Human-Centric Intelligent Systems (2023) 3:558–587
https://doi.org/10.1007/s44230-023-00048-w

RESEARCH ARTICLE

Efficient Feature Selection in High Dimensional Data Based on Enhanced Binary Chimp Optimization Algorithms and Machine Learning

Farid Ayeche[1] · Adel Alti[2,3]

Received: 13 June 2023 / Accepted: 9 October 2023 / Published online: 2 November 2023

Abstract

Feature selection with the highest performance accuracy is the biggest win for multidimensional data. The Chimpanzee Optimization Algorithm (ChOA) serves as a crucial technique for dealing with multidimensional global optimization issues. However, ChOA often lacks fast convergence and good selection of sensitive attributes leading to poor performance. To address these issues, most significant features were selected using two variants of ChOA called BChimp1 and BChimp2 (BChimp1 and BChimp are available at : https://www.mathworks.com/matlabcentral/fileexchange/133267-binary-chimpoptimization-algorithm-forfeatures-selection. September 22, 202). BChimp1 selects the optimal solution from the four best possible solutions and it applies a stochastic crossover on four moving solutions to deeply speed-up convergence level. BChimp2 uses the sigmoid function to select the significant features. Then, these features were trained using six-well known classifiers. The proposed techniques tend to select the most significant features, speed up the convergence rate and decrease training time for high-dimensional data. 23 standard datasets with six well-known classifiers were employed to assess the performance of BChimp1 and BChimp2. Experimental results validate the efficiency of BChimp1 and BChimp2 in enhancing accuracy by 83.83% and 82.02%, and reducing dimensionality by 42.77% and 72.54%, respectively. However, time-evaluation results of BChimp1 and BChimp2 in all datasets showed fast convergence and surpassed current optimization algorithms such as PSO, GWA, GOA, and GA.

Keywords Feature selection · BChimp · Machine learning · Dimensionality reduction · Relevancy · Classification accuracy

Abbreviations

BChimp 1	First Binary Chimp Optimization Algorithm
BChimp 2	Second Binary Chimp Optimization Algorithm
BMOChOA	Binary Multi-Objective Chimpanzee Optimization Algorithm
BPSO	Binary Particle Swarm Optimization
CEE	Classification Error Rate
ChOA	Chimpanzee Optimization Algorithm
DCNN	Deep Convolutional Neural Network
DPSO	Distributed Particle Swarm Optimization
DT	Decision Tree
Fuzzy-ChOA	Fuzzy Chimpanzee Optimization Algorithm
GA	Genetic Algorithm
GS	Greedy Search
GWO	Grey Wolf Optimization
HHO	Harris Hawk Optimization
HLBDA	Hyper Learning Binary Dragonfly Algorithm
KNN	K-Nearest Neighbor
MO-ChOA	Multi-Objective Chimpanzee Optimization Algorithm
MRMR	Minimum Redundancy Maximum Relevance
OBL	Opposition-Based Learning

✉ Adel Alti
a.alti@qu.edu.sa

Farid Ayeche
ayeche_farid@univ-setif.dz

[1] Faculty of Technologies, Technology Department, LMETR, University Ferhat Abbas Sétif-1, Sétif, Algeria

[2] Department of Management Information Systems, College of Business Qassim University, Buraydah, Kingdom of Saudi Arabia

[3] Faculty of Sciences, Computer Science Department, LRSD, University Ferhat Abbas Sétif-1, Sétif, Algeria

PSO	Particle Swarm Optimization
QDA	Quadratic Discriminant Analysis
RF	Random Forest
SVM	Support Vector Machine
SD	Standard Deviation
WOA	Whale Optimization Algorithm

1 Introduction

Data mining serves as a crucial approach for extracting meaningful information and knowledge. This would help in making the right decisions based on high-quality information models. As soon as the database models lead to an increased number of features (relevant, irrelevant, redundant, and noisy information) [1], essential concerns arise regarding the degradation of machine learning performance [2]. A huge number of attributes can result in increased computational complexity, high execution time and low precision. Here, we have faced a phenomenon called the curse of dimensionality. To deal with all these challenges, Feature Selection (FS) is considered the most effective solution. In recent times, feature selection has attracted considerable attention, because of its capability in selecting the most relevant features from higher-dimension [3]. Feature selection includes two categories of techniques, filtering, and wrapping [4–6]. The filtering technique [5] is simple and less expensive numerical computations based on statistical analysis and mutual information. The filtering technique is independent of learning algorithms and only requires relationships between inputs and outputs. The wrapper technique related to learning algorithms when it comes to determining significant features but requires a high computational cost compared to filtering techniques [6]. However, the more accurate results of this technique can result in high interest to academics and practitioners [7].

Feature selection is NP-hard problem derived from the exponential increase in the number of attributes. The complex nature of relations between features should not be neglected, especially with the growth of features. Although this parameter must be applied in the realm of feature selection applications [8]. It is noted that these applications often rely heavily on metaheuristic optimization techniques [9–15] to achieve higher classification accuracy based on genetic and chimp optimization algorithms. Such methods have been applied to increase performance. Unfortunately, they present a serious weakness, especially in terms of getting an optimum consensus between classification accuracy, optimal number of features, and fast computational time. To fill these gaps, we look for another solution that provides efficient optimization techniques with a high accuracy rate and acceptable execution time for various domain applications. We believe that machine learning can fill these gaps.

This paper proposes an efficient feature selection technique for reducing dimensionality and improving classification accuracy with acceptable execution time by using two variants of Binary Chimp optimization algorithms (BChimp1 and BChimp2) and six well-known classifiers (KNN, DT, SVM, RF, QDA, and MLP). All tests are applied on 23 different datasets [47] to evaluate the classification accuracy and execution time. Overall, we make the following contributions:

- Two variants of binary chimp optimization algorithms named BChimp1 and BChimp2 are proposed to select optimal features in multidimensional datasets and machine learning models are established both for training and testing steps.
- To enhance the quality of the classified features with acceptable execution time, a machine learning model is developed. The selected features have been validated using the most known classifiers such as KNN, DT, SVM, RF, QDA, and MLP.
- An evaluation of the proposed methods (BChimp1-ML and BChimp1-ML) on 23 datasets [47] of different features in various domains (19 of which are high-dimensional) to ensure optimal feature selection and acceptable execution time. However, the two proposed methods have been compared with GWO, WOA, GA, and PSO.

The rest of the paper is organized as follows. Section 2 discusses the related work of this study. Section 3 presents an overview of feature selection and Chimp Optimization Algorithm (ChOA). Section 4 specifies the proposed hybrid approach. Section 5 provides an experimental demonstration of the proposed framework. Section 6 provides a summary of the paper.

2 Related Works

There are two manners for selecting features: filtering and wrapping. The filtering approach directly extracts relevant features based on their correlation with the dependent variable. The wrapper approach determines optimal features based on classifier models. Our work is based on the wrapper feature selection approach.

2.1 Filtering Based Approaches

This section discusses some recent and relevant filtering-based techniques presented in the literature [7, 16–18]. Peng et al. [7] integrate mutual criteria such as maximum dependencies, maximum relevancies, and minimum redundancies (MRMR). Wang et al. [16] proposed a feature selection approach based on statistical tests. Gao et al. [17] designed a feature selection algorithm combining significant features. Labani et al. [18] proposed a new multiple-feature filtering method based on text classification algorithms.

2.2 Wrapper Based Approaches

Wrapper approaches are divided into 1) evolutionary algorithms that rely on evolution mechanisms such as mutation and crossing, 2) swarm intelligence that relies on the principle and effects of some animals or insects, and 3) physics-based algorithms are physical phenomena highly sensitive to initial conditions and natural behaviors [19] and 4) deep learning-based algorithms.

2.2.1 Evolutionary Algorithms

Genetic Algorithm (GA) is an efficient and useful metaheuristic optimization approach based on the evolution of genetic theory. They are coded using a problem-specific fitness function [20]. GA allows for exploring high-dimensional search space efficiently [21]. But still, there is a drawback in the GA that it suffers from high computational complexity and random selection of mutation and selection processes, which greatly affect the feature selection process [22]. In [23], Siedlecki et al. apply GA for selecting optimal features for the first time. Subsequently, Huang et al. [21] used the GA in feature selection to improve the overall effectiveness of SVM. Nemati et al. [24] proposed an optimization technique based on genetic and ant colony algorithms to enhance accuracy. The optimization is done by adopting the protein-based function. Stefano et al. [25] used genetic feature optimization algorithm in large text data. Rejer et al. [26] introduced a powerful GA algorithm based on mutation to optimize features for brain-machine interfaces. Jingwei et al. [27] employed a combination of the selection and the crossover operators to optimize the features in the multidimensional databases.

2.2.2 Swarm Intelligence Algorithms

Kennedy et al. [28] proposed simple and effective algorithms called Particle Swarm Optimization (PSO) for the problem of feature optimization. However, it is noted that PSO suffers from staying in the local minima. To address this issue, various variants of PSO are proposed. A binarization of PSO (BPSO) is applied by Chuang et al. [29] to genes in order to attain the ultimate performance. The proposed algorithm focused on achieving early convergence of PSO by changing the local search space if it did not find the optimal solution after three iterations. Unler et al. [30] proposed a novel DPSO based on probability-weighted estimation to improve selection. Moreover, to enhance effectiveness and solution quality, Ayeche et al. [31] proposed micro-GA for facial expression recognition based on swarm optimization algorithms. Tan et al. [32] designed a novel efficient attribute selection algorithm using PSO to improve skin cancer detection. Too et al. [33] proposed BPSO to select significant attributes. First, the population is divided according to the weight-based segmentation strategies and some important individuals are incorporated into the new population in order to enhance the diversity. Moreover, to enhance effectiveness and adaptability, Xue et al. [34] proposed an adaptive average parameter PSO approach for complex applications for attribute selection with multiple classifiers. Al-Tashi et al. [12] designed attributes selection scheme based on multi-objectives function and grey wolf optimization. Too et al. [35] also proposed a new quadratic binary Harris Hawk Optimization (HHO) for feature selection. Moghaddam et al. [36] presented a novel multi-objective forest optimization for attribute selection.

Too et al. [13], introduced Hyper Learning Binary Dragonfly Algorithm (HLBDA) capable of determining significant features in COVID-19 datasets. Hegazy et al. [37] developed the Chaotic Salp Swarm Algorithm (CSSA) that enhances the performance of the data classification, which significantly reduces the high number of attributes. Niu et al. [38] proposed a multi-objective fitness selection approach based on bacterial foraging optimization. In addition, Piri et al. [39] used a multi-objective retrieval approach based on Ant Lion optimization to enhance efficiency.

Recently Jayashree et al. [49] proposed an efficient selection method utilizing Binary Multi-Objective ChOA (BMOChOA) and KNN classifier. BMOChOA and KNN allow the selection of relevant features from medical data. They evaluated their approach with four multi-objective performance evaluators and 14 medical databases of different sizes. The significant gain in this approach is a reduced number of features and enhanced classification accuracy. However, the authors did not achieve optimization of features and not evaluate execution time.

In [50], ChoA is subjected to binarization and transfer functions. They are applied with a crossover operator to yield the excellent exploratory behavior of the ChOA. This approach ensures efficient and direct selection of significant features. However, the work presents some weaknesses versus multidimensional data and consumes much execution time.

