with an MI-based three-way interaction mechanism (Bennasar et al., 2013). Compared to the aforementioned metaheuristic algorithms, DBOA has two major advantages: (1) local minima avoidance, (2) maintaining high solution diversity while finding the optimal solution. The DBOA approach uses a Local Search Algorithm Based on Mutation (LSAM) operator to overcome the local optima problem and improve solution diversity. Researchers showed that the adaptive mechanism of the DBOA is capable of increasing the convergence rate with fewer iterations. It helps to maintain a trade-off between the exploration and exploitation phase to compute an accurate global solution in feature selection problems. Therefore, this approach demonstrated impressive results compared to baseline feature selection algorithms. These improvements motivated us to use DBOA in our work. Despite these advantages, the DBOA does not consider the relevance of the features and the class label in the feature selection process. Henceforth, a feature interaction-based FIM scheme is also merged with the DBOA to measure the significance of the selected features.

This paper introduces a novel hybrid approach called the Iterative Feature Selection using Dynamic Butterfly Optimization-based Interaction Maximization (IFS-DBOIM) algorithm to undertake feature selection and classification problems. Our method combines the FIM method with DBOA for selecting the optimal feature subset and maximizing classification accuracy. We validated our proposed feature selection method on twenty publicly open datasets using three classifiers: (1) Support Vector Machine (SVM), (2) Naïve Bayes (NB) algorithm, and (3) Decision Tree (DT). Also, a five-fold cross-validation approach is used to quantify classification results statistically to split the global population into the training and evaluation set. Experimental results show that the proposed IFS-DBOIM algorithm consistently outperforms other state-of-the-art feature selection methods with less computational complexity.

The contributions of our method are recapitulated as follows:

1. Maximizing relevance between a set of features and class labels to improve the classification accuracy of the proposed model
2. Using a tri-objective function to determine the quality of each solution generated by the dynamic butterfly optimization, the objectives are: (1) classification accuracy maximization, (2) minimizing the size of optimal feature subset, and (3) mean FIM score
3. Understanding the compatibility of the proposed feature selection method with the set of classifiers mentioned above.

The rest of this paper is structured as follows: Section 2 discusses the preliminaries of both DBOA and the three-way interaction-based information maximization scheme**.** In Section 3, the proposed IFS-DBOIM approach is explained. The experimental results and discussion are provided in Section 4. Finally, conclusions and the future scope of the work are presented in Section 5.

2. Preliminaries

This section explains the basic concepts used within our study, namely, Dynamic Butterfly Optimization Algorithm (DBOA) and Feature Interaction Maximization (FIM). In the first subsection, we cover the basic search strategy used by

butterflies and the solution quality improvement scheme introduced in the DBOA. Similarly, a three-way feature interaction approach for determining the relationship between previously and newly selected features is presented in the second subsection.

2.1. Dynamic butterfly optimization algorithm

Dynamic Butterfly Optimization Algorithm (DBOA) is an improved variant of the conventional Butterfly Optimization Algorithm (BOA). Previously, DBOA has shown its ability to solve different high dimensionality optimization problems such as stagnation into local minima and lacking solution diversity during the optimization process. The DBOA mimics the food foraging and partner-mating strategy using the fragrance of the food or the partner. The entire concept of sensing and food search processing depends on three essential factors, namely, sensory modality (*c*), stimulus intensity (*I*), and power exponent (*a*). In the DBOA, *I* correlates with the fitness value and the magnitude of fragrance (*f*) of the butterfly. Mathematically, the fragrance can be formulated as a function of stimulus intensity, sensory modality, and power exponent as follows:

$$f = cI^a \tag{1}$$

In our experiment, we consider *a* and *c* in the range of [0, 1]. For $a = 1$, the maximum fitness value or stimulus intensity is obtained, while $a = 0$ shows the minimum fragrance value that can not be sensed by any butterfly. Therefore, parameter a controls the nature of the butterfly optimization algorithm. Another parameter is *c* which determines the convergence speed of the algorithm. Theoretically, *c* lies in the range of [0, ∞] but is practically determined by the system's behavior to be optimized.

2.1.1. Food search strategy of butterflies

There are three phases in the DBOA implementation: (1) population initialization, (2) best solution computation using local search algorithm, and (3) termination. In each run, first, each instance of a given population is initialized, then searching for the most optimal solution is executed iteratively, and in the final phase, the DBOA is terminated with the best solution. The algorithm defines the solution space, an objective function, and parameter values in the initial phase of the problem. After determining all constraints, the algorithm creates the butterflies' initial population for optimization, with their fragrance and fitness value computed and stored. Now, the initialization phase is over, and the algorithm enters into the iteration phase, where the positions of butterflies are updated, and artificial butterflies are created. In this phase, the algorithm is executed several times, and all butterflies in the solution space move to the updated positions, and their fitness values are computed. Then these butterflies generate fragrance at their positions using Eq. 1. In the next phase, two key steps are defined for updating the butterfly position, i.e., exploration and exploitation. In the exploration phase, a butterfly converges to the globally fittest solution g^* according to Eq. 2.

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

where x_i^t is the solution vector x_i for the i^{th} butterfly in iteration number t. Here, g^* represents the best solution for the current iteration, r is a random number that lies in the range of [0,1], and f_i is the fragrance of i^{th} butterfly calculated in the same iteration. Similarly, the exploitation phase can be defined in terms of a local search strategy to choose the best solution according to Eq. 3**.**

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

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 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 the stopping criteria are matched. The pseudocode of the DBOA is discussed in Algorithm 1.

2.1.2. Solutions quality improvement strategy

DBOA employs a Local Search Algorithm based on Mutation (LSAM) operator to improve the quality of either the current best solution (g^*) or other solutions computed by the fitness function. LSAM receives the current best solution at the end of each iteration and updates it using the mutation operator. If the new mutated solution is better than the existing solution, then the new solution will replace the existing solution (g^*). Otherwise, LSAM selects a random solution from other solutions and performs the same mutation operation. If the fitness value of the mutated solution is better than the fitness value of the selected solution, then the process is repeated by *Num_iterations (N)* times. Finally, the computed solution is assigned as g^* for further processing. The pseudocode of the LSAM working is given in Algorithm 2.

Algorithm 1**:** General Pseudocode of DBOA

- Initialize $\boldsymbol{n}$ butterflies population positions $\boldsymbol{x_i}$ $(i = 1,2, \dots, n)$
- Set the initial value of parameters (switching probability $\boldsymbol{\rho}$, sensory modality $\boldsymbol{c}$, power exponent $\boldsymbol{a}$, and the number of iterations N)

```
while not reach N do
    for each butterfly bf in the population do
        Compute the fragrance value f for each bf using Eq. 1
    end for
    Find the best butterfly bf
    Assign the best butterfly to g*
    for each butterfly bf in the population do
        Generate a random value r over the interval [0,1]
        if (r < ρ)
            Update bf position by using Eq. 2        ---- (Exploration phase)
            else
            Update bf position by using Eq. 3        ---- (Exploitation phase)
        end if
        Evaluate the new butterfly
        If the new butterfly is better, update it in the population
    end for
    Update the value of the power exponent, and variable c
    Apply LSAM on the current best solution using the mutation operator (to improve the quality of solutions using algorithm 2)
    Update the best global solution if find the better solution
end while
```

21.	Return the best solution found by the DBOA

2.2. Feature interaction maximization

In information theory, the term interaction is mostly used to indicate the amount of information shared by a set of variables. It is a useful measure as it does not consider any prior assumptions about variables and can effectively deal with the nonlinear dependency between them. In Shannon's information theory, Variable Entropy (VE) and Mutual Information (MI) are two important measures to compute information from a set of features. Entropy is a quantitative measure to show the uncertainty of a random variable. suppose $X = \{x_1, x_2, \dots, x_n\}$ is a discrete random variable and $Y = \{y_1, y_2, \dots, y_m\}$ is a class label. If the probability density function is denoted by p(x) and
p(x) = Probability(X = x), then the entropy of X will be given by:

$$H(X) = - \sum_{i=1}^{N} p(x_i) * \log(p(x_i)) \tag{4}$$

where $0 \leq H(X) \leq 1$. The joint entropy $H(X,Y)$ of two discrete random variables X and Y is defined as:

$$H(X,Y) = -\sum_{i=1}^{N} \sum_{j=1}^{M} p(x_i, y_i) \log(p(x_i, y_i)) \tag{5}$$

If one of the two variables is known and the other is not, the remaining uncertainty is termed conditional entropy, and it is defined as:

$$H(X,Y) = -\sum_{i=1}^{N} \sum_{j=1}^{M} p(x_i, y_i) \log(p(x_i / y_i)) \tag{6}$$

The amount of the information shared by both variables is termed mutual information, which can be computed by

$$I(X,Y) = \sum_{i=1}^{N} \sum_{j=1}^{M} p(x_i, y_i) \log \frac{(p(x_i,\ y_i))}{p(x_i) * p(y_i)} \tag{7}$$

The value of MI given by the $I(X,Y)$ is always positive. A high MI value refers to significant associations between both variables; MI is zero if both variables are independent. In the feature selection, the MI can find the relationship between a variable X and a target class Y. This relation is also termed information gain, and a feature with higher mutual information is considered the most relevant to the target feature. The measures of information theory that can be derived from MI and used in our study are conditional mutual information $I(X_j;\ Y/X_i)$ and three-way interaction method $(X_j;\ X_i;\ Y)$. In both cases, the relation between a feature and target class is studied in the context of other features and can be computed from Eq. 8 and Eq. 9, respectively.

$$I\left(X_j; Y/X_i\right) = H\left(X_j;\ Y\right) - H\left(X_j / Y, X_i\right) \tag{8}$$

$$I\left(X_j;\ X_i;\ Y\right) = I\left(X_j, X_i;\ Y\right) - I\left(X_j; Y\right) - I(X_i;\ Y) \tag{9}$$

Unlike MI, three-way interaction measures can be either positive, negative, or zero. A positive score refers to the combined information associated with two features that can not be provided by each of them individually. It is negative when any of the

two features can compute the combined information. A zero score shows that both features are independent and don't share any information. Suppose X_i is a candidate feature and X_s is a feature belonging to the subset S (i.e., it has already been selected), and C is a target class (attribute), feature interaction maximization (FIM) can be defined as

$$FIM = \arg\max(I\ (X_i; C) + \min_{X_s \in S}(I(X_i; X_s; C)) \quad (10)$$

where

$$I(x_i; X_x; C) = I(x_i; X_s; C) - I(X_i; C) - I(X_s; C) \quad (11)$$

In Eq. 10, the mutual information between feature X_i and C computes the relationship between the candidate feature and the class attribute. The interaction information among X_i, X_s, and C is the redundancy term. The feature which is selected is the one that maximizes the objective function defined in Eq. 10. It has the maximum relevance to the class attribute and the minimum interaction with the selected features. The advantage of this criterion is its ability to select the features that have the highest discriminative power. The pseudocode of the three-way interaction maximization approach is given in "Algorithm 3". Our main objective is to estimate a subset of those features that have maximum interaction with the candidate feature and some resemblance to those that are already present in the subset. This process is iteratively performed for all possible solutions until a subset with a maximum interaction score is determined for further processing.

Algorithm 2**:** General Pseudocode of LSAM Working

- Current_Best = **g*** (represents the best solution computed at the end of current iteration by BOA)
- Old_fitness = fitness (**g***) … (fitness value of the best solution)
- ***K*** =1, Num_iterations (nm) = ***N***, Mutation_rate (mu) = 0.1

```
while (K< N)
    New_Position = Mutate (Current_Best, Mutation_Rate)
    New_Fitness = fitness (New_Position)
        if New _Fitness < Old _fitness
            g* = New_Position
            Old_fitness = New_Fitness
            Current_Best = g*
        else
            P_Selected = Select a search_agent randomly from other population elements
            P_fitness = fitnesss (P_Selected)
            if New_Fitness < P_fitness
                P_Selected = New_Position
            end if
        end if
    K = K+1
end while
Return g*
```

Algorithm 3: Pseudocode of feature interaction maximization scheme

- **Initialization phase:**

1. Set $X \leftarrow$ "initial set of n features," $S \leftarrow$ "empty set."
2. (Computation of the MI with the output class) For each feature $x_i \in X$ $I(C; x_i)$.
3. Find the first feature x that maximizes $I(C; x_i)$; set $X \leftarrow X \setminus \{x_i\}$; set $S \leftarrow \{x_i\}$.

- **Greedy selection phase**

1. Repeat until $|S| \leftarrow k$;

2. (Computation of the interaction information between variables) For all pairs of variables $I(x_i; X_x; C)$ with $x_i \in X$, $x_s \in S$, compute $I(x_i; X_x; C)$ if it is not already available.

3. (Selection of next feature) Choose feature x_i as the one that maximizes **Eq. 10.**

- **Output the set S with the selected features.**

2.3. Classification performance

To evaluate the performance of the proposed IFS-DBOIM method, three classification algorithms: (1) Support Vector Machine (SVM), (2) Naïve Bayes (NB) method, and (3) Decision Tree (DT) are used, and the results are recorded. A short discussion on these algorithms is given below.

i. **Support Vector Machine (SVM):** Support Vector Machine (SVM) is one of the most robust classification models based on statistical learning paradigms proposed by Vapnik and Vapnik (1999). The primary objective of the SVM algorithm is to select a hyperplane in an N-dimensional feature space that can effectively discriminate among the observations of different class labels. Therefore, an SVM classification model maps training samples to a set of data points in space to maximize the gap between the two categories. The SVM follows the structural risk minimization principle (Vapnik et al., 1992), where the fundamental goal is to find the most optimal hyperplane that guarantees minimum true error. The true error shows the probability that a classifier will wrongly classify a new or unseen test observation. SVM implementation is relatively efficient, and, unlike the Artificial Neural Networks (ANNs), it does not require a large dataset for training purposes. Also, an SVM model effectively deals with high-dimensional datasets without increasing spatial complexity.

ii. **Naïve Bayes Methods:** Naïve Bayes (NB) classifiers belong to a family of probabilistic classifiers that work on the Bayes Theorem (Ch et al., 1997**).** 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 DT approaches. Naïve Bayes classifiers are fast, memory efficient, and immune to overfitting, making them a robust classification approach to deal with noisy data samples.

iii. **Decision Tree Classifiers:** Decision Trees (DT) are rule-based classifiers that predict the class label using different "if-else" rules (Yin et al., 2003). These rules are very flexible and easy to implement while handling high-dimensional datasets; therefore, these classifiers are often used to design descriptive classification models. However, DT classifiers do not involve any domain knowledge or specific parameter tuning procedure, making them useful for dealing with different dataset

categories. During classification, it decomposes a dataset into multiple small subsets, and at the same time, an associated decision tree is hierarchically developed using the set of association rules.

3. Proposed methodology

The proposed IFS-DBOIM model is implemented in two phases: (i) the training phase using the DBOA, (ii) the feature selection phase. In the first phase, DBOA is employed iteratively to determine the diversity associated with training samples. Then, feature interaction maximization is adopted in the second stage to select the best feature subset. Most feature selection methods opt for the fitness function with two contradictory objectives: (1) maximum classification accuracy rate, (2) relative number of selected relevant features. Here, the relative number of selected features is derived by dividing the number of selected features by the total number of available features. Therefore, the proposed work defines a multiobjective fitness function for the most optimal feature set with the minimum number of features. The general structure of the proposed feature selection scheme is displayed in Fig. 1.

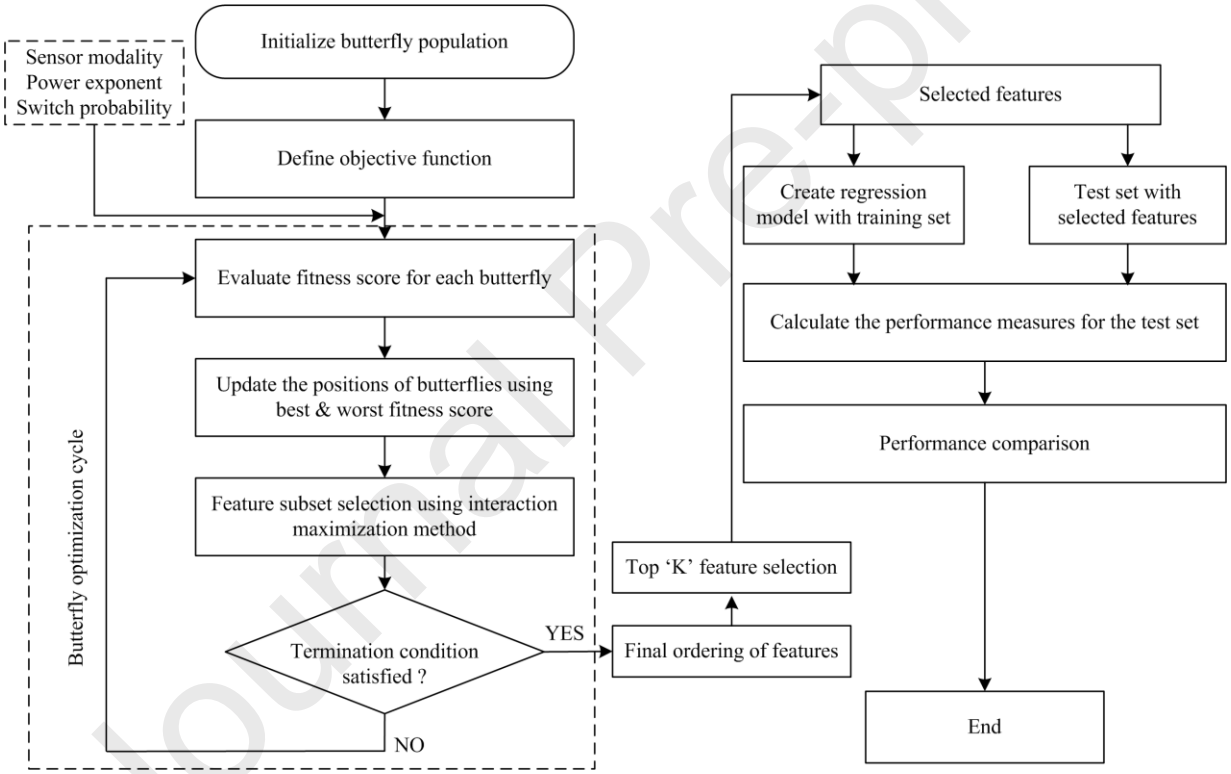


Fig. 1. Block diagram of the proposed IFS-DBOIM method

3.1. Dynamic butterfly optimization-based interaction maximization algorithm

The DBOA is a practical approach to cover the dynamics of the different butterflies' positions. Researchers have shown that the relevance of the selected features with class labels in each solution can be an effective objective that helps to obtain optimal solutions with higher classification accuracy. Therefore, in this work, we use a three-objective fitness function to evaluate the quality of each solution. The higher the classification accuracy rate, the lesser the relative number of selected

features, and the higher average feature interaction score in the solution leads to a better solution. Firstly, all three objective functions are individually computed and then assigned to the three-objective fitness function defined in Eq. 12. In the proposed IFS-DBOIM algorithm, each solution is evaluated according to the proposed fitness function, and the solution with the highest fitness score is assigned as the best solution.

$$Fitness\_function = f(f_1, f_2, f_3) = w_1 f_1 + w_2 f_2 + w_3 f_3 \quad (12)$$

where f_1 is the classification accuracy rate, f_2 is the relative number of selected relevant features, and f_3 is the average feature interaction score computed in Eq. 10. The sum of all the coefficients equals 1. Second, the interaction maximization approach selects the subset of best interactive features from each solution obtained in the first phase. Since all the feature subset solutions can not be implemented simultaneously in the classification process, they are arranged in the decreasing order of their overall interaction score. Finally, the feature subset with the maximum interaction score is used in the classification process. The pseudocode of the proposed IFS-DBOIM algorithm is given in algorithm 4.

Algorithm 4**:** General Pseudocode of IFS-DBOIM algorithm

```
Objective function f(x), x = (x_1, x_2, ..., x_d);
Initialize n butterflies population positions x_i (i = 1,2, ..., n)
Set the initial value of parameters (switching probability: ρ, sensory modality: c, power exponent: a)
while the stopping criteria are not met, do
    for each butterfly in the population do
    Calculate the fragrance using Eq. 1;
    end for
    Find the best butterfly by using Eq. 12;
    for each butterfly in the population do
        Generate a random number rand from [0,1];
        if rand < ρ
            Move towards the best butterfly using Eq. 2;
        else
            Move randomly using Eq. 3;
        end if
    end for
Update the value of the power exponent, and variable c
Apply LSAM on the current best solution using the mutation operator
Evaluate the new butterfly using Eq. 12
Update the best global solution and find the better solution
end while
Return the best solution found by the DBOA algorithm
```

3.2. Complexity analysis of IFS-DBOIM

The computational complexity of the IFS-DBOIM approach is estimated as a performance indicator that mainly depends on three steps: (1) position update using the DBOA, (2) selecting the best features using the FIM method, and (3) classifier training time. Therefore, the complexity can be mathematically represented as $O(C_{DBOA} + C_{FIM} + C_{cl})$ where O denotes worst-case time complexity, C_{DBOA}, C_{FIM}, and C_{cl} indicate the complexity of DBOA implementation while modifying the location of each instance, the FIM method, and the execution time of the classifier in the training phase, respectively. Here, the overall complexity is represented in terms of the number of iterations (K), population size (N), dimension size (d), and training time (T_t) and can be calculated as $O\ (K\ \{(N + K) + d\} + T_t)$ where $K(N + K)$ denotes C_{DBOA}, Kd is C_{FIM}, and T_t is classifier training time (C_{cl}). It is deduced from the above expression that the complexity of the IFS-DBOIM method mainly depends on the number of iterations and the population size. In the case of $K \gg N$, the total complexity will be $O\ (\{K^2 + dK\} + T_t)$ but if $N \gg K$, it will become $O\ (N + T_t)$.

4. Experimental results and discussion

In this section, the performance of the proposed model is validated on twenty benchmark datasets taken from the University of California Irvine (UCI) repository (Murphy et al., 1994) and compared with ten state-of-the-art optimization algorithms. This section is organized as follows: experimental setup and dataset description are given in subsection 4.1 and 4.2, respectively**.** Subsection 4.3 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 4.4.

4.1. Dataset description

Twenty benchmark datasets were taken from the UCI repository for classification to validate the proposed method's performance. The main motive behind selecting the twenty datasets from the UCI repository is that they are high dimensional and encompass various research domains. The details of selected datasets are given in Table 1 as the number of classes, 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 but 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™ i7 Processor, 4.2 GHz CPU frequency, 8 GB memory, and 1 TB secondary storage with a Windows 10 operating system.

Table 1. Categorical distribution of UCI datasets used in this study.

Sr. number	Dataset	Number of features (D)	Number of samples (S)	Number of classes
1	Australian	14	690	2
2	Credit	20	1000	2
3	CTG	22	2126	3
4	Exactly	13	1000	2
5	Diabetic	20	1151	2
6	Hill Vally	100	606	2

7	Ionosphere	34	351	2
8	Libras	90	360	15
9	M-of-N	13	1000	2
10	OBS-Network	21	1075	4
11	Penglungew	325	73	2
12	QSAR	41	1055	2
13	Sonar	60	208	2
14	Spambase	57	4601	2
15	Spect	22	267	2
16	TOX-171	5748	171	4
17	Vote	16	300	2
18	Vowel	13	990	10
19	Waveform	21	5000	3
20	Yale	1024	165	15
Mean	-	383.70	-	-

4.2. Experimental setup

In our work, a five-fold cross-validation scheme is applied to each dataset to test the effectiveness of the machine learning model 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 the interval of 0 and 1. To quantify results statistically, each fold is repeated 30 times, and every experiment is continually performed 100 times, giving a total of 15000 runs for each dataset. Next, the predictive classification model is developed on the training data and validated on the testing data, and 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 algorithms and the proposed algorithms are given in Table 2.

Table 2. The setting of algorithm-specific parameters

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 (Mirjalili & Seyedali, 2015)
SCA	*a*- Power exponent = 2 as in the original article (Mirjalili & Seyedali, 2016)
BOA	*a*- Power exponent = 0.1 as in the original article (Arora & Singh, 2019)
CBOA	Control parameter (*P*) = 0.5, Chaotic numbers ∈ (0,1), Constant (*b*) = 0.2 These parameters are listed in original article (Arora & Singh, 2017)
DBOA	*a*- Power exponent = 0.1 as in (Arora & Singh, 2019), Number of runs (nm) = *M* = 30, Mutation_ratio (mu) = 0.1
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 parameter) = 0.7.These parameters are listed in the original article (Zhang et al., 2020).
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]. These parameters are listed in original article (Arora & Singh, 2019)
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, $\gamma = 100$)

4.3. Performance measuring criteria

Performance assessment is one of the essential steps to show the effectiveness of the proposed method. A singular performance metric may become biased for one dataset and may also yield poor results for others. Therefore, five different

performance metrics: (1) classification accuracy, (2) feature reduction rate, (3) fitness score, (4) sensitivity, and (5) specificity are used to evaluate the performance of the proposed model. The fitness score is already computed using Eq. 12. In addition, to ensure the significance of improved classification accuracy, the nonparametric Wilcoxon test (Gehan et al., 1965) compared the *p*-value with the published results. If the *p*-value is less than 0.05, then the results obtained by the IFS-DBOIM are considered statistically significant. The details of performance metrics are given below.