2.2.3 Physics-Based Algorithms

Numerous algorithms are proposed such as Henry's law-based algorithm for optimizing solubility [40]. Kaveh et al. [41] employed the physical law of Newton and Coulomb to select significant attributes. Desuky et al. [42] recently introduced a new Archimedes optimization algorithm to select critical features for data classification.

2.2.4 Deep Learning Algorithms

Khishe et al. [51] proposed an improved ChOA to automatically discover optimal DCNN architectures. The proposed model introduced three enhancements namely, digitized-based coding for easier encoding, variable-length DCNNs, and using only part of the dataset for fitness assessment. The proposed model outperformed 23 classifiers on 9 benchmark datasets. It achieved the best performance in the Fashion dataset (5.08%) and ranked second in terms of size (750k parameters). In 87 out of 95 investigations, it surpassed other benchmarks, making it the first ChOA-based effort to autonomously evolve DCNN architectures.

Cai et al. [52] introduce deep neural networks based ChOA capable of diagnosing COVID-19 through X-ray images. The proposed approach can ensure fast training and parallel implementation. A COVID-Xray-5K and COVIDetectioNet databases are chosen to enhance its effectiveness. This approach resulted in superior performance with 99.11% of accuracy, in contrast with other standard DCNN, DCNN-GA, and MSAD models. In addition, the work identified infected areas by activation map with high accuracy values.

In [53], authors designed a features-reducing algorithm combining Chimp Optimization Algorithm (ChOA), Greedy Search (GS), and Opposition-Based Learning (OBL) to select the most significant features. The GSOBL-ChOA approach was evaluated in different benchmark datasets for various engineering fields. The authors demonstrate potential for GSOBL-ChOA approach for applications in attributes selection, it is outperformed other existing benchmarks in most experimental cases.

Khishe et al. [54] proposed a new variation of ChOA namely the Multi-Objective ChOA (MO-ChOA) addressing multidimensional data optimization challenges in engineering problems. This work maintains a leader selection by applying distributed mechanism to efficiently explore the search space and keep non-dominated results. The authors show that the obtained results can achieve better performance by light-weighting structure compared to some existing algorithms.

Saffari et al. [55] use ChOA as an artificial trainer model to identify and classify mammals which is a high-dimensional task. The approach is based on Fuzzy-ChOA with controlled parameters to balance exploration and exploitation. Experimental datasets of underwater marine mammal sounds are collected, pre-processed, and used for ANN classification. The proposed approach yields remarkable results regarding the convergence speed and recognition performance compared with various benchmark algorithms.

3 Background Techniques

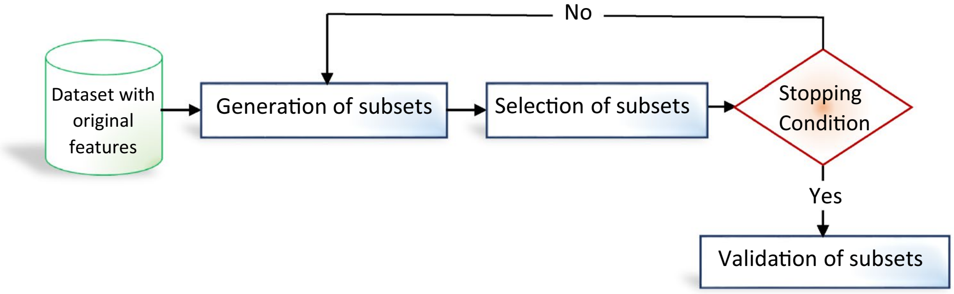
3.1 Feature Selection

In feature selection, important features are filtered and selected. It is based on the reduction and/or elimination of redundant, unnecessary, and irrelevant features. This technique is necessary to ensure better data understanding and reduce processing time. As opposed to other approaches, the feature selection approach is predictable. This approach involved four-phase process as shown in Fig. 1: 1) generation of subsets, 2) evaluation of subsets, 3) stopping criteria and 4) validation of results [3]. The first phase consists of adopting a search strategy to select a list of candidates. The second phase consists of assessing a list of candidates using an evaluation criterion. Depending on the stopping criterion (phase 3), phases 1 and 2 will be repeated until the best combination of feature subsets is determined. Finally, the selected subset is validated with an independent data set or using prior domain-specific knowledge according to the task performed.

3.2 Chimp Optimization Algorithm (ChOA)

The ChOA was proposed by Khishe et al. [14] in 2020 which is based on the chimp's hunting behavior that is guided and controlled by their intelligence and sexual motivations as shown in Fig. 2. ChOA plays a significant role in choosing important features is that it has excellent global and local discovery [14] used for optimization in the opposite of the other existing algorithms. Since they're easier to implement, they may prove to be more useful with higher stability and low feature assessment than PSO

Fig. 1 Feature selection process



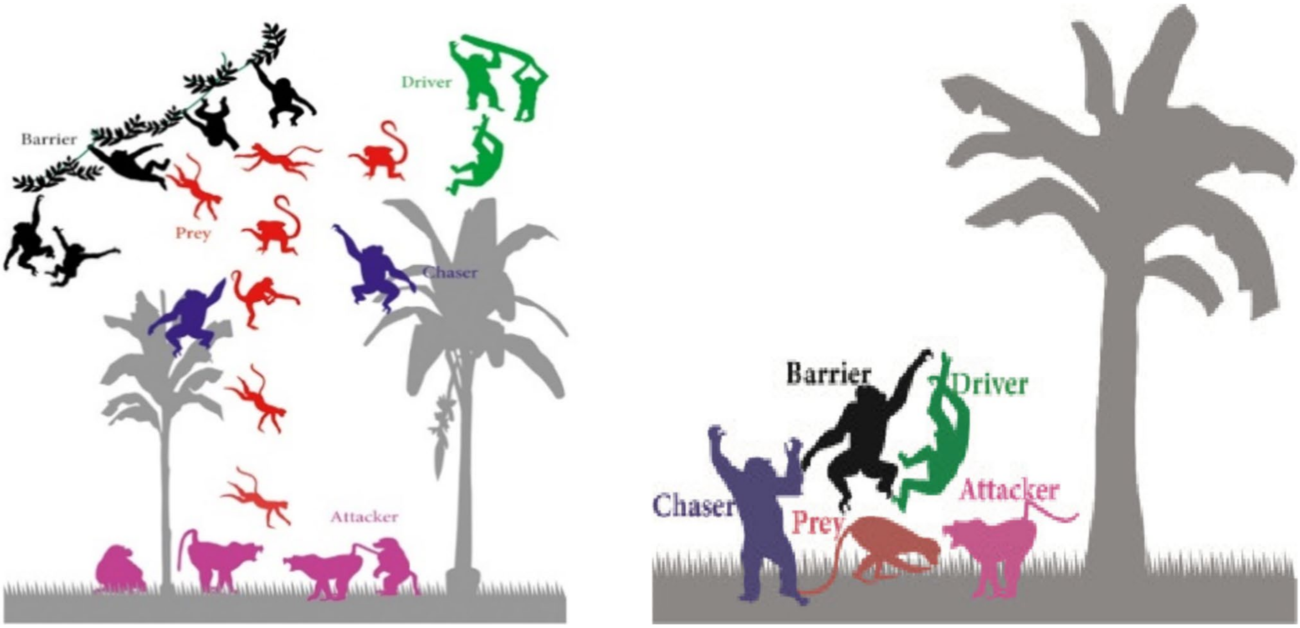


Fig. 2 The hunting process of chimps (exploration and exploitation)

or GA, moreover, as an optimization approach, it does not need many initial parameters to ensure optimal convergence. In a chimpanzee colony, individuals are not all of the same capacity. It can use its appropriate strategy to predict the location of prey. Such diversity and difference are very essential to the success of the hunting process. Chimpanzees live in groups that consist of *Attackers*, *Barriers*, *Chasers*, and *Drivers*. Drivers follow and surround prey without attacking it. They drive it toward the barriers. The barriers on their side build a dam along the path of the prey, eventually trapping it in a tree. The chaser's mission is to move quickly after pouncing on the prey to cause panic and catch up. Finally, the attackers predict the prey's escape way, either catching it or pushing it toward the hunters.

The ChOA technique represents driving and chasing the prey as follows:

$$X(t+1) = X_p(t) - A.D \tag{1}$$

$$With D = \left| C.X_p(t) - M.X(t) \right| \tag{2}$$

where

t : current iteration;
X: current position of chimp;
X_p: vector of prey position;
A : coefficient of prey;
D : travel distance that separates the prey and the chimp.
C, M are vectors of coefficients.

The vectors of coefficients A, C, and f are calculated using equations (3)–(5):

$$\begin{aligned} A_1 &= 2.f.r_{11} - f \\ A_2 &= 2.f.r_{21} - f \\ A_3 &= 2.f.r_{31} - f \\ A_4 &= 2.f.r_{41} - f \end{aligned} \tag{3}$$

$$\begin{aligned} C_1 &= 2.r_{12} \\ C_2 &= 2.r_{22} \\ C_3 &= 2.r_{32} \\ C_4 &= 2.r_{42} \end{aligned} \tag{4}$$

where f decreases nonlinearly from 2 to 0, as computed by equation (6):

$$f = 2 - 2\left(\frac{t}{T}\right) \tag{5}$$

r_1, r_2 are random numbers in [0, 1] which are calculated as follows:

$$
\begin{aligned}
c_1g_1 &= 1.95 - 2.\left(\frac{t^{\frac{1}{3}}}{T^{\frac{1}{3}}}\right), r_{11} = c_1g_1.\ \mathrm{rand}()\\
c_2g_1 &= 2.\left(\frac{t^{\frac{1}{3}}}{T^{\frac{1}{3}}}\right) + 0.5, r_{12} = c_2g_1.\ \mathrm{rand}()\\
c_1g_2 &= 1.95 - 2.\left(\frac{t^{\frac{1}{3}}}{T^{\frac{1}{3}}}\right), r_{21} = c_1g_2.\ \mathrm{rand}()\\
c_2g_2 &= 2.\left(\frac{t^{\frac{1}{3}}}{T^{\frac{1}{3}}}\right) + 0.5, r_{22} = c_2g_1.\ \mathrm{rand}()\\
c_1g_3 &= -2.\left(\frac{t^3}{T^3}\right) + 2.5, r_{31} = c_1g_3.\ \mathrm{rand}()\\
c_2g_3 &= 2.\left(\frac{t^{\frac{1}{3}}}{T^{\frac{1}{3}}}\right) + 0.5, r_{32} = c_2g_3.\ \mathrm{rand}()\\
c_1g_4 &= -2.\left(\frac{t^3}{T^3}\right) + 2.5, r_{41} = c_1g_4.\ \mathrm{rand}()\\
c_2g_4 &= 2.\left(\frac{t^{\frac{1}{3}}}{T^{\frac{1}{3}}}\right) + 0.5, r_{42} = c_2g_4.\ \mathrm{rand}()
\end{aligned}
\tag{6}
$$

where *rand*() is the standard function distribution with a scale of 0 to 1.