1) Classification Accuracy (*CA*): It is an important measure that describes a classifier's ability to discriminate between different samples using a selected optimal feature set. The classification accuracy can be calculated using Eq. 13.

$$CA = \frac{1}{M}\sum_{i=1}^{M}\frac{1}{N}\sum_{j=1}^{N} match\ (C_j, L_j) \tag{13}$$

where M is the total number of iterations the algorithm has been 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 (*FRR*): Feature reduction rate is an indicator that shows how effectively the irrelevant and redundant features are iteratively eliminated from the original feature set. Therefore, the algorithm with the minimum number of selected features has a high feature reduction rate compared to other methods for a given dataset. It is an important measure that reduces the computational complexity of the IFS-DBOIM method without compromising the classification accuracy. It is computed using Eq. 14.

$$FRR = \ 1 - \frac{total\ number\ of\ selected\ features}{total\ number\ of\ original\ features} \tag{14}$$

3) Sensitivity: It indicates the proportion of actual positive cases that got predicted as positive (or true positive). This implies that there will be another proportion of actual positive cases, which would get predicted incorrectly as negative (false negative). Mathematically, it is defined in Eq. 15.

$$Sensitivity = \frac{TP}{TP+FN} \tag{15}$$

4) 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:

$$Specificity = \frac{TN}{TN+FP} \tag{16}$$

where TP is True_Positive, TN is True_ Negative, *FP* is False_Positive, and *FN* is False_Negative samples detected during classification.

4.4. Results comparison and discussion

In this subsection, two groups of experiments are conducted for evaluating the performance of the proposed IFS-DBOIM method. The main reason for categorizing the experiments is to understand the comparative performance of two different sets

of optimization algorithms. The first set contains six optimization algorithms that are conventionally used, while the second set consists of four popular variants of the original BOA. For example, in the first group of experiments, the results obtained by the IFS-DBOIM method is compared with the following metaheuristic methods, namely, Ant Lion Optimization (**ALO**) (Mirjalili et al., 2015), original Butterfly Optimization Algorithm (**BOA**) (Arora and Singh 2019), Genetic Algorithm (**GA**) (Whitley, 1994), Grasshopper Optimization Algorithm (**GOA**) (Mirjalili et al., 2018), Particle Swarm Optimization (**PSO**) (Kennedy et al., 1995), and Sine- Cosine Algorithm (**SCA**) (Mirjalili et al., 2016). In contrast, in the other group, the results of our method are compared with: (1) Chaotic BOA (**CBOA**) (Arora and Arora, 2017), (2) Dynamic Butterfly Optimization Algorithm (**DBOA**) (Tubishat et al., 2020), (3) Optimization and Extension of binary Butterfly Optimization Algorithm (**OEbBOA**) (Zhang et al., 2020), and (4) S-shaped binary Butterfly Optimization Algorithm (**S-bBOA**) (Arora et al., 2019). Note that all results related to the aforementioned algorithms are obtained from Sadeghian et al. (2021). In addition, the maximum number of fitness evaluations is used to fix the number of iterations in the current work to have a fair comparison between the state-of-the-art methods and the proposed IFS-DBOIM algorithm. The main reason for selecting the number of fitness evaluations to fix the number of iterations is that each time the fitness is evaluated, it conveys some information about the problem instance. Thus, limiting the number of fitness evaluations reflects the total "amount of information" that one algorithm can obtain from the problem. Here, the obtained information is represented in terms of two performance measures: (1) classification accuracy and (2) fitness score to compare the performance of all the listed algorithms.

4.4.1. Performance comparison in the first group of experiment

Table 3 summarizes the results that are obtained from the first group of experiments. The performance of the IFS-DBOIM method and other aforementioned metaheuristic feature selection methods are compared in achieving classification accuracy (*CA*) and the number of selected features (*P*) by each method over 30 iterations. The maximum classification accuracy percent and the optimal number of selected features are shown in bold. In addition, the mean, standard deviation (S.D), and rank correspond to the average classification accuracy, standard deviation, and the effectiveness of the respective algorithm. The approach with a lower rank is considered more effective than the one with a higher rank.

According to Table 3, the IFS-DBOIM method has shown a superior classification accuracy (93.36%) than six other baseline feature selection methods on all the datasets, except for four (Australian, Exactly, M-of-N, and Vowel) on which PSO and SCA outperform the other approaches. It should be noted that these have the lowest number of features among all the datasets. This observation shows that the PSO and SCA can effectively detect the change in the average fitness function in the early phase of optimization when employed on low-dimensional datasets. Few other factors such as the initialization method, the search mechanism, and the effective parameter tuning might also favor the better performance of PSO and SCA. However, the IFS-DBOIM achieves the best mean classification accuracy of 90.43%, followed by PSO that realizes a mean classification accuracy of 86.15%. Otherwise, our approach performs exceptionally well on high-dimensional datasets (Tox-171, Yale, and Penglungew) compared to the PSO method. On these three datasets, our method obtains a mean classification accuracy of 94.71%, which is 10.71% higher than the mean classification accuracy of 84% achieved by the PSO method.

Table 3 ranks the methods' performances in terms of their average classification accuracy rates. According to the ranking, the IFS-DBOIM approach realizes the best overall classification performance, followed by PSO, GA, SCA, BOA, GOA, and ALO methods. This phenomenon is due to the improved local search ability and increased solution diversity of the

introduced IFS-DBOIM approach, which finds the best outcome from the pool of various solutions. The outstanding performance of the proposed method on the high-dimensional datasets can be explained in terms of the working of DBOA and FIM schemes. In the early phase of each iteration, LSAM employs mutation operation on the current best solution and computes the mutated solution. This intermediate solution is compared with other candidate's solutions, forming an exploration strategy. Therefore, this step enhances the algorithm's ability to discover improved solutions based on the current ones populating in the search space. However, the IFS-DBOIM algorithm outperforms others on the high-dimensional datasets. Though the applied LSAM strategy effectively produces mutated solutions, they are the best in the event of increased features. It helps in avoiding the local optima, thereby replacing the worst solution with a better one.

In contrast to DBOA, the other competitive metaheuristic algorithms do not scale well with complexity. That is, where the number of elements exposed to mutation is large, there is often an exponential increase in search space size. This makes it extremely difficult to use these algorithms on high-dimensional optimization problems like medical image classification and microarray gene expression datasets. In addition, the role of FIM is also important to explain the outstanding performance of the proposed feature selection method. It is known that FIM filters the relevant set of features based on their relevance and redundancy with the previously selected features. Therefore, the probability of determining the set of more informative features increases with the dimensionality of datasets. This step not only helps to realize better classification accuracy but also reduces irrelevant and redundant features during search space reduction. However, other baseline algorithms lack filtering optimal feature sets because they don't use any selection criterion to measure the significance of the features. Based on the combined performance of DBOA and FIM, it can be concluded that the proposed IFS-DBOIM method effectively maintains the trade-off between the classification accuracy and the number of selected features.

In Table 3, the average (avg.) number of selected features using the IFS-DBOIM and six similar methods are reported. The experimental results demonstrate that the proposed method selects the least number of avg. features (154.28) from all the datasets that, as per Table 1, have 383.70 avg. number of features. Henceforth IFS-DBOIM achieves a 59.79% feature reduction rate followed by GA, PSO, ALO, GOA, SCA, and BOA. It shows that the proposed method effectively reduces the large feature search space without compromising the classification accuracy. In a nutshell, the proposed IFS-DBOIM approach identifies the optimal feature combination with the highest probability. To summarize, our method shows great compatibility with the SVM classifier as it obtains the best classification accuracy on sixteen out of twenty datasets, thereby proving its potential to avoid the overfitting problem faced by other algorithms on high-dimension datasets.

Table 4 recapitulates the fitness values of seven feature selection algorithms for all twenty datasets. It concludes the effectiveness of the IFS-DBOIM method over the remaining approaches except for four datasets (Australian, Exactly, M-of-N, and Vowel) on which PSO outperforms the others. Since the minimum fitness value indicates superior solution quality, the classification accuracy achieved by the PSO is better than the IFS-DBOIM method on the datasets mentioned above. Globally, the proposed feature selection method realized the lowest fitness values on the remaining sixteen datasets. Due to this, the maximum classification accuracy rate is achieved by our method on the corresponding datasets, followed by PSO, GA, SCA, BOA, GOA, and ALO.

Fig. 2 shows the classification accuracies of seven feature selection methods on twenty experimental datasets. We selected Penglungew, TOX-171, and Yale datasets as the benchmark to interpret the average classification accuracy achieved by the algorithms because they are high-dimensional datasets with an imbalanced ratio between the number of instances and original features. On the Penglungew dataset, the ALO method realizes maximum classification accuracy in the initial iterations. Moreover, after forty iterations, the proposed method outperforms the others, showing that the effective number of features selected from the IFS-DBOIM approach bears more relevance than those computed by other methods. The average classification accuracy of our method is better than those of the other six methods after sixty iterations except over the Australian, Exactly, M-of-N, and Vowel datasets. Interestingly, PSO realized maximum classification accuracy on all these four datasets, which shows that its global search ability has the upper hand over our interaction maximization strategy in selecting the relevant features. However, the drawback is that PSO selected more features in all four cases than our method. One plausible reason for this behavior is that the proposed method's FIM algorithm eliminates less informative features in the early iterations without considering their roles in classification accuracy. Overall, it can be concluded that the proposed method maintains the most robust balance between the lowest number of selected features and the classification accuracy.

The classification accuracy of the IFS-DBOIM approach for the remaining two high dimensional datasets, TOX-171, and Yale is always higher than the other methods. The TOX-171 is a high-dimensional microarray dataset containing a huge number of irrelevant and redundant features. Our method achieves the best classification accuracy of 97.02% using only 2404 features compared to other feature reduction methods. Similarly, Yale is another popular high-dimensional face recognition-based imaging dataset over which our method achieves superior classification accuracy. The proposed method realized maximum classification accuracy of 87.92% with the minimum number of selected features. The performance of the IFS-DBOIM approach indicates that it can effectively deal with the Curse of Dimensionality (CoD) of high-dimensional datasets.

No.	ALO		BOA		GA		GOA		PSO		SCA		IFS-DBOIM			
	CA	*P*	*CA*	*P*	*CA*	*P*	*CA*	*P*	*CA*	*P*	*CA*	*P*	SVM	NB	DT	*P*
1	79.17	06.61	78.64	07.06	84.10	05.63	79.54	06.42	**85.04**	06.39	78.91	06.43	82.10	85.32	80.12	04.18
2	71.03	09.44	72.30	11.63	75.05	09.03	72.51	09.93	76.23	09.97	72.68	12.36	**83.68**	67.80	69.33	07.30
3	93.40	11.11	93.05	13.06	94.52	09.33	93.05	10.97	95.69	10.18	93.49	12.66	**98.46**	86.52	79.66	04.20
4	70.40	07.08	75.11	11.13	81.70	08.03	72.30	06.69	**87.46**	06.73	87.46	10.90	83.20	77.54	69.12	05.78
5	68.34	09.71	67.98	11.46	70.63	08.36	68.54	09.46	71.70	09.01	68.50	11.60	**80.02**	89.66	91.28	06.02
6	55.23	49.19	55.70	60.56	60.02	46.33	56.03	49.38	61.48	48.57	55.73	57.06	**74.66**	83.69	85.20	39.60
7	88.04	15.79	88.37	20.13	91.41	13.80	88.85	16.64	93.07	15.80	88.09	17.76	**98.33**	81.32	80.33	08.38
8	75.27	45.08	75.50	52.06	80.13	40.16	75.64	44.37	80.97	42.33	75.73	48.23	**96.18**	83.18	89.70	29.02
9	84.20	07.18	90.20	11.66	91.15	08.50	85.40	07.22	**95.96**	06.97	92.48	10.30	77.80	61.11	65.50	06.42
10	93.78	09.36	93.59	08.60	94.45	05.36	93.56	09.87	95.00	08.74	93.89	07.83	**99.34**	64.40	60.33	02.73
11	90.81	160.86	91.18	162.00	93.77	135.96	91.24	161.40	93.94	151.74	91.86	178.46	**99.20**	80.03	80.02	98.62
12	84.58	20.24	85.22	24.83	87.86	19.20	85.13	20.24	88.81	19.81	85.11	26.23	**92.18**	**77.30**	80.30	14.20
13	83.47	28.97	84.20	40.96	89.57	26.56	84.77	29.25	91.80	28.59	85.32	35.83	**98.50**	81.20	89.12	19.40
14	89.58	28.16	90.60	42.33	91.71	28.80	90.19	28.88	93.02	29.00	90.62	41.03	**96.12**	72.40	76.50	21.68
15	75.09	10.82	76.59	14.96	80.67	11.03	75.65	10.86	81.73	10.50	77.90	14.06	**88.20**	68.02	73.10	07.25
16	74.51	2875.79	75.59	3621.30	82.14	2825.90	78.53	2878.48	85.36	2846.23	76.26	3268.73	**97.02**	87.58	81.18	2403.74
17	94.66	07.85	95.05	09.93	95.94	07.16	94.77	07.93	96.38	07.58	95.05	08.73	**99.33**	91.04	78.66	03.82
18	90.11	07.31	91.80	10.50	93.24	08.86	90.57	07.25	**93.50**	07.05	92.02	10.63	87.20	84.33	77.54	07.92
19	79.96	11.32	82.68	19.40	82.54	13.93	80.64	11.79	83.17	11.47	83.09	19.00	**89.22**	88.10	68.12	12.33
20	62.57	509.28	63.05	591.20	68.15	492.96	64.08	509.80	72.70	501.21	64.17	563.23	**87.92**	71.10	67.33	383.16
Mean	80.21	191.55	81.32	237.23	84.43	186.24	81.04	191.84	86.15	188.89	82.41	218.05	**90.43**	80.08	78.62	**154.28**
S.D.	10.75		10.74		09.67		10.40		09.41		10.69		**07.09**	**07.55**	08.21	

Rank	7	IV	5	VII	3	II	6	V	2	III	4	VI	**1**			I

Table 3. Performance of the IFS-DBOIM method in terms of classification accuracy rates (%) and the number of selected features for UCI datasets in the first group of experiments. Here, *CA* represents average classification accuracy (%), and *P* indicates the average number of selected features.

4.4.2. Performance comparison in the second group of experiment

In this subsection, another experiment is conducted to evaluate the performance of the IFS-DBOIM method. Table 5 summarizes the experimental results regarding classification accuracy and the number of selected features of the IFS-DBOIM and the four variants of the original BOA (CBOA, DBOA, OEbBOA, S-bBOA). The maximum classification accuracy and the optimal number of selected features are shown in bold. According to Table 5, the IFS-DBOIM method achieves maximum classification accuracy except for four datasets (Australian, Exactly, M-of-N, and Vowel) on which DBOA and S-bBOA outperform the other approaches with a slight performance difference from the proposed method. The IFS-DBOIM approach realizes much higher classification accuracy than other similar methods on sixteen datasets. The mean classification accuracy of the IFS-DBOIM method is 90.43% which is a 1.28% enhancement in the conventional DBOA, which forms a part of our method. Since the standard deviation of our method is lower than that of the DBOA, most of the individual classification results are ideal and clustered around the mean classification accuracy.

Table 4. The average fitness values of all competing algorithms over 30 runs

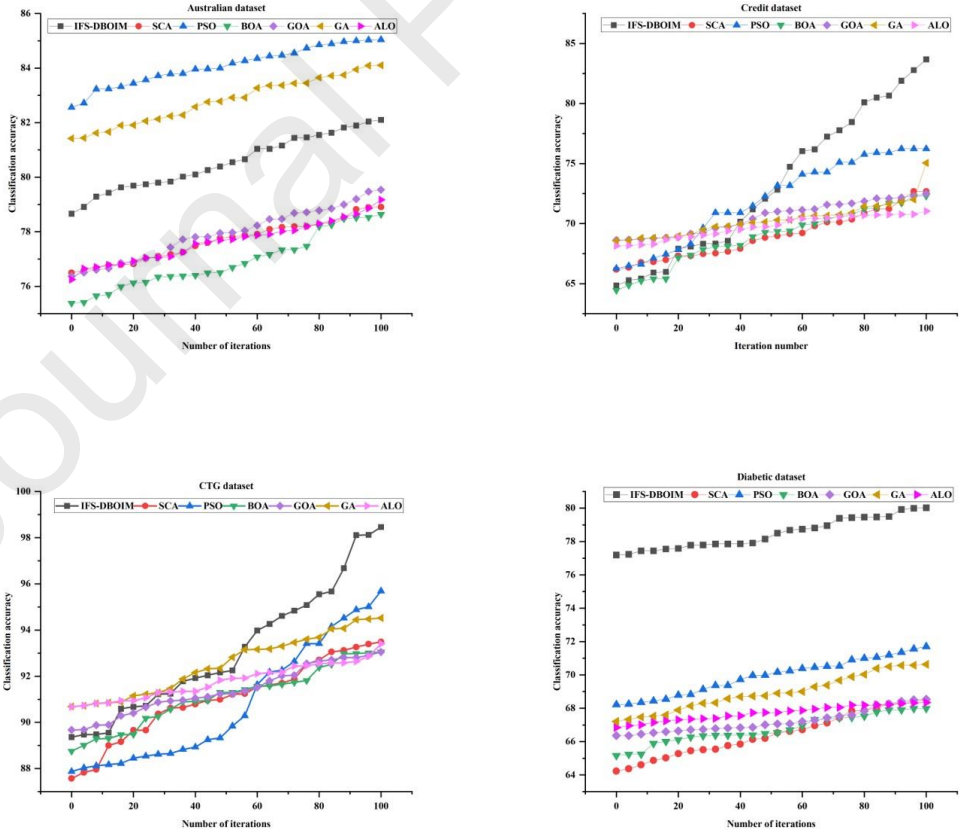
No.	ALO	BOA	GA	GOA	PSO	SCA	IFS-DBOIM
1	0.21	0.21	0.16	0.20	**0.15**	0.21	0.15
2	0.29	0.28	0.25	0.27	0.23	0.27	**0.17**
3	0.07	0.07	0.05	0.07	0.04	0.07	**0.03**
4	0.29	0.25	0.18	0.27	**0.12**	0.23	0.14
5	0.31	0.32	0.29	0.31	0.28	0.31	**0.22**
6	0.44	0.44	0.40	0.44	0.38	0.44	**0.35**
7	0.12	0.12	0.08	0.11	0.07	0.12	**0.05**
8	0.24	0.24	0.20	0.24	0.19	0.24	**0.16**
9	0.16	0.10	0.09	0.15	**0.04**	0.08	0.06
10	0.06	0.06	0.05	0.06	0.05	0.06	**0.04**
11	0.09	0.09	0.06	0.09	0.06	0.08	**0.04**
12	0.15	0.15	0.12	0.15	0.11	0.15	**0.09**
13	0.16	0.16	0.10	0.15	0.08	0.15	**0.07**
14	0.10	0.10	0.08	0.10	0.07	0.10	**0.06**
15	0.25	0.23	0.19	0.24	0.18	0.22	**0.15**
16	0.25	0.24	0.18	0.21	0.14	0.24	**0.10**
17	0.05	0.05	0.04	0.05	0.03	0.05	**0.01**
18	0.10	0.08	0.07	0.09	**0.07**	0.08	0.07
19	0.20	0.18	0.17	0.19	0.17	0.17	**0.13**
20	0.37	0.37	0.32	0.36	0.27	0.36	**0.18**
Mean	0.20	0.19	0.15	0.19	0.14	0.18	**0.11**
S.D.	0.10	0.10	0.09	0.10	0.09	0.10	0.07
Rank	4	4	2	3	2	3	**1**

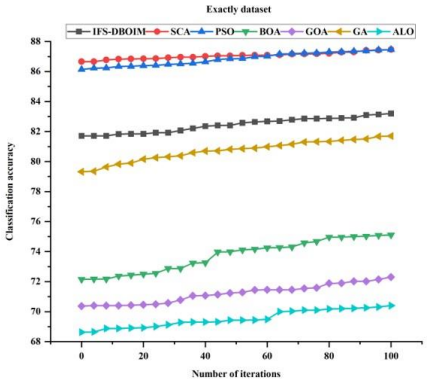
Table 5. Performance of the IFS-DBOIM method in terms of classification accuracy rates (%) and the number of selected features for UCI datasets in the second group of experiments. Here, *CA* represents average classification accuracy (%), and *P* indicates the average number of selected features.

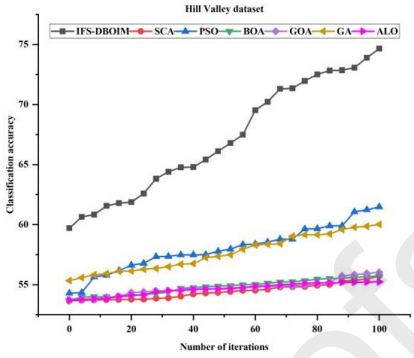
No.	CBOA		DBOA		OEbBOA		S-bBOA		IFS-DBOIM			
	CA	*P*	*CA*	*P*	*CA*	*P*	*CA*	*P*	SVM	NB	DT	*P*
1	79.77	06.12	**88.04**	04.53	83.22	07.18	79.12	09.12	82.10	85.32	80.12	04.18
2	74.35	10.08	77.71	08.70	78.11	13.55	74.42	12.52	**83.68**	67.80	69.33	07.30
3	94.21	07.12	**98.46**	04.53	97.15	06.20	96.58	05.44	**98.46**	86.52	79.66	04.20

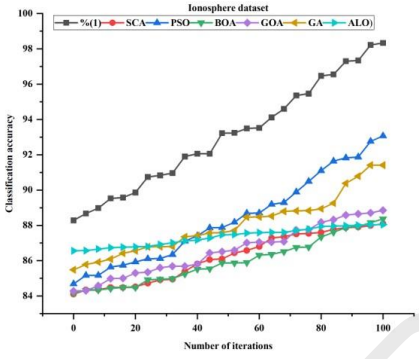
4	82.31	08.30	**99.85**	06.50	89.25	08.92	97.24	10.16	83.20	77.54	69.12	05.78
5	68.77	07.22	73.15	06.56	71.33	06.46	68.11	07.34	**80.02**	89.66	91.28	06.02
6	56.11	38.72	63.98	43.20	61.44	52.40	57.88	44.12	**74.66**	83.69	85.20	39.60
7	97.70	11.33	95.48	09.16	96.65	10.44	90.70	08.40	**98.33**	81.32	80.33	08.38
8	79.44	38.42	83.75	33.93	81.00	41.22	76.61	36.14	**96.18**	83.18	89.70	29.02
9	92.66	07.30	99.76	06.76	96.99	06.98	97.20	08.26	77.80	61.11	65.50	06.42
10	93.77	04.36	97.50	02.73	95.11	03.33	94.76	05.25	**99.34**	64.40	60.33	02.73
11	91.20	164.48	96.74	104.63	92.91	142.23	87.75	192.41	**99.20**	80.03	80.02	98.62
12	87.55	23.50	90.28	18.20	91.03	24.75	86.22	27.62	**92.18**	77.30	80.30	14.20
13	94.20	18.24	96.13	21.40	95.14	25.28	93.62	23.68	**98.50**	81.20	89.12	19.40
14	90.91	31.20	94.13	33.00	91.40	29.68	91.11	27.58	**96.12**	72.40	76.50	21.68
15	82.90	11.78	86.54	09.70	85.16	13.21	84.63	12.44	**88.20**	68.02	73.10	07.25
16	77.12	3320.11	89.07	2743.86	83.07	2907.46	76.11	3120.12	**97.02**	87.58	81.18	2403.74
17	96.44	04.98	98.16	04.70	98.44	05.76	96.53	07.98	**99.33**	91.04	78.66	03.82
18	91.77	14.86	94.73	08.46	92.12	11.16	95.13	10.28	87.20	84.33	77.54	07.92
19	80.30	19.30	84.42	15.26	83.10	18.40	74.29	21.14	**89.22**	88.10	68.12	12.33
20	67.22	612.33	75.18	466.96	70.15	524.42	64.54	532.16	**87.92**	71.10	67.33	383.16
Mean	83.93	217.98	89.15	177.63	86.63	192.95	84.12	206.10	**90.43**	80.08	78.62	**154.28**
S.D.	10.91	723.90	09.85	597.17	10.01	632.92	11.72	678.69	**07.09**	07.55	08.21	**522.61**
Rank	15		2		3		4		**1**			

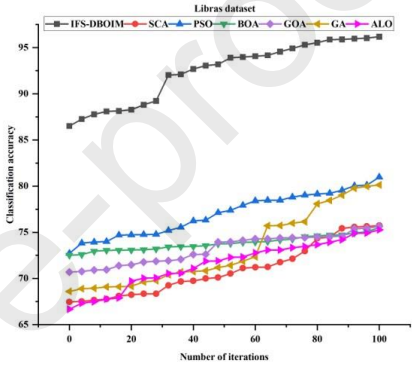
Although the number of selected features is less than the original set, it improved the classification accuracy showing that all features are not required for achieving the best results. The results demonstrate that the IFS-DBOIM realized a maximum feature reduction rate on all the datasets by selecting an average of only 154.28 features. It clearly shows that our approach can effectively determine an optimal set of features to realize the maximum feature reduction rate on all datasets.