In Equation (2), the variable M represents a chaotic value ranging from 0 to 1, which is obtained from one of the chaotic maps listed below:

$$
\begin{aligned}
&Quadratic : M = x_{(i+1)} = x_i^2 - c, c = 1\\
&Gaussian : M = x_{(i+1)} = \begin{cases} 1, & ifxi = 0\\ \dfrac{1}{\mathrm{mod}\left(x_i, 1\right)} & else \end{cases}\\
&Logistic : M = x_{(i+1)} = \alpha x_i(1 - x_i), \alpha = 4\\
&Bernoulli : M = x_{(i+1)} = 2x_i mod 1
\end{aligned}
\tag{7}
$$

where t, T are the current iteration, and the maximum number of iterations, respectively.

To ensure a hybrid efficient search system, the ChOA uses four independent groups, and two main strategies have been proposed named ChOA1 and ChOA2 to achieve superior performance in the classification accuracy, irrespective of the optimization problem. In ChOA, the first task of the attacker is to inform the driver, barrier, and chaser about the location of prey. Then the best leaders guide other chimps toward the prey using the best four locations. The new position of the chimp can be expressed as:

$$X(t+1) = \frac{X_1 + X_2 + X_3 + X_4}{4} \tag{8}$$

Where X_1, X_2, X_3andX_4 are the positions calculated using Eq. (9)–(12) respectively.

$$\mathrm{X_1 = |X_a - A_1.D_a|} \tag{9}$$

$$\mathrm{X_2 = |X_b - A_2.D_b|} \tag{10}$$

$$\mathrm{X_3 = |X_c - A_3.D_c|} \tag{11}$$

$$\mathrm{X_4 = |X_d - A_4.D_d|} \tag{12}$$

where

X_a, X_b, X_c and X_d are the positions of chimps respectively at iteration t;
$\mathrm{A_1}$, $\mathrm{A_2}$, $\mathrm{A_3}$ and $\mathrm{A_4}$ are the coefficient vectors calculated by equations (3).
$\mathrm{D_a}$, $\mathrm{D_b}$, $\mathrm{D_c}$ and $\mathrm{D_d}$ calculated using Eq. (13)–(16).

$$\mathrm{D_a = |C_1.X_a - M_1.X|} \tag{13}$$

$$\mathrm{D_b = |C_2.X_b - M_2.X|} \tag{14}$$

$$\mathrm{D_c = |C_3.X_c - M_3.X|} \tag{15}$$

$$\mathrm{D_d = |C_4.X_d - M_4.X|} \tag{16}$$

where

$\mathrm{C_1}$, $\mathrm{C_2}$, $\mathrm{C_3}$ and $\mathrm{C_4}$ are calculated using equation (4).
$\mathrm{M_1}$, $\mathrm{M_2}$, $\mathrm{M_3}$ and $\mathrm{M_4}$ are the coating functions.

According to equations (7) to (15), chimpanzees constantly update their positions based on the locations of their group leaders, which represent the best positions across the population. The distance *D* between a chimpanzee and its prey determines the area of exploration, and parameters *A* and *C* determine its search areas. This variable *m* refers to chimpanzee behavior during the final stages of the hunting process. Sexual motivations drive them to temporarily ignore their hunting tasks to concentrate solely on obtaining meat, symbolizing their dominance. This erratic and impulsive behavior helps them to explore new approaches and overcome challenges related to finding prey, putting their ingenuity to the test. A vital phase of the ChOA (Chimpanzee Optimization Algorithm) involves chimpanzees utilizing their chaotic behavior to overcome optimization problems. Their position update method is highly flexible, providing a significant advantage over local optima traps and resolving otherwise insurmountable complex problems. In this innovative approach, chimpanzees demonstrate how they optimize their path toward more promising solutions by incorporating a controlled chaotic dimension into their optimization process. By harnessing this clever strategy, they offer valuable perspectives for resolving complex problems across various scientific and engineering domains.

Algorithm 1 is an illustration of the steps of ChOA to compute the optimal position of the attacker. In this algorithm, a population is used to initialize the positions of the first four chimpanzees: $X_{Attacker}$, $X_{Barrier}$, X_{chaser}, and the X_{Driver} as well as their scores to zero. It was evaluated based on the fitness function. To find the optimal solution, the location positions and scores of the four main chimps are updated. Also, coefficients f, r_1 and r_2 are updated using Eq. 6 and the coefficients A and C are computed based on f, r_1 and r_2 using Eq. 3 and Eq. 4 respectively. The chimpanzee's new positions are updated using Eq. 6. Finally, the optimal binary solution of X_{best} is returned.

Generally, the Chimp Optimization Algorithm is used to solve so-called continuous optimization problems, in which chimpanzees change their positions at any point in space. However, there are some optimization issues in which the solutions are bit sequences of 0 or 1 such as feature selections. We then need a binary version of ChOA. In this work, two novel binary wolf optimizations named BChimp1 and BChimp2 are proposed to improve selection accuracy and convergence rate. The updating of the positions of the chimpanzees is done using the positions of the first four chimpanzees. The goal is to attract each chimpanzee in the flock to the first four leaders. At any time, the set of solutions is binary, located at a hypercube's corner. In other words, change the locations of a particular chimpanzee with respect of BChoA's principles, while maintaining the binary constraints established by equation (7).

In the following subsections, we will describe the two approaches proposed to achieve the binary aspect of the chimpanzee optimization algorithm.

4 Our Proposal

The proposed system can select important features based on enhanced binary Chimp optimization algorithms and machine-learning classifiers. To develop our proposal, we used several classifiers, objective functions, and optimization methods to choose the optimal features. We have combined two variants of ChoA as well as six standard classifiers to ensure a good balance between relevance, convergence level, and training time. Figure 3 describes in detail the suggested system, which consists of two key stages: dimensionality reduction and machine

```
Inputs :   — $n$: Size of population. $It\_max$: Maximum number of iterations.
Outputs :  — $X_{best}$ : Optimal chimpanzee position. $fitness\ X_{best}$ : The best value of fitness
Begin
    Initialize the positions of the first four chimpanzees and their scores to 0.
    Initialize each chimp's positions ($X_i \;\; i = 1..n$);
    Evaluate fitness of each chimp;
    While (t < $It\_max$ )
        $X_{Attacker}$ ← Position of best chimp
        $X_{Barrier}$ ← Position of second-best chimp
        $X_{chaser}$ ← Position of third-best chimp
        $X_{Driver}$ ← Position of fourth-best chimp
        Calculate $f$ using Eq. 5
        Calculate $A_j$ and $C_j$ using Eq. 3 and Eq. 4
        $M$ = chaotic_value using Eq. 7
        For $i$ = 1 to $n$ Do
            For $d$ = 1 to $Dimension$ Do
                Calculate $D_1$, $D_2$, $D_3$ and $D_4$using Eq. 9-12
                Calculate $X_1$, $X_2$, $X_3$ and $X_4$using Eq. 13-16
                Calculate the position of chimp $i$ using Eq. 8
            End For
        End For
        $t = t + 1$
    End While
End
Return $X_{best}$ ($X_{Attacker}$) and $fitness\ X_{best}$
```

Algorithm 1: Pseudo-code of original ChOA

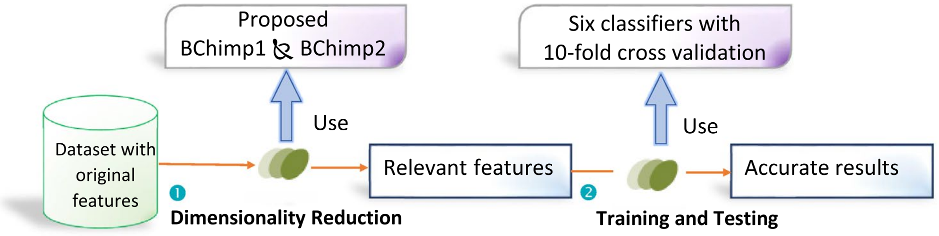


Fig. 3 Overall process of binary Chimp feature selection and machine learning approach

learning. The first stage consists of applying two enhanced binary chimp optimization algorithms to ensure dimensionality reduction. The second phase consists of applying six well-known classifiers to achieve high level of accuracy.

4.1 BChimp1: First Binary Chimp Optimization Algorithm

We propose BChimp1, a novel binary approach that builds on the principle of ChoA [14]. It investigates traces of movements of an *Attacker, Driver*, *Barrier*, and *Chaser*. They are jointly collaborated during the hunting phase to find potential prey.

4.1.1 Mathematical Formalization of BChimp1

The position of chimpanzee in BChimp1 is updated using the crossover operator expressed as:

$$X(t+1) = Crossover(Y_1, Y_2, Y_3, Y_4) \tag{17}$$

where Y_1, Y_2, Y_3 and Y_4 are the binary vectors representing the traces of movements of the first four chimpanzees. $Crossover(Y_1, Y_2, Y_3, Y_4)$ is the crossover operation between solutions. The vectors Y_1, Y_2, Y_3 and Y_4 are the binary vectors affected by the movement of the first four champs, respectively, which can be calculated as:

$$Y_1^d = \begin{cases} 1, if\left(X_A^d + bstep_A^d\right) \geq 1 \\ 0, otherwise \end{cases} \tag{18}$$

where X_A^d is the position vector of an attacker and the search space has the dimension d and $bstep_A^d$ is the binary step that is calculated by equation (19) as follows:

$$b\,step_A^d = \begin{cases} 1, if\, cstep_A^d \geq r_3 \\ 0, otherwise \end{cases} \tag{19}$$

where r_3 is a random vector with values between 0 and 1 and $cstep_A^d$ is continuous-valued step size, as defined by equation (20) as follows:

$$c\,step_A^d = \frac{1}{1 + exp(-10(A_1^d.D_A^d - 0.5))} \tag{20}$$

where A_1^d and D_A^d are calculated by equations (3) and (12).

$$Y_2^d = \begin{cases} 1, if\left(X_B^d + bstep_B^d\right) \geq 1 \\ 0, otherwise \end{cases} \tag{21}$$

Where X_B^d is the position vector of the barrier and the search space has the dimension d and $bstep_B^d$ is the binary step that is calculated by equation (22) as follows:

$$bstep_B^d = \begin{cases} 1, if\, cstep_B^d \geq r_4 \\ 0, otherwise \end{cases} \tag{22}$$

Where r_4 is a random vector with values between 0 and 1 and $cstep_B^d$ is continuous-valued step size, as defined by equation (23) as follows:

$$cstep_B^d = \frac{1}{1 + exp(-10(A_2^d.D_B^d - 0.5))} \tag{23}$$

where A_2^d and D_B^d are calculated by equations (3) and (13).

$$Y_3^d = \begin{cases} 1, if\left(X_C^d + bstep_C^d\right) \geq 1 \\ 0, otherwise \end{cases} \tag{24}$$

where X_C^d is the chaser's position vector and the search space has the dimension d and $bstep_C^d$ is the binary step that is calculated by equation (25) as follows:

$$b\,step_c^d = \begin{cases} 1, if\, cstep_C^d \geq r_5 \\ 0, otherwise \end{cases} \tag{25}$$

where r_5 is a random vector with values between 0 and 1; $cstep_C^d$ is continuous-valued step size, as defined by equation (26) as follows:

$$c\,step_C^d = \frac{1}{1 + exp(-10(A_3^d.D_C^d - 0.5))} \tag{26}$$

where A_3^d and D_C^d are calculated by equations (3) and (14).