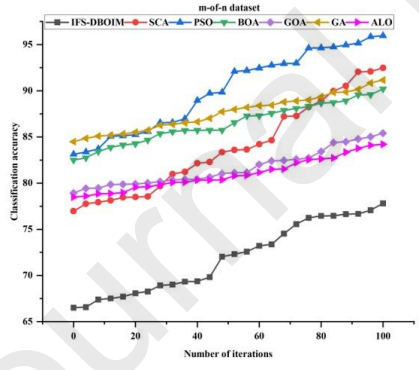


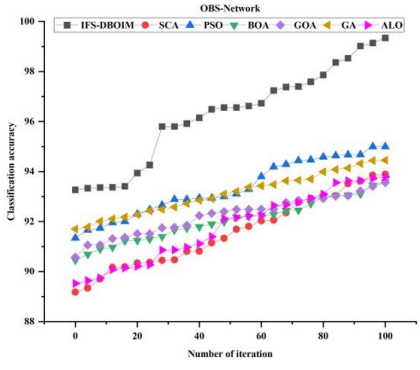


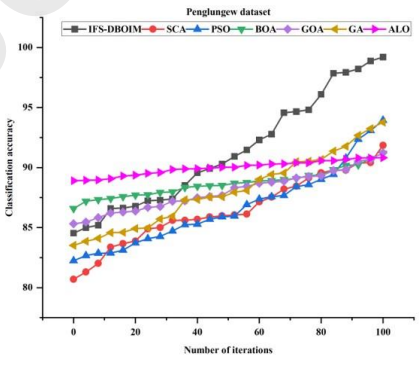


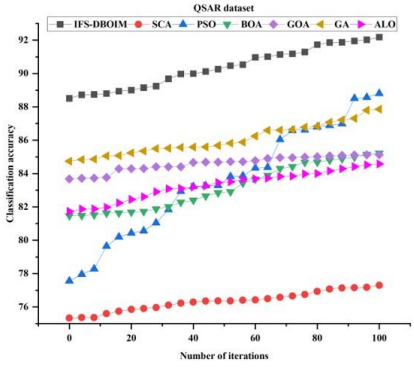












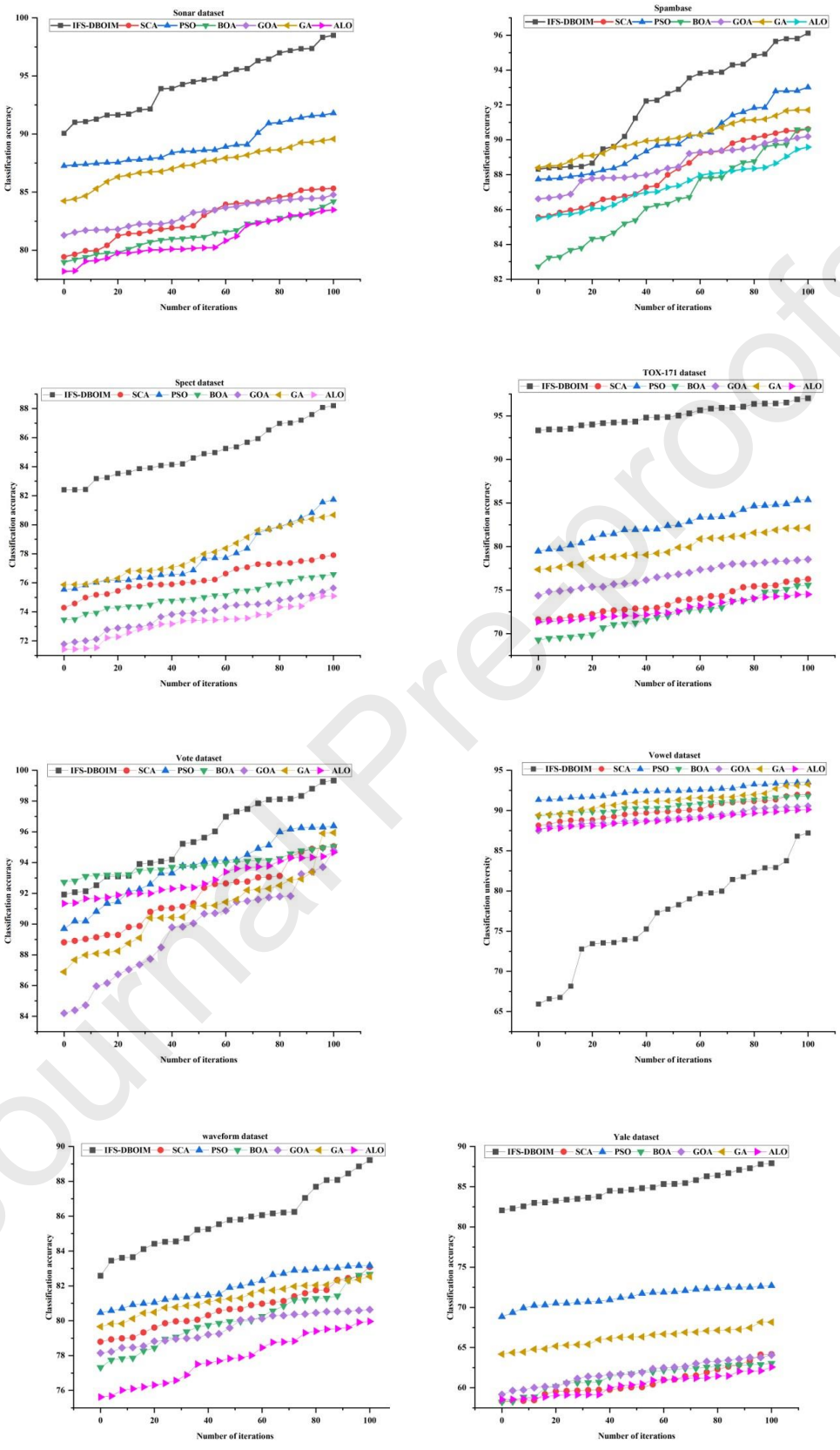


Fig. 2. shows the classification accuracies of seven feature selection methods on all twenty datasets.

4.4.3. Sensitivity and specificity comparison

In addition to classification accuracy and the number of selected features, sensitivity and specificity were also computed to compare the performance of IFS-DBOIM with those from other baseline feature selection methods. From Eq. 15 and 16, it can be deduced that both sensitivity and specificity are only applied to binary classification problems as they are two instance-based ratios. Therefore we have used 13 binary datasets out of 20 to evaluate both performance measures. Fig. 3 and 4 compare all the state-of-the-art methods' sensitivity and specificity scores with the proposed IFS-DBOIM approach. From Fig. 3, it can be observed that the IFS-DBOIM method realizes a higher score compared to the baseline algorithms on eleven datasets with Diabetic and Sonar as exceptions.

Similarly, as shown in Fig. 4, the IFS-DBOIM method again achieves the best specificity score on eleven datasets except for the credit and sonar. Henceforth, the IFS-DBOIM approach can be an effective feature reduction method for binary decision-based medical datasets where a machine learning model must correctly identify a truly infected person. As discussed earlier, due to its global search ability, the PSO algorithm is second-best on four datasets (Australian, Credit, Exactly, Spect), thereby achieving a higher sensitivity score than our method. Two algorithms (ALO and GOA) performed almost equally well and are the second-best approaches that gain a good specificity score after the IFS-DBOIM method on different datasets.

4.4.4. Time complexity analysis

Time complexity is another important performance measure that indicates the total time that an algorithm consumes to find the optimal global solution. Table 6 presents the average computation times of ten baseline algorithms and the proposed method over 30 independent runs. It can be observed that the IFS-DBOIM realized the lowest computation time on sixteen datasets out of twenty, whereas the PSO outperformed on three and the SCA on a single dataset. Table 6 ranks the approaches' performance according to their average computation times. According to the ranking, the IFS-DBOIM realizes the minimum average computation time on all 20 datasets, followed by PSO, SCA, CBOA, BOA, S-bBOA, ALO, OEbBOA, DBOA, GOA, and GA.

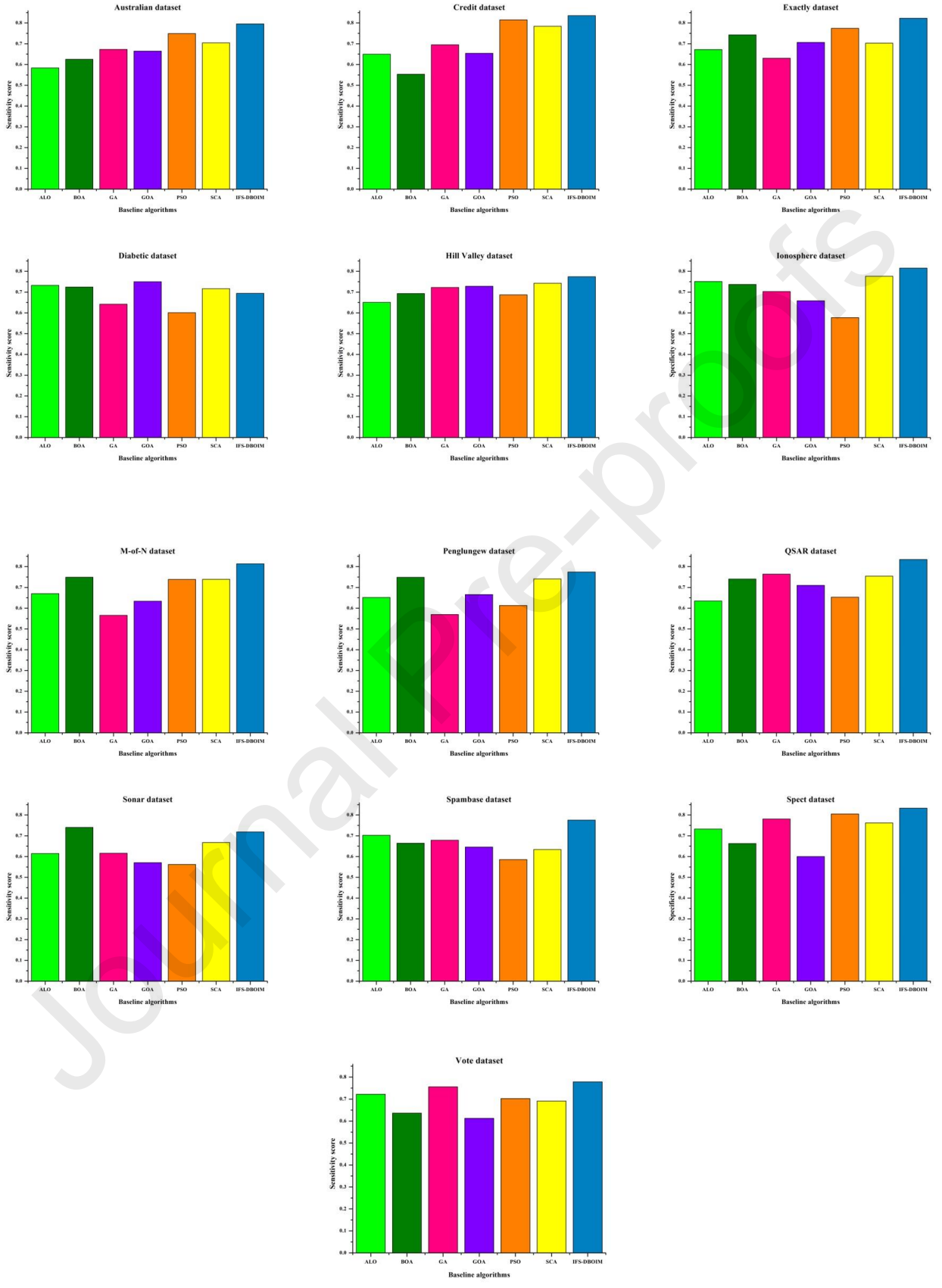


Fig. 3. Sensitivity score comparison between baseline algorithms and the IFS-DBOIM method

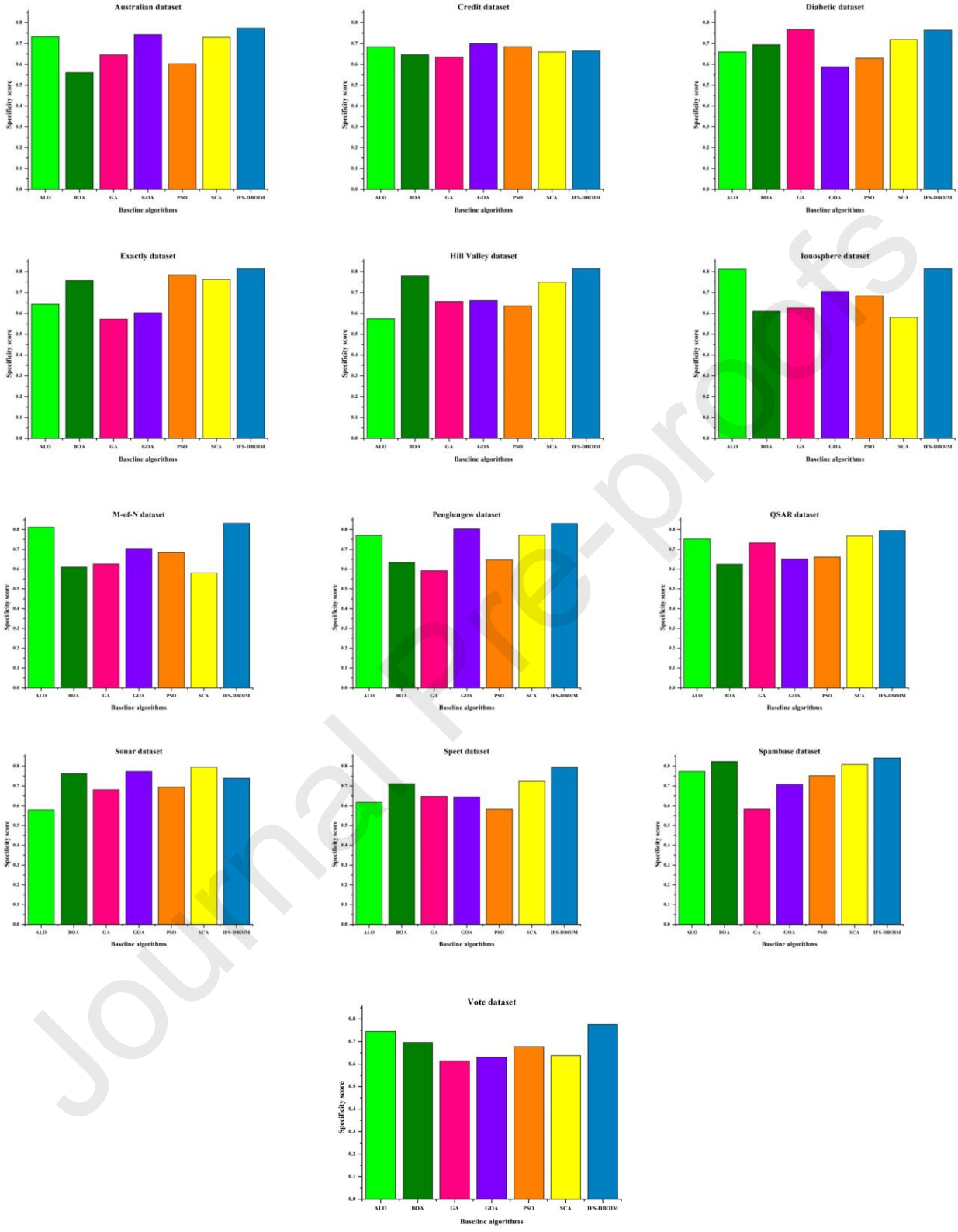


Fig. 4. Specificity score comparison between baseline algorithms and the IFS-DBOIM method

Table 6. The average computation time of all the feature selection algorithms

No.	ALO	BOA	GA	GOA	PSO	SCA	CBOA	DBOA	S-bBOA	OEbBOA	IFS-DBOIM
1	17.9928	12.8279	32.5452	08.5666	**14.7995**	07.9821	10.1263	14.6505	09.2691	08.8369	06.5092
2	14.2644	16.7271	39.6852	13.3632	**12.5622**	**08.0548**	09.0182	09.6948	14.0237	17.6129	03.9430
3	13.9191	13.3163	43.5328	21.9403	**01.6120**	09.1377	10.5137	10.0245	08.9304	11.9435	02.6607
4	15.3252	18.3749	26.2702	14.6695	**11.5927**	12.1645	10.6399	09.8345	14.3155	09.0001	05.7631
5	09.8224	10.9443	36.1667	12.9858	**14.3548**	07.8316	09.1419	11.9984	10.0148	12.4231	04.9602
6	62.1708	68.4902	121.5638	49.5319	28.5684	33.7429	44.4315	27.6239	53.7914	62.1490	26.9295
7	08.2389	16.1903	09.2415	12.5306	**10.2109**	08.4858	16.2451	08.9120	11.3701	09.3120	06.0657
8	13.1719	15.9559	21.5219	24.9715	27.5799	26.7290	21.4319	13.8912	15.3537	19.2686	05.2121
9	06.8102	19.8221	16.8181	10.1202	35.9206	35.0650	33.4030	27.5081	24.2653	23.5039	03.0298
10	33.2083	28.9701	27.1077	23.9225	28.4071	29.2741	17.0897	20.2749	26.6075	26.6169	12.2039
11	84.1420	55.1306	81.4679	91.5894	44.5693	73.5139	88.1629	106.8780	99.7538	104.9329	38.7194
12	44.9212	30.4958	48.3133	46.8103	**14.9827**	35.1049	27.2550	45.9618	48.3404	33.1419	19.4310
13	25.1168	71.9746	58.8064	70.8094	50.2621	52.3582	49.9849	20.2455	20.2455	20.2455	09.1824
14	41.4417	29.8081	35.2765	56.3049	33.6364	37.8075	60.4321	59.4122	48.5415	57.4169	24.8346
15	63.7634	39.3629	37.5651	78.5694	35.4418	**17.2148**	26.9699	34.2839	32.8441	28.5032	22.1990
16	71.4914	49.2861	53.0729	47.1934	38.1478	56.2494	36.8829	95.4080	41.6249	45.7104	32.5029
17	34.4980	33.7968	43.0084	71.0893	**19.6820**	28.7624	47.4390	59.0941	37.2784	69.8320	23.7445
18	37.0979	62.7126	84.1728	83.5219	39.3002	40.4153	90.7825	76.6445	34.7554	47.7872	14.9010
19	73.6648	63.7505	100.9986	48.9532	62.5369	89.4274	69.8651	48.1595	89.0975	73.0170	18.4980
20	103.1927	87.1477	94.0709	43.3385	61.3864	78.1453	64.5940	117.5730	114.8045	108.3351	53.2703
Mean	38.7126	37.2542	50.5602	41.5390	29.2776	34.3733	37.2204	40.9036	37.7613	39.4794	16.7280
Rank	7	5	11	10	2	3	4	9	6	8	1

4.4.5. Wilcoxon' test results analysis

Table 7 reports the *p*-values of the IFS-DBOIM in comparison with other baseline feature reduction methods obtained using Wilcoxon's rank-sum test. This test is conducted to determine whether the difference between the results of the proposed method and other approaches is significant or not. Specifically, if the *p*-value is less than 0.05, the results are considered significant, whereas a greater *p*-value indicates otherwise. It can be observed that in most of the comparisons, the *p*-values obtained using the rank-sum test are lesser than 0.05, which proves that the effectiveness of our method is statistically significant. Compared with the original DBOA, the *p*-value is less than 0.05 for 14 out of 20 datasets (except Australian, Credit, Exactly, Hill Valley, OBS-Network, and Waveform), which shows the significance of the improvement introduced in our method.

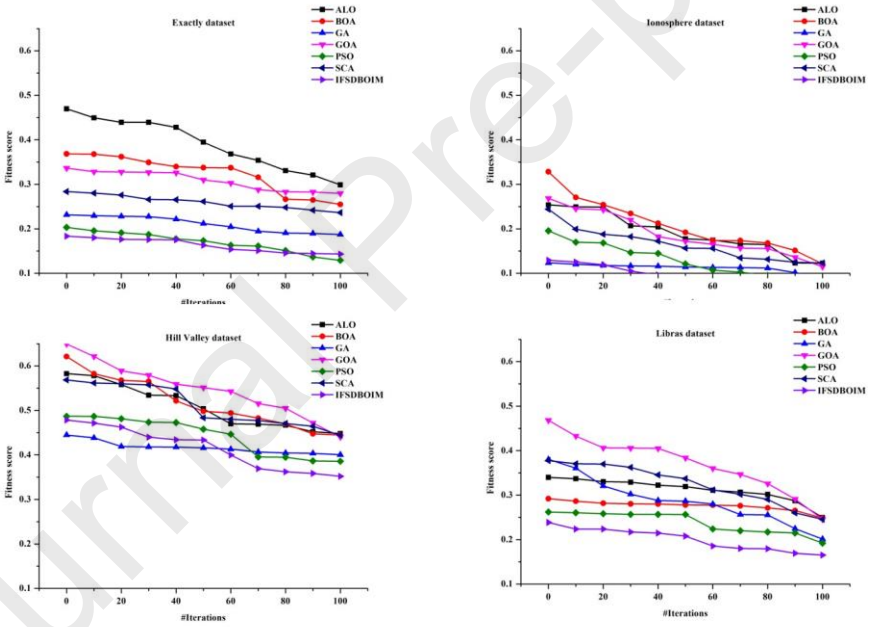
Table 7. The *p*-value based on the Wilcoxon test of all the algorithms in terms of average classification accuracy over 30 runs ($p \geq \mathbf{0.05}$) is marked with bold color.

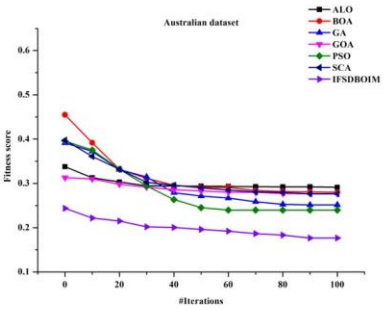
No.	ALO	BOA	GA	GOA	PSO	SCA	CBOA	DBOA	S-bBOA	OEbBOA
1	1.68E-09	2.53E-09	1.08E-05	5.41E-09	**7.25E- 04**	7.70E-09	2.0E-04	**1.0E-00**	1.8E-04	2.98E-06
2	6.61E-11	1.94E-10	3.41E-06	2.98E-10	**1.62E-02**	3.42E-10	**3.01E-03**	**2.05E-01**	3.4E-07	**2.8E-03**
3	7.30E-11	3.65E-11	3.13E-10	2.98E-11	2.09E-07	3.65E-11	**5.07E-02**	1.70E-05	3.7E-09	1.3E-04
4	1.97E-11	1.90E-11	5.41E-10	6.15E-11	1.28E-05	1.89E-11	1.37E-07	**3.10E-02**	6.2E-11	4.2E-07
5	2.95E-08	1.10E-08	1.72E-03	7.88E-07	**3.64E-02**	3.49E-08	**3.7E-04**	2.41E-07	6.03E-05	0.03E-01
6	1.07E-09	3.82E-09	1.86E-03	1.70E-08	**2.15E-02**	8.47E-09	9.1E-11	**3.8E-04**	13.1E-08	2.07E-09
7	1.61E-10	5.31E-10	1.866E-06	4.83E-10	**3.76E-03**	9.90E-11	4.7E-09	7.3E-05	**6.18E-03**	6.01E-06
8	2.02E-07	5.38E-07	2.15E-03	4.10E-07	**5.10E-02**	1.15E-06	5.21E-06	6.1E-11	9.31E-10	1.7E-11
9	1.99E-11	1.89E-11	2.97E-10	2.10E-11	**7.25E-04**	1.17E-10	0.7E-06	4.07E-07	**1.13E-03**	3.3E-05
10	2.52E-08	3.09E-09	4.15E-08	4.03E-09	1.80E-06	4.41E-09	4.0E-07	**1.8E- 01**	**2.4E-02**	2.0E-04
11	4.41E-06	3.32E-06	1.14E-03	4.42E-06	**2.19E-02**	1.43E-05	**3.0E-03**	1.1E-05	3.39E-07	3.8E-06
12	1.09E-10	2.73E-10	7.87E-05	7.34E-10	**4.52E-03**	2.60E-10	2.0E-11	1.0E--06	7-4E-09	**3.1E-03**
13	2.86E-10	1.77E-10	1.10E-06	1.68E-10	**6.19E-04**	4.60E-10	1.06E-14	4.30E-05	2.4E-13	0.3E-09
14	3.02E-11	3.34E-11	1.55E-09	4.97E-11	1.07E-04	4.50E-11	2.03E-17	2.02E-05	1.3E-08	3.4E-13
15	3.46E-10	3.63E-10	3.37E-07	1.32E-10	**3.36E-04**	1.20E-08	1.91E-10	1.16E-17	3.4E-14	2.3E-12
16	6,12E-10	6.12E-10	2.00E-06	1.85E-08	**3.58E-03**	5.97E-09	3.21E-15	6.31E-08	5.1E-11	1.6E-08
17	3.89E-07	2.16E-07	1.66E-04	1.92E-07	**4.67E-04**	2.08E-07	3.14E-15	0.19E-13	2.0E-09	1.3E-07
18	9.19E-10	1.72E-06	1.71E-03	3.06E-07	**4.82E-02**	1.86E-05	4.01E-11	3.14E-11	1.9E-16	6.6E-13

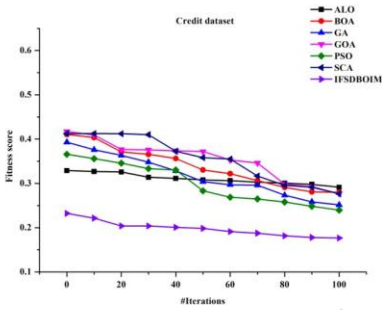
19	3.01E-11	1.87E-07	1.43E-06	3.01E-11	1.40E-04	1.49E-06	4.07E-07	**1.13E-03**	3.3E-05	3.14E-11
20	7.12E-09	1.43E-08	6.36E-05	7.09E-08	**1.71E-01**	7.70E-08	1.18E-10	4.59E-09	3.37E-13	1.49E-09

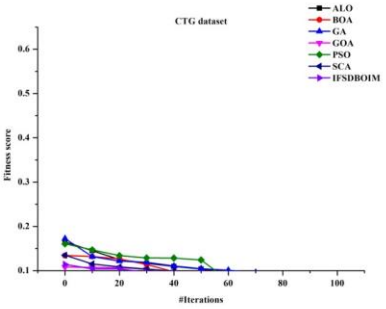
4.4.6. Convergence analysis

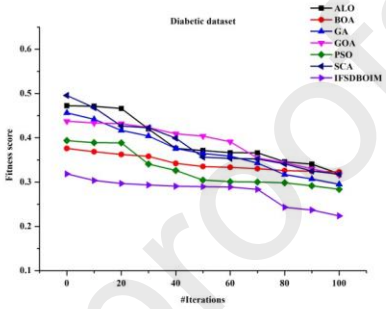
The convergence rate is an important performance measure that represents how quickly an algorithm approaches its solution. In Fig. 5, convergence curves of all the competing algorithms were plotted between fitness score and iteration counts to show their behavior on all the datasets. After contemplating on Fig. 5, two main comparison perspectives arise. First, in the early iterations, IFS-DBOIM achieves lower fitness scores on all the datasets except for the three (Exactly, M-of-N, and Vowel) even though it did not converge like state-of-the-art methods. This might be imputed to the executed initialization method, population size, and the search strategy of the different algorithms. Generally speaking, the convergence rate of the original DBOA is slower than other baseline algorithms, which explains a similar low execution rate of the derived IFS-DBOIM. In addition, it is assumed that the involvement of the mutual information-based FIM further slows down the convergence rate to better the fitness score in our method. Evidently, this highlights the significance of the employed local search strategy (LSAM) to find the possible global optima or a closer solution.