$$Y_4^d = \begin{cases} 1, if\left(X_D^d + bstep_D^d\right) \geq 1 \\ 0, otherwise \end{cases} \tag{27}$$

where X_D^d is the driver's position vector and the search space has the dimension d and $bstep_D^d$ is the binary step that is calculated by equation (28) as follows:

```
Inputs :   — $n$ : size of population.
           — $It$ : current iteration. $It\_max$ : maximum number of iterations
Outputs :  — $X_{best}$ : optimal chimpanzee binary position
           — $fitness\ X_{best}$ : the best value of fitness
Begin
    Randomly initialize each chimp's positions ($X_i\ \ i = 1..n$);
    Initialize parameters $f$, $m$, $a$, and $c$, it = 0;
    Evaluate the fitness of each chimp;
    $X^d_{Attacker}$ ← Position of best chimp
    $X^d_{Barrier}$ ← Position of second-best chimp
    $X^d_{Chaser}$ ← Position of third-best chimp
    $X^d_{Driver}$ ← Position of fourth-best chimp
    For it= 1 to $It\_max$ Do
        For i= 1 to $n$ Do
            Compute $Y_1$ using Eq. 18
            Compute $Y_2$ using Eq. 21
            Compute $Y_3$ using Eq. 24
            Compute $Y_4$ using Eq. 27
            $X_{i\ new}$ ← $\mathrm{Crossover}(Y_1, Y_2, Y_3, Y_4)$
        End For
        Evaluate the fitness of each chimp;
        Update $X^d_{Attacker}$,$X^d_{Barrier}$, $X^d_{Chaser}$,$X^d_{Driver}$
        Update$f$, $m$, $a$, and $c$
    End For
    $X_{best} = X^d_{Attacker}$
    $fitness\ X_{best} = f(X_{best})$
End
Return $X_{best}$ and $f(X_{best})$
```

Algorithm 2: First Binary Chimp Optimization Algorithm

$$bstep^d_D = \begin{cases} 1, if\, cstep^d_D \geq r_6 \\ 0, otherwise \end{cases} \tag{28}$$

where r_6 is a random vector with values between 0 and 1 and $cstep^d_D$ is continuous-valued step size, as defined by equation (29) as follows:

$$cstep^d_D = \frac{1}{1 + exp(-10(A^d_4.D^d_D - 0.5))} \tag{29}$$

where A^d_D and D^d_D are calculated by applying equations (3) and (15).

4.1.2 Position Updating Mechanism in BChimp1

The Chimp updating position is the crossing function of four vectors Y^d_1, Y^d_2, Y^d_3 and Y^d_4.The mathematical formula of the new Chimp position is expressed by

$$X^d(t+1) = \begin{cases} Y^d_1, r_7 < \frac{1}{4} \\ Y^d_2, \frac{1}{4} \leq r_7 < \frac{1}{2} \\ Y^d_3, \frac{1}{2} \leq r_7 < \frac{3}{4} \\ Y^d_4, otherwise \end{cases} \tag{30}$$

Where r_7 is a random vector with values between 0 and 1.

4.1.3 Algorithm of BChimp1

The BChimp1 approach consists mainly of reducing a dimension D by choosing relevant features. The binary representation method has come up with the solution by translating the continuous value to a binary value of 0 or 1. Each dimension d corresponds to the index of a represented feature. The feature i is selected if the position of the chimp is greater than 0.5, which can be expressed as:

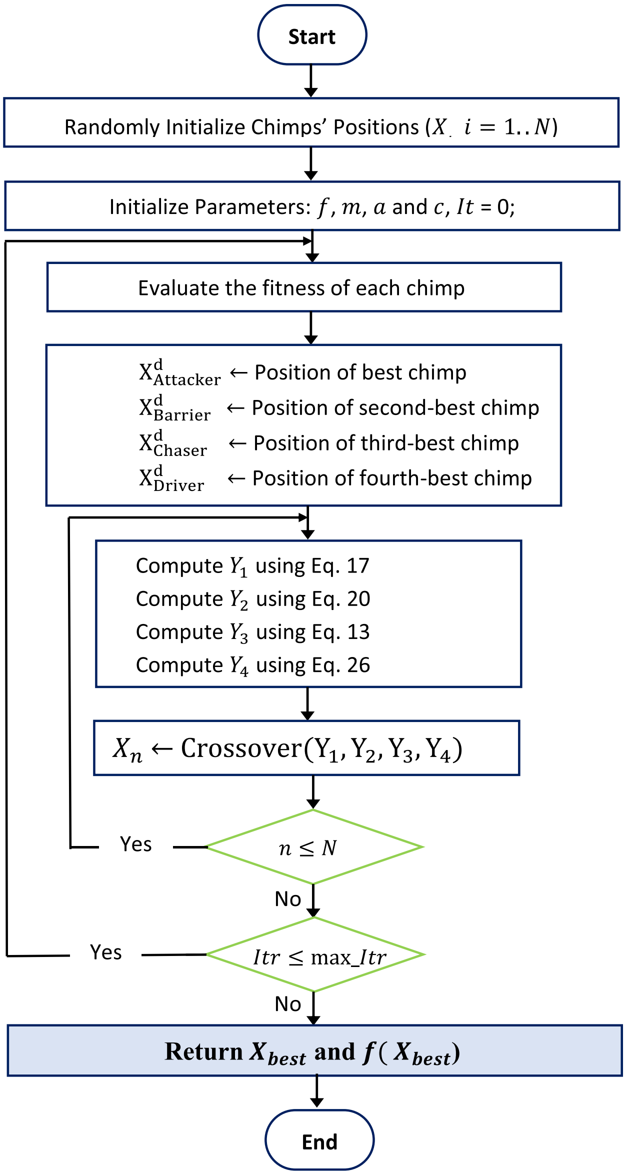

Start
Randomly Initialize Chimps' Positions ($X_{.}\ i = 1..N$)
Initialize Parameters: f, m, a and c, It = 0;
Evaluate the fitness of each chimp
X^{d}_{Attacker} ← Position of best chimp
X^{d}_{Barrier} ← Position of second-best chimp
X^{d}_{Chaser} ← Position of third-best chimp
X^{d}_{Driver} ← Position of fourth-best chimp
Compute Y_1 using Eq. 17
Compute Y_2 using Eq. 20
Compute Y_3 using Eq. 13
Compute Y_4 using Eq. 26
X_n ← Crossover(Y_1, Y_2, Y_3, Y_4)
Yes
$n \le N$
No
Yes
$Itr \le$ max_Itr
No
Return X_{best} and $f(X_{best})$
End

Fig. 4 Flowchart of BChimp1

$$\begin{cases} X_i^d > 0.5, & \textit{if Feautre is selected} \\ \quad \textit{else}, & \textit{Fautre is not selected} \end{cases} \tag{31}$$

Where X_i^d denotes the solution i of the dimension d.

Algorithm 2 shows the steps of BChimp1 and Fig. 4 presents the associated. Each solution (selected features) is defined by a *vector* of n attributes with a binary value of 1 (selected) or 0 (not selected). First, we randomly initialize the chimps' positions ($X_i i = 1..N$) in the group of individuals. The computation of the fitness score for each chimp's position is done to obtain the four best solutions $X^d_{Attacker}$, $X^d_{Barrier}$, X^d_{Chaser}, and X^d_{Driver}. Then the initialization of the parameters f, m, a, and c are performed. During each iteration, the binary solution Y_1, Y_2, Y_3 and Y_4 of each chimpanzee are calculated by equations (18), (21), (24), and (26), respectively. For each iteration, a crossover operation between the binary solution is performed to update the chimpanzee's current solution. The calculated new fitness values is carried out using current results. Moreover, the positions of the four chimps and parameters f, m, a, and c are updated. This process is repeated until the stop criterion is met. Finally, the optimal binary solution of X_{best} is returned.

4.2 BChimp2: Second Binary Chimp Optimization Algorithm

4.2.1 Positions Updating Mechanism in BChimp2

In this section, we propose BChimp2. In this approach, the binary vector is used to update the chimps' position, as shown in equation (31):

$$X^d(t+1) = \begin{cases} 1, ifs\left(\frac{X_1^d+X_2^d+X_3^d+X_4^d}{4}\right) & \ge r_8 \\ 0, otherwise \end{cases} \tag{32}$$

Where r_8 is a random vector in the range of [0, 1], d is a dimension of search space, and $s(x)$ is sigmoid function calculated as follows:

$$s(x) = \frac{1}{1 + exp(-10(x - 0.5))} \tag{33}$$

The output of the sigmoid function in BChimp2 is interpreted as the selection attribute to the relevant class when the probability is greater than a threshold (e.g., 0.5) and with iterated computation in BChimp2 within the objective functions that leads to an optimal.

4.2.2 Algorithm of BChimp2

Algorithm 3 depicts the pseudo-code of BChimp2. First, we randomly initialize the positions of chimps in the population (either bit 1 or bit 0). Second, we evaluate the fitness of chimpanzees. The algorithm keeps the four best solutions. For each iteration, we calculate Y_1, Y_2, Y and Y_4, using equations (17), (20), (23), and (26) respectively. During each iteration, the new position of the chimpanzee is updated using equation (31). Then, we evaluate the fitness of each chimpanzee, and we update the different parameters (f, m, a, and c). This process is repeated for each chimp of the population until the desired criterion is met.

4.3 Feature Selection Using BChimp 1 and BChimp 2

The aim of our work is to come up with an intelligent selection of the most important features for achieving the best

```
Inputs :   — $n$ : size of population.
           — $It$ : current iteration.
           — $X_i$ : chimps' positions.
           — $It\_max$ : maximum number of iterations.
Outputs :  — $X_{best}$ : optimal chimpanzee binary position
           — $fitnessX_{best}$ : the best value of fitness
Begin
    Randomly initialize each chimp's positions ($X_i$  $i = 1..n$);
    Initialize parameters $f$, $m$, $a$, and $c$, it = 0;
    Evaluate fitness for each chimp;
    $X^d_{Attacker}$ ← Position of best chimp
    $X^d_{Barrier}$ ← Position of second-best chimp
    $X^d_{Chaser}$ ← Position of third-best chimp
    $X^d_{Driver}$ ← Position of fourth-best chimp
    For $it$ = 1 to $It\_max$ Do
        For $i$ = 1 to $n$ Do
            Compute $Y_1$ using Eq. 17
            Compute $Y_2$ using Eq. 20
            Compute $Y_3$ using Eq. 23
            Compute $Y_4$ using Eq. 26
            Compute $X_{i\ new}$ using Eq. 31
        End For
        Evaluate the fitness of each chimp;
        Update $X^d_{Attacker}$,$X^d_{Barrier}$, $X^d_{Chaser}$,$X^d_{Driver}$
        Update$f$, $m$, $a$, and $c$
    End For
    $X_{best} = X^d_{Attacker}$
    $fitnessX_{best} = f(X_{best})$
End
Return $X_{best}$ and $fitnessX_{best}$
```

Second Binary Chimp Optimization Algorithm

F_1	F_2	F_3	F_4	F_5	F_6	F_7	F_8	F_9	F_{10}
0	0	0	**1**	**1**	0	**1**	**1**	0	**1**

0: Future selected 1: Future not selected

Fig. 5 Solution representation example

accuracy rates and fast processing time for large dataset classification. BChimp1 and BChimp2 are applied to select a small list of attributes from a large dataset. Although, BChimp1 and BChimp2 must keep a high degree of prediction in the description of target attributes [43]. However, when the dimension of the dataset is high, the traditional exhaustive search cannot select relevant features accurately and therefore not practical. Indeed, if the dimension of the dataset is N then the dimension of search space is 2^N (N = 1000, the search space is 2^{1000}!!!). Therefore, we opt for our proposed binary chimp algorithms due to their ability to solve such hard problems.