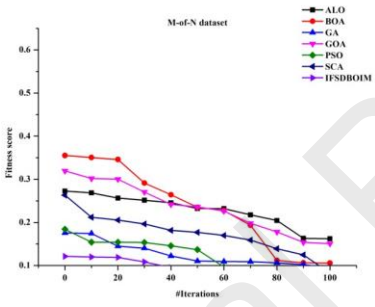


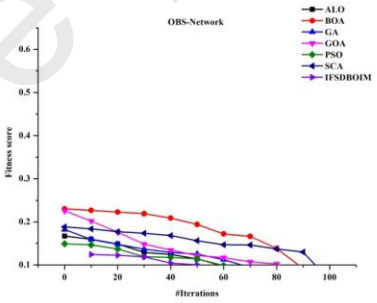


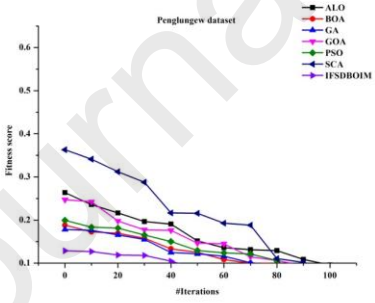


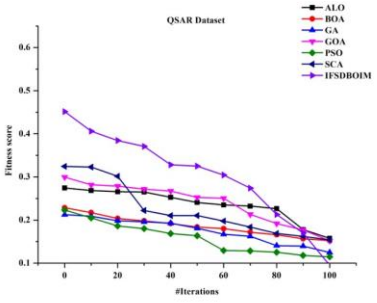












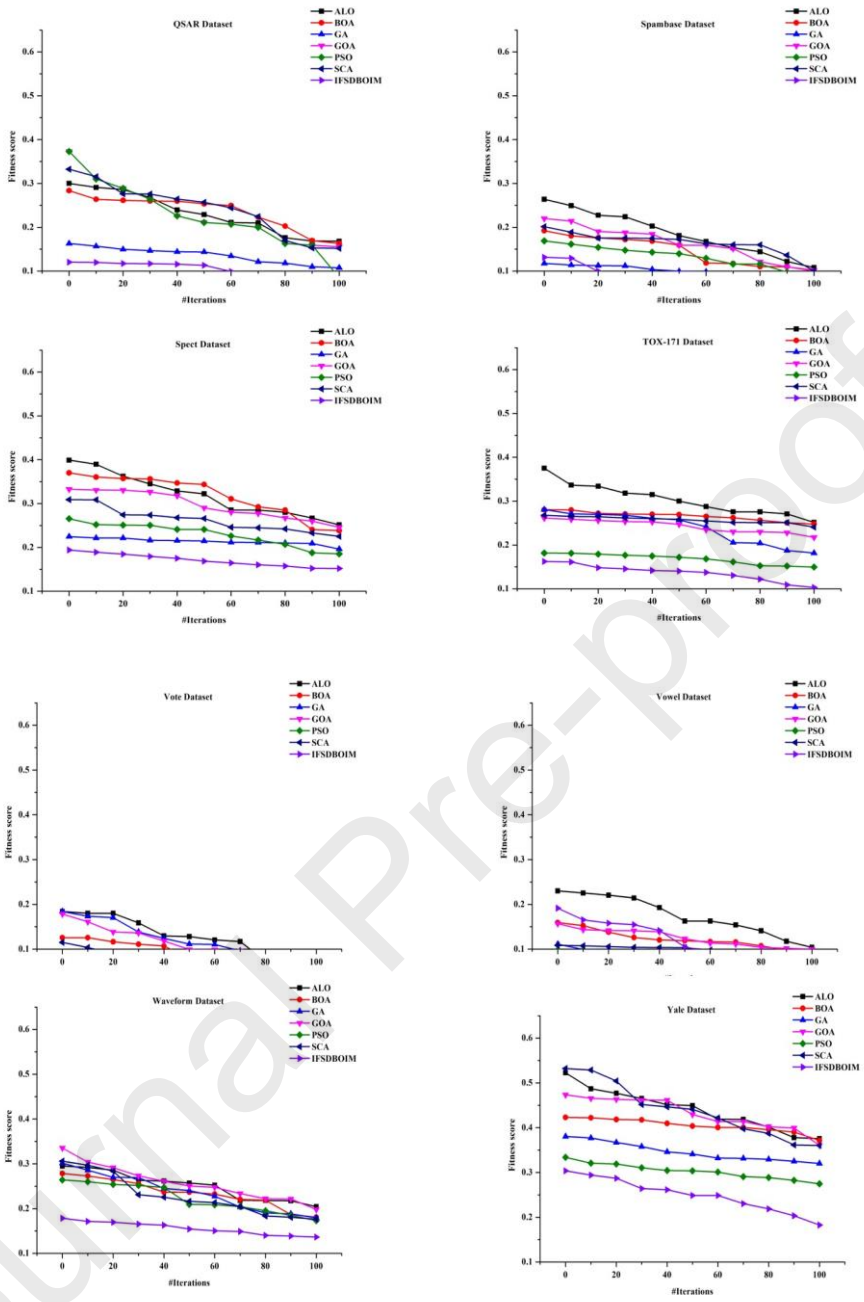


Fig. 5. Comparison of convergence curves of all competing algorithms over the 20 datasets.

4.4.7. Implications of the study

Search space optimization is an indispensable part of machine learning that aims to locate plausible input points with acceptable solution quality. In this research work, we developed a hybrid feature selection model, namely, Iterative Feature Selection using Dynamic Butterfly Optimization-based Interaction Maximization (IFS-DBOIM) algorithm to eliminate irrelevant and redundant data points from high-dimensional space. Although baseline metaheuristic algorithms (ALO, BOA, GA, GOA, PSO, and SCA) realize good classification accuracy when used for solving dimension reduction, still most of them suffer from two major problems: (1) poor ability to escape from local optima, which lead to premature convergence problems, and (2) low solution diversity. The DBOA effectively resolves both problems by introducing a Local Search Algorithm based

on Mutation (LSAM) that works in two ways: (1) improving the quality of the current best solution by introducing mutations which further helps to solve local optima problems; (2) random solutions selection with each iteration to enhance the diversity of the final set. In addition, baseline algorithms can only monitor the newly selected feature's relevance and redundancy level. At the same time, the IFS-DBOIM rigorously considers the interaction of the features with the previously selected ones. If the new feature is important and non-redundant, it must share minimum mutual information with the already selected features. Henceforth, the final feature subset becomes more relevant but with fewer features. Since the performance of the IFS-DBOIM is outstanding on high-dimensional datasets, it indicates that its local search strategy is better than that of the remaining algorithms. Therefore, it can be effectively used to solve similar research problems such as EEG-based Channel Selection (CS) in the Brain-Computer Interface (BCI) modeling (Jin et al., 2020), design parameter selection in engineering models (Bilewu et al., 2015), image segmentation (Oliva et al., 2019), and gene clustering (Banu et al., 2015).

4.4.8. Limitations of the work

In general, the proposed feature selection method uses two selection criteria: relevancy and redundancy. This method maintains a good balance between relevancy maximization and redundancy minimization while designing the optimal feature subset. Despite realizing higher classification accuracy rates on multiple datasets using the IFS-DBOIM approach, the proposed method has some limitations that need to be addressed.

The proposed approach calculates the redundancy in terms of sharable information between the previously selected features and the new ones without considering the class label. The feature may share information, but it does not imply that they are redundant; they may share the valuable information with the class attribute. In addition, measuring mutual information from finite data is difficult and non-reliable to determine the redundancy of the features. Therefore, this criterion may not compute the absolute non-redundant set of features for achieving better classification accuracy. Another problem that all metaheuristic-based feature selection methods share is their non-deterministic nature to determine the optimal feature subset. In contrast with exact algorithms whose final result is always fixed, metaheuristics do not provide that kind of bound. They can be very effective on a given instance of a problem but may provide the worst results on another, highlighting that the proposed method is problem-specific and hard to generalize. In practice, the significance of each of the above-mentioned problems depends on the data and properties of each dataset.

5. Conclusions and future scope

This paper proposes a hybrid feature reduction algorithm, Iterative Feature Selection using Dynamic Butterfly Optimization-based Interaction Maximization (IFS-DBOIM). It employs three objective functions to evaluate the fitness of each solution. Firstly, the DBOA is used as a supervised learning method to generate a set of solutions. Then, an information theory-based three-way interaction mechanism is adopted to extract the best optimal feature subset with maximum classification accuracy and reduced model complexity. In the experiment section, twenty standard datasets from the UCI repository are used to assess the performance of the IFS-DBOIM approach. Experimental results showed that the proposed method produced more promising results than the earlier ones on most datasets, especially those with high dimensionality but an extremely small sample size.

The simulation results confirmed that the proposed method achieved better classification accuracy (90.43%) and feature reduction rate than the other baseline algorithms. The statistical significance of the reported results is further confirmed by paired Wilcoxon rank-sum test. Moreover, the proposed method balances better the trade-off between classification accuracy and stability. The performance of the IFS-DBOIM is also compared in terms of classification accuracy, feature reduction rate, fitness score, sensitivity, specificity, time complexity, convergence rate with other popular methods. The results indicate the superiority of the proposed hybrid method with few limitations. Since it is a non-deterministic algorithm, it suffers from a lack of generalization and relies on the characteristics of applied datasets.

Dimensionality is one of the crucial factors that directly affect the performance of expert and intelligent systems. Compared with the research in the related field, our method realized higher comprehensive performance as its key contributions include: (1) the application of information theory and (2) the maintenance of a balance between the exploration and exploitation phase in DBOA for solving complex feature reduction problems. Therefore, the IFS-DBOIM method can be used in preprocessing to find the minimum but relevant features for improving the classification accuracy.

In the future, the IFS-DBOIM may be merged with other recently developed optimization strategies such as EarthWorm Optimization Algorithm (Wang et al., 2018), Elephant Herding Optimization (Wang et al., 2015), Harris Hawks Optimization (Heidariet al., 2019), Jellyfish Search Optimization (Chou et al., 2021), Moth Search algorithm (Wang et al., 2018), Red Deer Algorithm (Fathollahi-Fard et al., 2020), Sea Lion Optimization (Masadeh et al., 2019), and Slime Mould Algorithm (Li et al., 2020) for finding a more robust feature subset. To avoid the overfitting issue in the IFS-DBOIM method, a pruning scheme can be incorporated to develop a new parsimonious classification model. In addition to classification, the proposed method can be demonstrated for tackling regression problems. The proposed method may be tried upon other homologous problems such as EEG-based Channel Selection in Brain-Computer Interface, parameter selection in machine design, and tertiary structure prediction of proteins.

Acknowledgments

It is with true pleasure that we acknowledge the contributions of Swati Mishra, Masters in Bioinformatics, who has dedicated her precious time to reviewing and editing the manuscript. Her suggestions were highly effective in reaching the goals. We are grateful for her academic and scientific writing skills, which helped us to publish this paper.