4.3.1 Solution Representation

The search space of the proposed BChimp 1 and BChimp 2 is composed of a set of binary solutions for selecting a set of significant features. Each chimpanzee defines a solution. The solution is a vector of d elements, where d represents number of attributes in the original database. If the i^{th} feature in the original database is selected, the element i takes 1 else it takes 0 where $0 \leq i \leq d - 1$. Figure 5 illustrates the binary representation of each solution.

4.3.2 Selection Criteria

BChimp1 and BChimp2 optimize classification accuracy rates by selecting relevant features. They explore the search

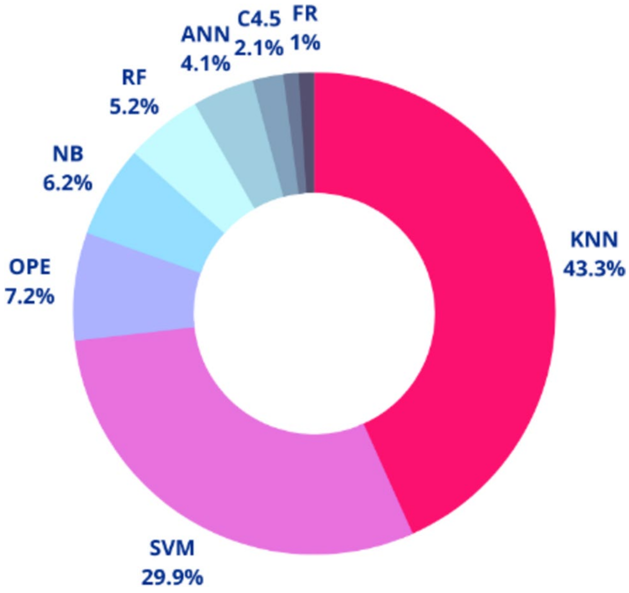


Fig. 6 Different classifiers rates in solving feature selection problems

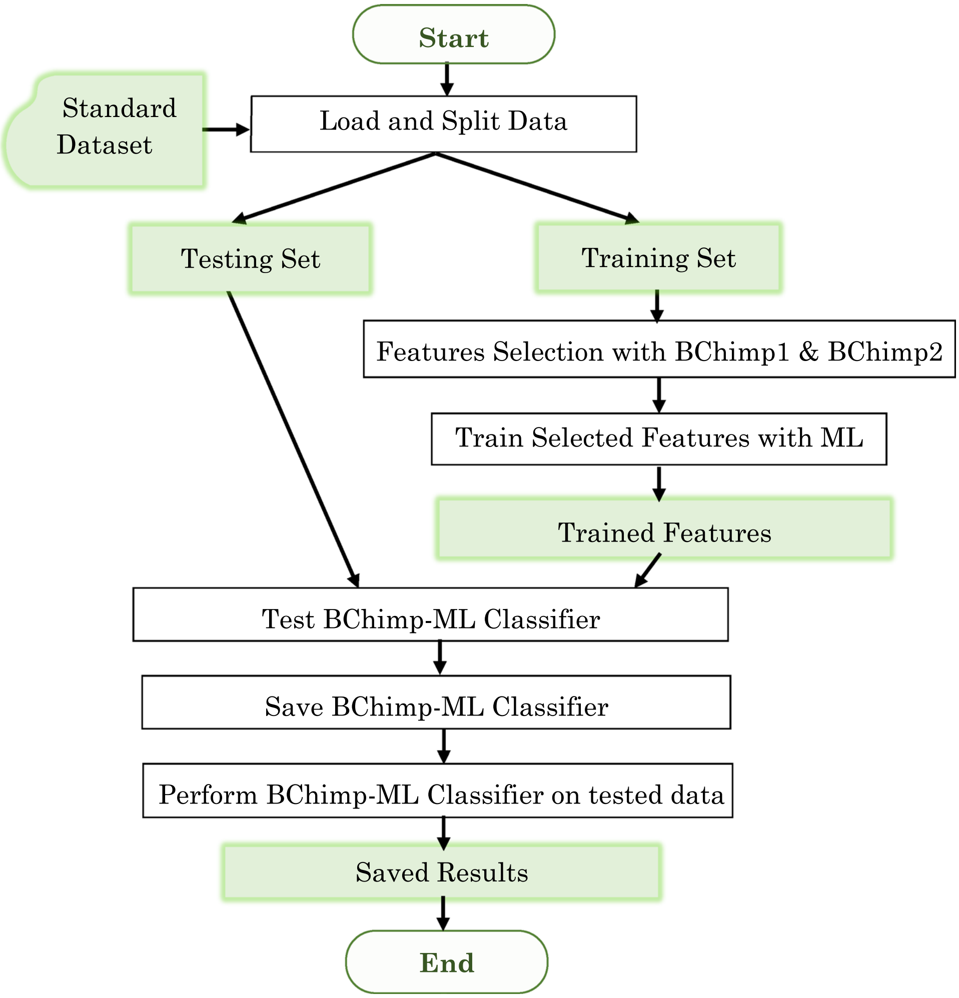
space effectively and select the ideal solution, BChimp1 selects the ideal solution from the four best possible solutions and applies a stochastic crossover on four moving solutions to deeply speed-up convergence level while BChimp2 uses the sigmoid function to select the optimal relevant features.

4.3.3 Objective Function

It plays a crucial role in selecting the best attribute with wrapper-based algorithms. In general, the choice of objective function for feature selection can be varied depending on the classification problem. The main aim is to maximize the number of relevant features or minimize the number of irrelevant features. The evaluation of BChimp 1 and BChimp 2 is done using the Classification Error Rate (CEE) metric [44] defined as fitness function (*Fitness*), which can be expressed as:

$$Fitness = CEE = \frac{Number\ of\ incorrectly\ classified\ instances}{Total\ number\ of\ instances} \tag{34}$$

Fig. 7 A general model for proposed work using binary Chimp-machine learning approach


Start
Standard Dataset
Load and Split Data
Testing Set
Training Set
Features Selection with BChimp1 & BChimp2
Train Selected Features with ML
Trained Features
Test BChimp-ML Classifier
Save BChimp-ML Classifier
Perform BChimp-ML Classifier on tested data
Saved Results
End

If *CEE* is close to 1, it will lead to the best classifier, if it is too close to 0, it will lead to a poor classifier.

4.3.4 Choice of an Efficient Classification Model

The important factor of the feature selection problem is how to choose the best classifiers that will be considered for cross-validation while maintaining high accuracy. Machine learning approaches can be used for validating feature selection algorithms, including various types of classifiers, such as KNN, SVM, NB, ANN, RF, KLM, FR, OPF, ID3, C4.5. Figure 6 illustrates the rate at which different classifiers are used in the feature selection problem.

Table 1 Experimental Parameter settings

Parameter	Value
# executions	10
# research agents	10
# maximum iterations T	100
Dimension of problem	Features size
BGA Parameters	cr= 0.8; mr = 0.2
BPSO Parameters	$c1$= 2; $c2$ = 2
BChimp Chaotic Function	Logistic function [14]

The KNN is used data classifier in the field of attribute selection despite its higher efficiency, especially for large databases compared to existing classifiers. The proposed work is based on a widely-accepted metric of KNN algorithm: Classification Error Rate (CEE) with (k = 5) [45]. The KNN algorithm used the Euclidean Distance (ED) defined in [46] as follows:

$$ED(X,Y) = \sqrt{\sum_{d=1}^{D} \left(X^d - Y^d\right)^2} \tag{35}$$

where X and Y: define the particular features and the total number of features, respectively.

- **K-Nearest Neighbor (KNN):** The key idea of KNN is to find the closest neighbors based on Euclidean distance. KNN is an easy-to-implement and easily understood-algorithm, making it suitable for basic classification tasks. However, it requires a high computation time on large datasets and suffers from the selection of the correct value of K which is crucial for its performance.
- **Decision Tree (DT):** is an intuitive model for classification and regression, it is popular and widely used in machine learning. Based on the input features, the model builds a tree. A decision tree is easy to under-

Table 2 Experimental databases.

Category	N.	Dataset name	#Instances	# Features	# Classes
General	1	Iris	150	4	3
	2	Breast EW	569	30	2
	3	Clean 1	476	168	2
Face Images	4	Yale	165	1024	15
	5	warpPIE10P	210	2420	10
	6	warpAR10P	130	2400	10
	7	orlraws10P	100	10304	10
	8	ORL	400	1024	40
	9	COIL20	1440	1024	20
Biological	10	Lung Discrete	73	325	7
	11	Colon	62	2000	2
	12	Lung	203	3312	5
	13	Lymphoma	96	4026	9
	14	GLIOMA	50	4434	4
	15	TOX 171	171	5748	4
	16	Prostate G	102	5966	2
	17	Leukemia	72	7070	2
	18	ALLAML	72	7129	2
	19	nci9	600	9712	9
	20	CLL SUB 111	111	11340	3
	21	SMK CAN 187	187	19993	2
	22	GLI 85	85	22283	2
	23	Coil	1440	1025	20

stand as it handles both numerical and categorical data and captures non-linear relationships. However, they can over fit, especially when the tree becomes too complex. Random Forest and Gradient Boosting can be efficient for systems with non-linear relationships.

- **Support Vector Machine (SVM)**: is used to separate different class data points by a hyperplane. The maximum margin between classes makes it robust to outliers and noise. The kernel trick handles non-linear relationships between features and classes. However, SVM may not scale well with large datasets, and choosing the right kernel and tuning parameters is important for optimal performance. Still, SVM remains a popular and powerful choice for many classification tasks in machine learning.
- **Random Forest (RF)**: is an efficient model that combines a group of decision trees to ensure accurate predictions. Random Forest is robust and resists overfitting problems. Random Forest is a versatile and powerful classifier that excels in treating multidimensional databases and generating accurate results. It is acceptable to use in various realm applications. It is suitable when robustness and generalization are essential.
- **Quadratic Discriminant Analysis (QDA)**: is useful when the data does not follow a linear separation pattern. It leverages probability density functions to create quadratic decision boundaries, which allows it to handle more complex datasets.
- **Multi-Layer Perceptron (MLP):** is a powerful and versatile algorithm capable of handling complex relationships between attributes and target classes. However, MLP is a is a fundamental building block of modern commonly used in various fields and deep learning models.

To assess the performance of the proposed approach, we separate the database into 70% training set and 30% testing set. Evaluation of feature selection is conducted on the training and testing sets. The cross-validation technique minimizes the overfitting rate with K = 10. The cross-validation technique consists of dividing the training set into 10 times with the same set size. Training occurs with K-1 folds, and validation occurs with the remaining folds. The evaluation process will have been repeated 10 times, each time replacing the training and validation folds. Finally, the average results of different data series will be recorded.

5 Experimental Setup and Evaluation Criteria

We begin this section with an overview of the experimental setup and parameter settings. Then, we present different performance metrics and descriptions of multidimensional databases. The flow diagram of the proposed system is shown in Fig. 7.

5.1 Experimental Setup

In this section, the experiments aim to visualize the efficiency of the proposed optimization technique compared with existing optimization algorithms based on accuracy rates and convergence levels. All techniques are developed using MATLAB 2020 with the same objective function through a PC Intel(R) Core (TM) i5-2410M CPU 2.30 GHz, 4 GB of RAM, and 64-bit operating system.

5.2 Parameter Settings

The parameters for carrying out our experiments are shown in Table 1. In the experiments, the size of population is 10. The maximum iteration T is defined as 100. The feature selection process is iterated 10 times for each data set. Finally, the average values of metrics are computed for 10 experiments. In our experiments, the optimizers of BGA are employed and the crossover rate is set to 0.8. A mutation rate is set to 0.2. In terms of discrete optimization, our experiments employ BPSO with coefficients c_1 and c_1 both equal to 2.

5.3 Databases Description

To better test and evaluate the effectiveness of our techniques, it is necessary to use large databases. Many features in large databases represent real-world scenarios. All tests are applied to 23 large databases collected from the University of Arizona [47], of 23 databases, 18 databases have more than 1000 features. A description of the datasets used in experiments is given in Table 2. All databases are high-dimensional and defined by a high number of attributes (18/23 are greater than 1000 attributes).

5.4 Performance Metrics

To verify the performance of proposed algorithms, four main metrics are measured:

1) The mean fitness (μ) and standard deviation (SD) of the fitness values for each technique after 100 runs are computed to show outperformed method. The standard deviation is defined by the following formula:

$$SD = \sqrt{\frac{1}{n}\sum_{1}^{n}(x_i - \mu)^2} \tag{36}$$

where x_i indicates the fitness of each of the candidate solutions, μ indicates the mean fitness, calculated as

Table 3 Fitness assessment by different methods

No.	Dataset name	BChimp1	BChimp2	GWO	WOA	GA	PSO
1	Iris	0.0200	**0.0100**	0.0133	0.0133	–	0.0167
2	Yale	0.3061	**0.2697**	0.3091	0.3000	0.3455	0.3242
3	warpPIE10P	0.0929	**0.0429**	0.0667	0.0714	0.0690	0.0643
4	WarpAR10P	0.4038	**0.3115**	0.4154	0.3346	0.3846	0.3731
5	Orlraws10P	0.0450	0.0450	0.0550	**0.0250**	0.0750	0.0750
6	ORL	0.1025	0.0813	0.0800	0.0950	0.0825	**0.0738**
7	COIL20	0.0122	0.0073	**0.0069**	0.0094	0.0094	0.0087
8	Lung Discrete	0.0357	**0.0286**	0.0786	0.0714	0.0571	0.0643
9	Colon	0.1167	0.0667	0.1167	**0.0583**	0.0917	0.1167
10	Lung	0.0175	0.0225	0.0225	0.0175	**0.0075**	0.0175
11	Lymphoma	0.1211	0.1211	0.1263	0.1316	0.1158	**0.1105**
12	GLIOMA	0.1800	0.1000	0.1700	**0.0800**	**0.0800**	0.1500
13	TOX 171	0.2000	**0.1000**	0.1941	0.1500	0.1294	0.1588
14	Prostate_G	0.1300	0.0950	0.1050	**0.0600**	0.1250	0.0850
15	Leukemia	0.0357	**0.0286**	0.0714	0.0357	0.0357	0.0643
16	ALLAML	0.1357	**0.0571**	0.0929	0.0571	0.1143	0.1286
17	nci9	0.5417	**0.4667**	0.5417	0.4917	0.5083	0.5083
18	CLL_SUB_111	0.2545	**0.1682**	0.3090	0.2682	0.2227	0.2818
19	SMK CAN 187	0.2703	**0.1946**	0.2487	0.2351	0.2730	0.2757
20	GLI 85	0.0353	0.0353	0.0588	**0.0235**	0.0647	0.0353
21	Breast_EW	0.0150	0.0168	0.0257	0.0301	0.0150	**0.0115**
22	Clean 1	0.0716	**0.0463**	0.0558	0.0884	0.0484	0.0484
23	COIL	**0.0073**	**0.0073**	0.0122	0.0104	0.0083	0.0125

Bold represents the best results obtained through optimization method, for each database

Table 4 The standard deviation assessment by different methods

No.	Dataset name	BChimp1	BChimp2	GWO	WOA	GA	PSO
1	Iris	0.0000	0.0000	0.0000	0.0000	–	0.0115
2	Yale	0.0127	0.0176	0.0057	0.0066	0.0136	0.0025
3	warpPIE10P	0.0019	0.0034	0.0025	0.0019	0.0024	0.0151
4	WarpAR10P	0.0119	0.0229	0.0070	0.0241	0.0134	**0.0018**
5	Orlraws10P	0.0012	0.0025	**0.0000**	0.0140	0.0019	0.0066
6	ORL	0.0038	0.0098	0.0064	0.0030	0.0069	**0.0021**
7	COIL20	0.0015	0.0022	0.0013	**0.0011**	0.0016	0.0040
8	Lung discrete	0.0115	0.0090	**0.0046**	0.0143	0.0087	0.0138
9	Colon	0.0108	0.0128	0.0146	0.0232	0.0110	**0.0016**
10	Lung	0.0009	0.0010	0.0029	0.0015	0.0011	0.0031
11	Lymphoma	0.0000	0.0018	0.0050	0.0026	0.0019	0.0039
12	GLIOMA	0.0049	**0.0000**	0.0020	0.0134	0.0086	0.0187
13	TOX_171	0.0100	0.0222	0.0102	0.0140	0.0259	**0.0027**
14	Prostate_GE	0.0026	0.0022	**0.0020**	0.0061	0.0060	0.0072
15	Leukemia	0.0058	0.0061	**0.0047**	0.0090	0.0079	0.0063
16	ALLAML	0.0036	0.0078	0.0045	0.0111	0.0147	0.0094
17	nci9	0.0118	0.0117	**0.0051**	0.0283	0.0137	0.0148
18	CLL_SUB_111	0.0142	0.0249	0.0156	0.0244	0.0188	**0.0067**
19	SMK CAN 187	0.0078	0.0121	**0.0050**	0.0174	0.0086	0.0150
20	GLI 85	0.0088	0.0085	0.0042	0.0273	0.0124	**0.0039**
21	Breast_EW	0.0024	0.0070	**0.0016**	0.0019	0.0036	0.0087
22	Clean1	0.0054	0.0125	0.0066	0.0043	0.0099	**0.0022**
23	COIL	0.0010	0.0021	0.0012	**0.0006**	0.0015	0.0115

Bold represents the best results obtained through optimization method, for each database

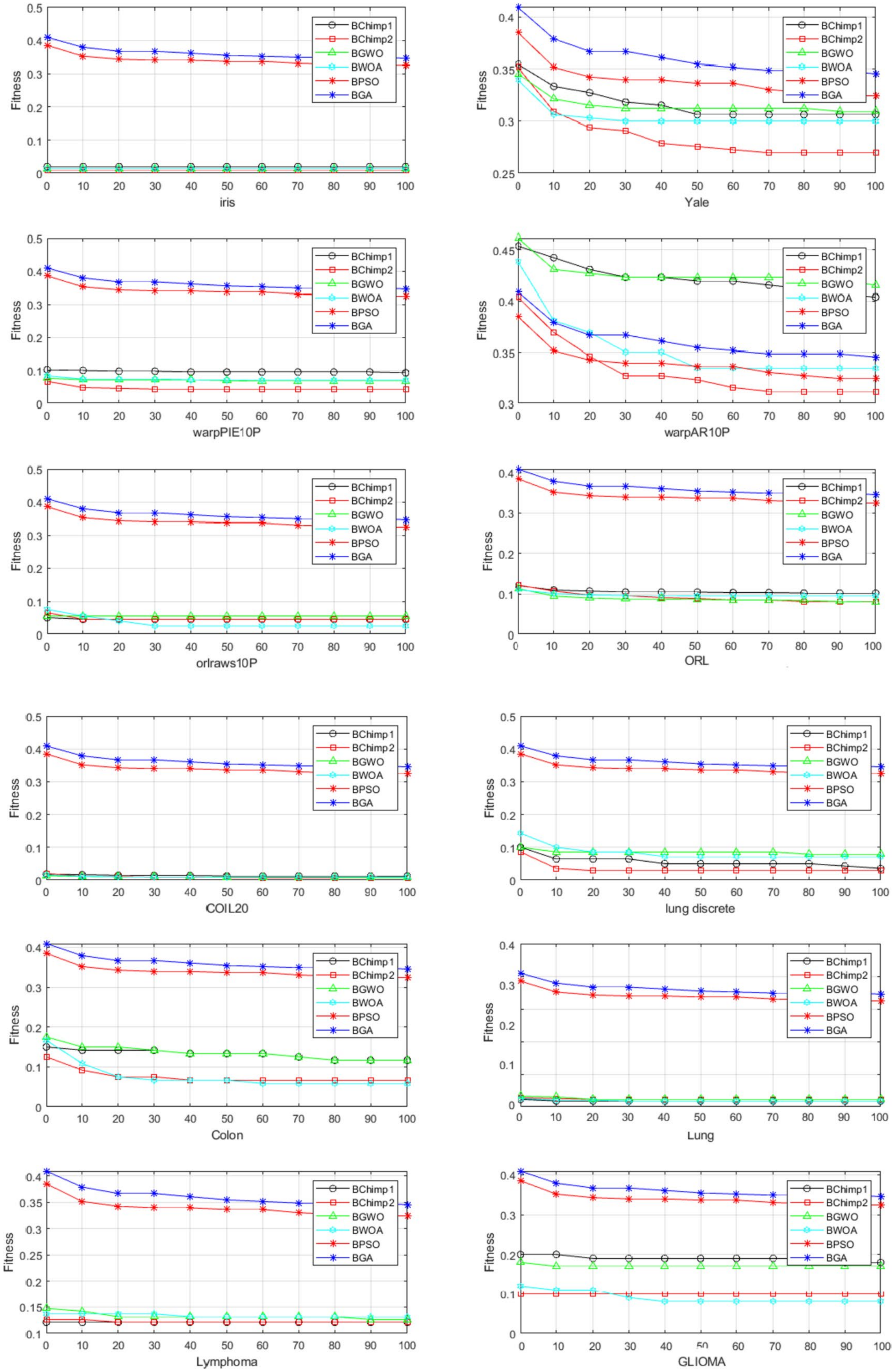


Fig. 8 Comparison of convergence of the proposed techniques with other techniques