References

1. Aljarah, I., Ala'M, A. Z., Faris, H., Hassonah, M. A., Mirjalili, S., & Saadeh, H. (2018). Simultaneous feature selection and support vector machine optimization using the grasshopper optimization algorithm. *Cognitive Computation*, *10*(3), 478-495.
2. Arora, S., & Anand, P. (2019). Binary butterfly optimization approaches for feature selection. *Expert Systems with Applications*, *116*, 147-160.
3. Arora, S., & Singh, S. (2017). An improved butterfly optimization algorithm with chaos. *Journal of Intelligent & Fuzzy Systems*, *32(1), 1079-1088*.
4. Arora, S., & Singh, S. (2019). Butterfly optimization algorithm: a novel approach for global optimization. *Soft Computing*, 23(3), 715-734.
5. Banu, P. N., & Andrews, S. (2015). Gene clustering using metaheuristic optimization algorithms. *International Journal of Applied Metaheuristic Computing (IJAMC)*, *6*(4), 14-38.
6. Battiti, R. (1994). Using mutual information for selecting features in supervised neural net learning. *IEEE Transactions on neural networks*, *5*(4), 537-550.
7. Bennasar, M., Setchi, R., & Hicks, Y. (2013). Feature interaction maximisation. *Pattern recognition letters*, *34*(14), 1630-1635.
8. Bergstra, J., & Bengio, Y. (2012). Random search for hyper-parameter optimization. *Journal of machine learning research*, *13*(2).
9. Bilewu, S. O., Sule, B. F., & Ayanshola, A. M. (2015). Optimum parameter selection for the morphometric description of watersheds: A case study of central Nigeria.
10. Bommert, A., Sun, X., Bischl, B., Rahnenführer, J., & Lang, M. (2020). Benchmark for filter methods for feature selection in high-dimensional classification data. *Computational Statistics & Data Analysis*, *143*, 106839.
11. Çavuşoğlu, Ü. (2019). A new hybrid approach for intrusion detection using machine learning methods. *Applied Intelligence*, *49*(7), 2735-2761.
12. Ch, R., & MAP, M. (1997). Bayesian learning. *book: Machine Learning. McGraw-Hill Science/Engineering/Math*, 154-200.
13. Chen, S., Chen, R., Wang, G. G., Gao, J., & Sangaiah, A. K. (2018). An adaptive large neighborhood search heuristic for dynamic vehicle routing problems. *Computers & Electrical Engineering*, 67, 596-607.
14. Chou, J. S., & Truong, D. N. (2021). A novel metaheuristic optimizer inspired by behavior of jellyfish in ocean. *Applied Mathematics and Computation*, 389, 125535.
15. El Akadi, A., El Ouardighi, A., & Aboutajdine, D. (2008). A powerful feature selection approach based on mutual information. *International Journal of Computer Science and Network Security*, *8*(4), 116.
16. Fathollahi-Fard, A. M., Hajiaghaei-Keshteli, M., & Tavakkoli-Moghaddam, R. (2020). Red deer algorithm (RDA): a new nature-inspired meta-heuristic. *Soft Computing*, 24(19), 14637-14665.
17. Ganganath, N., Cheng, C. T., & Chi, K. T. (2015, May). Distributed anti-flocking control for mobile surveillance systems. In *2015 IEEE International Symposium on Circuits and Systems (ISCAS)* (pp. 1726-1729). IEEE.
18. Gao, W., Hu, L., Zhang, P., & Wang, F. (2018). Feature selection by integrating two groups of feature evaluation criteria. *Expert Systems with Applications*, 110, 11-19.
19. Gehan, E. A. (1965). A generalized Wilcoxon test for comparing arbitrarily singly-censored samples. *Biometrika*, *52*(1-2), 203-224.
20. Ghosh, M., Malakar, S., Bhowmik, S., Sarkar, R., & Nasipuri, M. (2019). Feature selection for handwritten word recognition using memetic algorithm. In *Advances in intelligent computing* (pp. 103-124). Springer, Singapore.
21. Gu, Q., Li, Z., & Han, J. (2012). Generalized fisher score for feature selection. arXiv preprint arXiv:1202.3725.
22. Heidari, A. A., Mirjalili, S., Faris, H., Aljarah, I., Mafarja, M., & Chen, H. (2019). Harris hawks optimization: Algorithm and applications. *Future generation computer systems*, 97, 849-872.
23. Holland, J. H. (1992). Genetic algorithms. *Scientific american*, *267*(1), 66-73.
24. Hu, Z., Bao, Y., Xiong, T., & Chiong, R. (2015). Hybrid filter–wrapper feature selection for short-term load forecasting. *Engineering Applications of Artificial Intelligence*, *40*, 17-27.
25. Jin, Z., Zhou, G., Gao, D., & Zhang, Y. (2020). EEG classification using sparse Bayesian extreme learning machine for brain–computer interface. *Neural Computing and Applications*, *32*(11), 6601-6609.

26. Kennedy, J., & Eberhart, R. (1995, November). Particle swarm optimization. In *Proceedings of ICNN'95-international conference on neural networks* (Vol. 4, pp. 1942-1948). IEEE.
27. Kira, K., & Rendell, L. A. (1992). A practical approach to feature selection. *In Machine learning proceedings 1992* (pp. 249-256). Morgan Kaufmann.
28. Li, G., Wang, G. G., Dong, J., Yeh, W. C., & Li, K. (2021). DLEA: A dynamic learning evolution algorithm for many-objective optimization. *Information Sciences*, 574, 567-589.
29. Li, G., Wang, G. G., & Wang, S. (2021). Two-Population Coevolutionary Algorithm with Dynamic Learning Strategy for Many-Objective Optimization. *Mathematics*, 9(4), 420.
30. Li, W., & Wang, G. G. (2021). Elephant herding optimization using dynamic topology and biogeography-based optimization based on learning for numerical optimization. *Engineering with Computers*, 1-29.
31. Li, S., Chen, H., Wang, M., Heidari, A. A., & Mirjalili, S. (2020). Slime mould algorithm: A new method for stochastic optimization. *Future Generation Computer Systems*, 111, 300-323.
32. Luo, J., Chen, H., Xu, Y., Huang, H., & Zhao, X. (2018). An improved grasshopper optimization algorithm with application to financial stress prediction. *Applied Mathematical Modelling*, *64*, 654-668.
33. Masadeh, R., Mahafzah, B. A., & Sharieh, A. (2019). Sea lion optimization algorithm. *Sea*, 10(5).
34. Mirjalili, S. (2015). The ant lion optimizer. *Advances in engineering software*, 83, 80-98.
35. Mirjalili, S. (2016). SCA: a sine cosine algorithm for solving optimization problems. *Knowledge-based systems*, 96, 120-133.
36. Mirjalili, S. Z., Mirjalili, S., Saremi, S., Faris, H., & Aljarah, I. (2018). Grasshopper optimization algorithm for multiobjective optimization problems. *Applied Intelligence*, *48*(4), 805-820.
37. Murphy, P. M. (1994). UCI repository of machine learning databases. *ftp:/pub/machine-learning-databaseonics. uci. edu*.
38. Oliva, D., Hinojosa, S., Osuna-Enciso, V., Cuevas, E., Pérez-Cisneros, M., & Sanchez-Ante, G. (2019). Image segmentation by minimum cross entropy using evolutionary methods. *Soft Computing*, *23*(2), 431-450.
39. Peng, H., Long, F., & Ding, C. (2005). Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy. *IEEE Transactions on pattern analysis and machine intelligence*, *27*(8), 1226-1238.
40. Sadeghian, Z., Akbari, E., & Nematzadeh, H. (2021). A hybrid feature selection method based on information theory and binary butterfly optimization algorithm. *Engineering Applications of Artificial Intelligence*, *97*, 104079.
41. Simon, D. (2008). Biogeography-based optimization. *IEEE transactions on evolutionary computation*, *12*(6), 702-713.
42. Thangaiah, P. R., Shriram, R., Vivekanandan, K., & Reader, B. S. M. E. D. (2009). Adaptive hybrid methods for Feature selection based on Aggregation of Information gain and Clustering methods. *International Journal of Computer Science and Network Security*, *9*(2), 164-169.
43. Tubishat, M., Alswaitti, M., Mirjalili, S., Al-Garadi, M. A., & Rana, T. A. (2020). Dynamic butterfly optimization algorithm for feature selection. *IEEE Access*, *8*, 194303-194314.
44. Vapnik, V. (1992). Principles of risk minimization for learning theory. In *Advances in neural information processing systems* (pp. 831-838).
45. Vapnik, V. N. (1999). An overview of statistical learning theory. *IEEE transactions on neural networks*, *10*(5), 988-999.
46. Verleysen, M., & François, D. (2005, June). The curse of dimensionality in data mining and time series prediction. In *International work-conference on artificial neural networks* (pp. 758-770). Springer, Berlin, Heidelberg.
47. Wang, G. G. (2018). Moth search algorithm: a bio-inspired metaheuristic algorithm for global optimization problems. *Memetic Computing*, 10(2), 151-164.
48. Wang, G. G., Deb, S., & Coelho, L. D. S. (2018). Earthworm optimisation algorithm: a bio-inspired metaheuristic algorithm for global optimisation problems. *International journal of bio-inspired computation*, 12(1), 1-22.
49. Wang, G. G., Deb, S., & Coelho, L. D. S. (2015, December). Elephant herding optimization. *In 2015 3rd International Symposium on Computational and Business Intelligence (ISCBI)* (pp. 1-5). IEEE.
50. Wang, G. G., Deb, S., & Cui, Z. (2019). Monarch butterfly optimization. *Neural computing and applications*, 31(7), 1995-2014.
51. Wang, G. G., Wei, C. L., Wang, Y., & Pedrycz, W. (2021). Improving distributed anti-flocking algorithm for dynamic coverage of mobile wireless networks with obstacle avoidance. *Knowledge-Based Systems*, 225, 107133.
52. Whitley, D. (1994). A genetic algorithm tutorial. *Statistics and computing*, *4*(2), 65-85.
53. Xu, L., Tu, Y., & Zhang, Y. (2020). A grasshopper optimization-based approach for task assignment in cloud logistics. *Mathematical Problems in Engineering*, *2020*.
54. Yi, J. H., Deb, S., Dong, J., Alavi, A. H., & Wang, G. G. (2018). An improved NSGA-III algorithm with adaptive mutation operator for Big Data optimization problems. *Future Generation Computer Systems*, *88*, 571-585.
55. Yin, X., & Han, J. (2003, May). CPAR: Classification based on predictive association rules. In *Proceedings of the 2003 SIAM international conference on data mining* (pp. 331-335). Society for Industrial and Applied Mathematics.
56. Zhang, B., Yang, X., Hu, B., Liu, Z., & Li, Z. (2020). OEbBOA: A novel improved binary butterfly optimization approaches with various strategies for feature selection. *IEEE Access*, 8, 67799-67812.
57. Zhang, H., Wang, G. G., Dong, J., & Gandomi, A. H. (2021). Improved NSGA-III with Second-Order Difference Random Strategy for Dynamic Multi-Objective Optimization. *Processes*, 9(6), 911.
58. Zhang, J., Xiong, Y., & Min, S. (2019). A new hybrid filter/wrapper algorithm for feature selection in classification. *Analytica chimica acta*, *1080*, 43-54.
59. Zhang, Y., Li, H. G., Wang, Q., & Peng, C. (2019). A filter-based bare-bone particle swarm optimization algorithm for unsupervised feature selection. *Applied Intelligence*, *49*(8), 2889-2898.

Declaration of Interest

Anurag Tiwari Indian Institute of Technology (BHU),

Varanasi, India
Lab No. 9, Computer Science and Engineering Department

10-11-2021

Dear Prof. Binshan Lin
Editor-in-Chief,
Expert Systems with Applications

We wish to submit an original research article entitled "**A Hybrid Feature Selection Approach based on Information Theory and Dynamic Butterfly Optimization Algorithm for Data Classification**" for consideration by Expert Systems With Applications. We confirm that this work is original and has not been published elsewhere, nor is it currently under consideration for publication elsewhere.

This paper develops a novel hybrid method for dimension reduction using a **dynamic butterfly algorithm and a three-way interaction-based mutual information** paradigm. The proposed scheme maximizes both the **classification accuracy and interaction gain** among the most relevant features selected from the original feature set. The experimental results are validated on **twenty benchmark datasets** taken from the UCI repository.

This work is significant because the proposed work's findings are highly effective in producing promising results compared to various existing baseline feature selection methods. Also, the robustness and convergence rate of the proposed algorithm is superior to base models.
We believe that this manuscript is appropriate for publication by **Expert Systems With Applications** because it comes under the journal's Aims and Scope. The authors **declare** that they have no known competing financial **interests** or personal relationships that could have appeared to influence the work reported in this paper.

Thank you for your consideration of this manuscript.

Sincerely,
Anurag Tiwari (**Corresponding Author**)
anuragtiwari.rs.cse17@itbhu.ac.in

ORCID ID Information
Author 1 (Corresponding Author): Anurag Tiwari **ORCIDID:** 0000-0002-4772-6672
Author 2: Amrita Chaturvedi **ORCIDID:** 0000-0002-3345-5103

CRediT author statement

Anurag Tiwari: Conceptualization, Methodology, Software, Data curation, Writing- Original draft preparation, Software, Validation, Visualization, Writing- Reviewing and Editing.

Amrita Chaturvedi: Supervision, Investigation.

Highlights

1. A hybrid feature selection is proposed for selecting relevant attributes.
2. Dynamic Butterfly algorithm is used for search space optimization.
3. Our aim was balancing the trade-off between exploration and exploitation.
4. Twenty datasets and ten algorithms are used in results comparison.
5. Experimental results confirmed the superiority of our method.