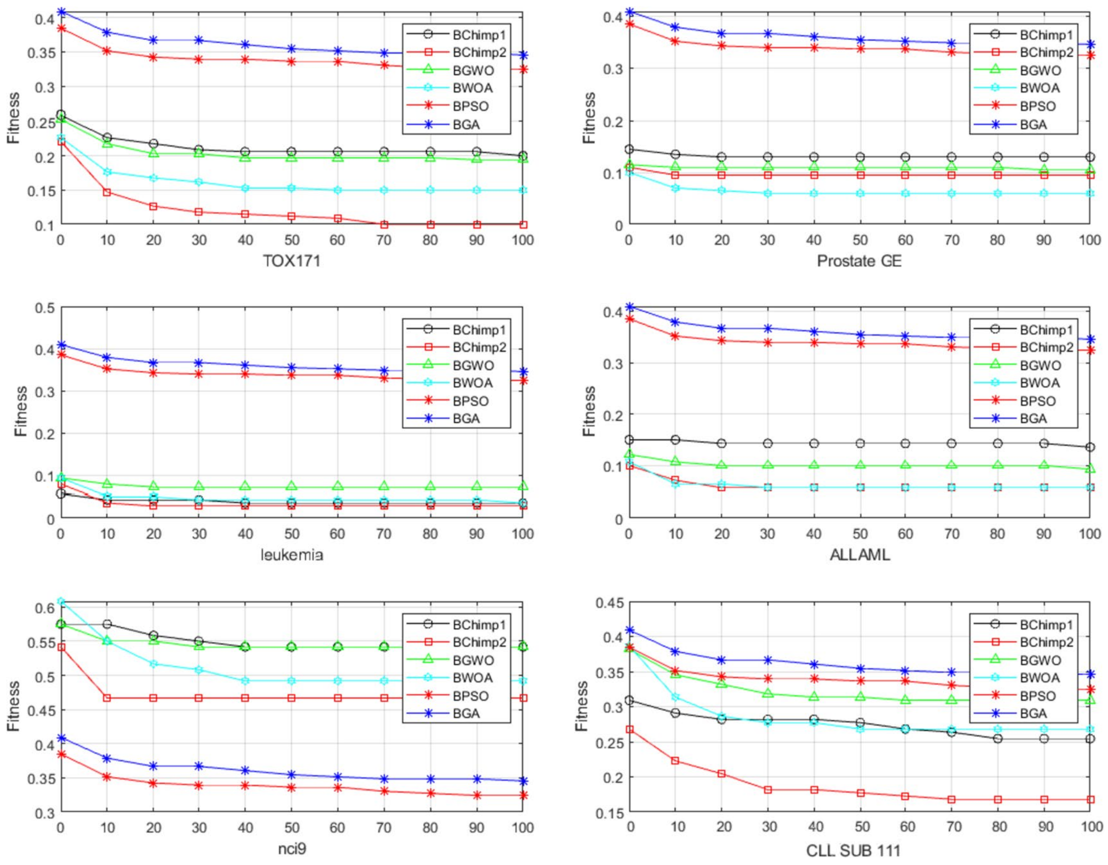


Fig. 8 (continued)

$\mu = \Sigma(xi)/n$ and n indicates the number of the candidate solutions.

2) The dataset size difference before and after the attribute selection process is computed to show how much features are reduced. The dataset difference is defined as follows:

$$D\_size = F\_size - O\_size \tag{37}$$

where F_size and O_size represent dataset size before and after the feature selection respectively.

3) The average accuracy rate is computed after 10 runs to show the outperformed method. For quality evaluation, our experiments employ accuracy (Ac), precision (Pre), recall, specificity, and F-score as the metrics to assess the level of security achieved, as shown in Eq. (37).

$$Ac = \frac{TP + TN}{TP + TN + FP + FN} \tag{38}$$

$$Pre = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

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

$$\text{F\_score} = 2 * \frac{Prec * Recall}{Prec + Recall}$$

where

- TP is True Positive,
- TN is True Negative,
- FP is False Positive,
- FN is False Negative.

The average accuracy value is defined as follows:

$$\overline{Ac} = \frac{\sum_{i=1}^{n} Ac_i}{n} \tag{39}$$

where Ac_i represents the accuracy value for each run.

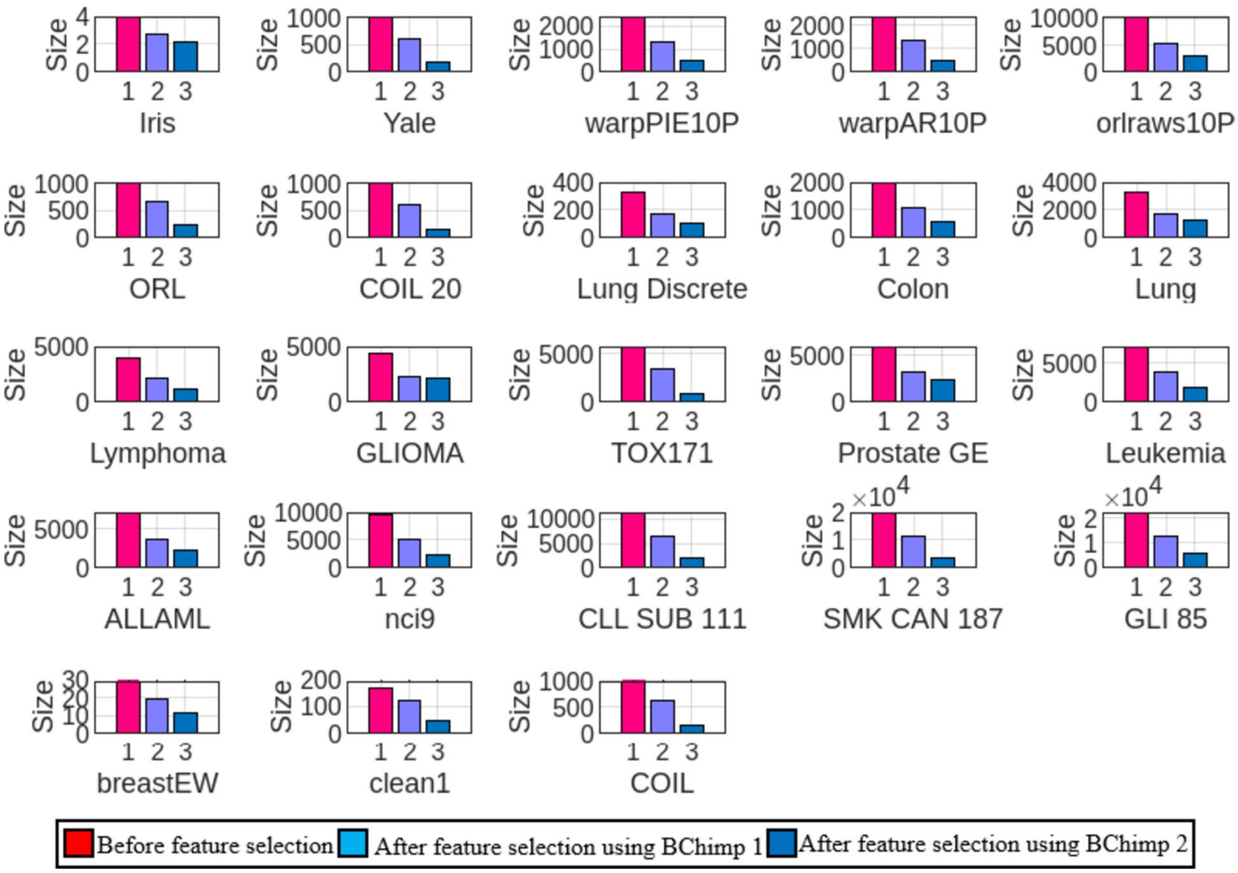


Fig. 9 Dimensionality reduction before and after features selection

4) The convergence rate of each method is defined to study the convergence speed of each method. The convergence rate is defined as follows:

$$Cr = It \to f^* \tag{40}$$

where It represents the number of steps required for achieving the optimal attributes.

6 Performance Comparison

6.1 Comparison of the Average Fitness Values

To demonstrate the efficiency of the proposed algorithms, the average fitness values of BChimp1 and BChimp2 were compared with PSO [28], GWO [13], WOA [48], and GA [9] after 10 executions using 23 multidimensional databases.

Table 3 presents the comparison results, which reveal that all the mean fitness values of BChimp1 and BChimp2 are very good and achieved the exact minimum. From the mean fitness values, we can confirm that BChimp 2 is more efficient.

6.2 Comparison of the Standard Deviation of Fitness Values

The standard deviation of fitness values between the proposed selection techniques and PSO [28], GWO [13], WOA [48], and GA [9] is compared in Table 4. This further confirms that the fitness values of different selection algorithms are very close to each other. However, the GWO algorithm outperforms other algorithms in several cases.

6.3 Comparison of Convergence of Features Selection Algorithms

Through this section, a comparison of the proposed BChimp1 and BChimp2 with relevant optimization techniques has been conducted in terms of convergence,

Table 5 Dimensionality feature reduction before and after selection

No.	Dataset name	Before selection	BChimp1	BChimp2
1	Iris	4	2.70	**2.20**
2	Yale	30	19	**11.60**
3	warpPIE10P	168	123.60	**45.80**
4	WarpAR10P	1024	605.90	**190.80**
5	Orlraws10P	2420	1304.80	**527.70**
6	ORL	2400	1349.80	**484.60**
7	COIL20	10304	5279.10	**2916.40**
8	Lung discrete	1024	658.60	**233.60**
9	Colon	1024	601.50	**144.20**
10	Lung	325	171.70	**100.70**
11	Lymphoma	2000	1077.60	**556.80**
12	GLIOMA	3312	1755.20	**1244.30**
13	TOX_171	4026	2125.80	**1114**
14	Prostate_GE	4434	2231.90	**2088.10**
15	Leukemia	5748	3481.70	**952.90**
16	ALLAML	5966	3186.80	**2454.60**
17	nci9	7070	3769	**1820.10**
18	CLL_SUB_111	7129	3583.60	**2192.70**
19	SMK CAN 187	9712	5266	**2239.90**
20	GLI 85	11340	6582.40	**1983.30**
21	Breast_EW	19993	11338.80	**3480.20**
22	Clean1	22283	12883.90	**5808.80**
23	COIL	1025	625.80	**157.70**

Bold represents the best results obtained through optimization method, for each database

represented by the level of convergence in solving high-dimensional feature optimizations. All feature selection algorithms are evaluated for 10 runs. The graph of their convergences is shown in Fig. 8. The obtained convergence behavior and speed of BChimp1 and BChimp2 are very encouraging compared to other techniques presented in [13, 28, 48], and [9]. However, BChimp1 and BChimp2 can converge quickly and deeply to find the exact minimum compared to PSO [28], GWO [13], WOA [48], and GA [9]. Because the two proposed algorithms have succeeded in combining the good quality of exploration and exploitation of the chimpanzee hunting process on the one hand and the other hand, the two algorithms provide an optimal binary solution. From the convergence levels, we can confirm that BChimp 2 offers a high convergence compared to BChimp1 and other algorithms for all experimented datasets.

6.4 Comparison of Dimensionality Reduction Before and After Feature Selection

This section presents the experimental results of dimensionality reduction before and after selection using the two proposed techniques (BChimp 1 and BChimp 2) applied to 23 datasets. Features are reduced during the selection process to a set of the most relevant features. Figure 9 and Table 5 illustrate the comparison of dimensionality reduction before and after binary feature selection on 23 datasets. It is noted that the proposed optimization techniques (BChimp 1 and BChimp 2) offer good dimensionality reduction as shown in all datasets of Table 5. The proposed technique BChimp 2 has an average reduction rate of 72.54% and almost the lowest dimensionality reduction rate among the 23 datasets used and presented particularly a reduction rate of 0.0037 with the COIL dataset. The dimensionality reduction rate for BChimp1 is 42.77%, and the lowest dimensionality reduction rate can be seen with GLIOMA of 0.01.

From the dimensionality reduction values, we can conclude that BChimp 2 is more effective while achieving the lowest dimensionality reduction.

6.4.1 Accuracy Comparison of KNN Classifier With/Without Binary Chimp Methods

Figure 10 depicts the comparison of accuracy values for KNN classifier with/without feature selection using BChimp1 and BChimp2 on 23 datasets. From Fig. 10, we observed that the proposed BChimp2 algorithm with the KNN classifier has higher accuracy values than BChimp1 model with KNN classifier. We also observed that KNN classifier with significant relevant feature selection by the proposed binary chimpanzee methods improved classification accuracy.

6.4.2 Accuracy Comparison of DT Classifier With/Without Binary Chimp Methods

To assess the quality of solutions, the DT machine learning model is combined with the proposed algorithms (BChimp1 and BChimp2) on 23 datasets already described in Section 6.1 and compared to the original DT machine learning model. The accuracy comparison of DT classifier with and without binary chimpanzee methods (BChimp1 and BChimp2) is depicted in Fig. 11. The accuracy scores are very close to each other before and after the selection process by BChimp 1 and BChimp 2. However, BChimp1 outperformed the DT classifier in nine cases, while BChimp2 outperformed the DT classifier in seven cases.

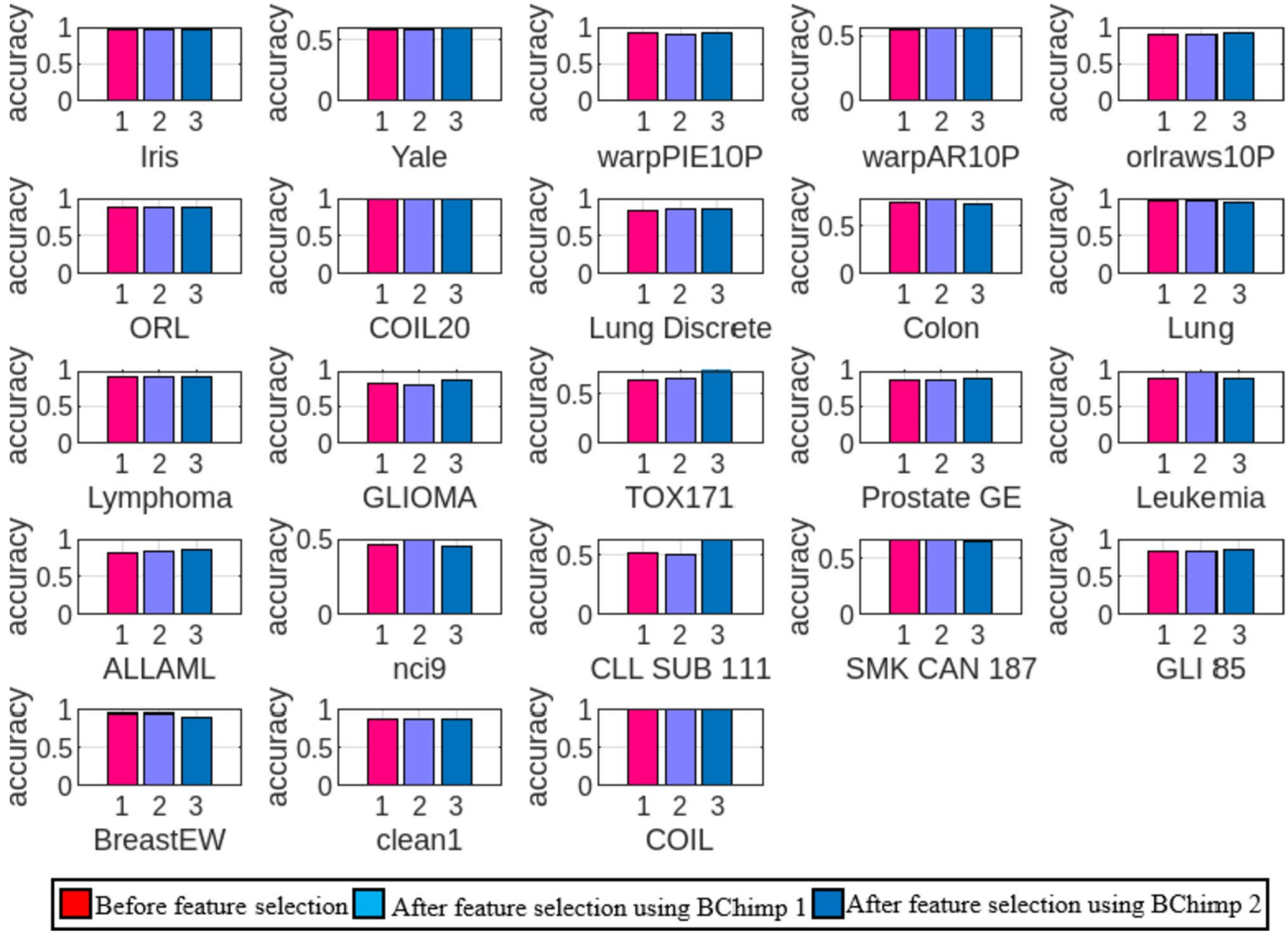


Fig. 10 Accuracy comparison of KNN classifier with/without Binary chimpanzee methods

6.4.3 Accuracy Comparison of SVM Classifier With/Without Binary Chimp Methods

Furthermore, when compared the SVM classifier to BChimp1-SVM and BChimp2-SVM as shown in Fig. 12, the recognition rates are very close to each other before and after the selection process by BChimp1 and BChimp2, BChimp1 outperformed it in six cases and BChimp2 outperformed it in ten cases.

6.4.4 Accuracy Comparison of FR Classifier With/Without Binary Chimp Methods

The comparison of the FR classifier to BChimp1-FR and BChimp2-FR is shown in Fig. 13. The accuracy values are very close to each other before and after the selection process by BChimp1 and BChimp2, BChimp1 outperformed it in eight cases, and BChimp2 outperformed it in six cases.

6.4.5 Accuracy Comparison of QDA Classifier With/Without Binary Chimp Methods

The comparison of the QDA classifier to BChimp1-QDA and BChimp2-QDA is showed in Fig. 14. The recognition rates are very close to each other before and after the selection process by BChimp1 and BChimp2, while BChimp 1 outperformed it in 4 cases and BChimp2 outperformed it in 7 cases.

6.4.6 Accuracy Comparison of MLP Classifier With/Without Binary Chimp Methods

Figure 15 depicts the accuracy comparison of MLP classifier with/without proposed binary Chimp methods (BChimp1 and BChimp2). The recognition rates are very close to each other before and after the selection process

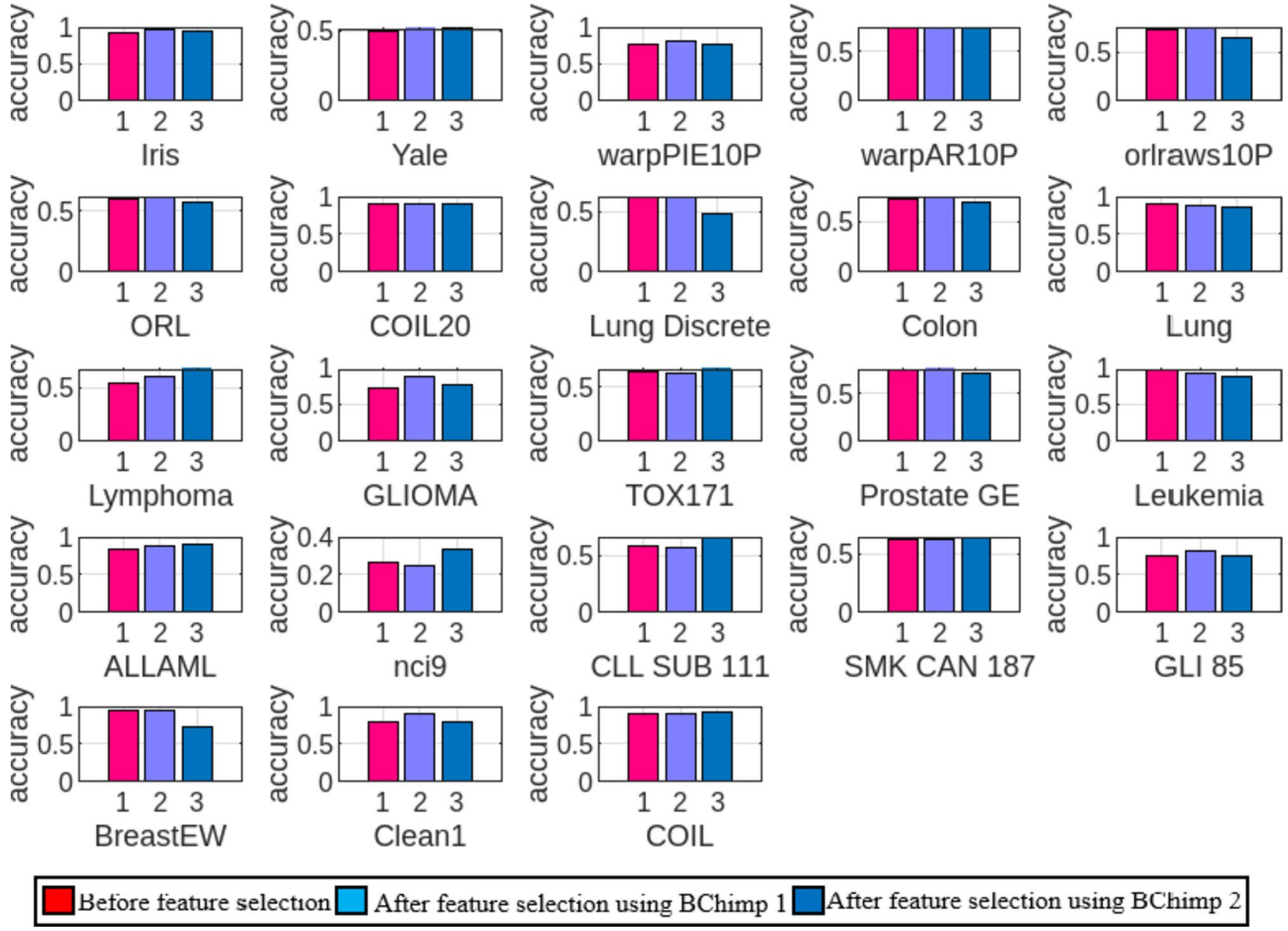


Fig. 11 Accuracy comparison of DT classifier with/without binary chimpanzee methods

by BChimp1 and BChimp2, while BChimp1 outperformed it in 7 cases and BChimp2 outperformed it in 3 cases.

6.4.7 Accuracy Comparison of Six Classifiers With/Without Proposed Binary Chimp Methods

Table 6 shows the accuracy comparison of six classifiers KNN, DT, SVM, RF, DA, and MLP with/without proposed binary chimp techniques over 23 datasets. The accuracy of the proposed BChimp2 is 82.02 % and 83.83% for BChimp1 for all six classifiers over all 23 datasets used in experiences.

6.4.8 Comparision of Classification Results Across Different Datasets

To evaluate the BChimp 2 using different classifiers (KNN, DT, SVM, and RF), the accuracy, precision, recall, and F-score are considered as the average of 10 runs across 23 datasets. According to the classification results in Figs. 16, 17, 18, and 19, the proposed BChimp2 SVM demonstrates the best results compared to BChimp2–KNN, BChimp2–DT, and BChimp2–RF for all datasets. This indicates that the use of BChimp2 algorithm enhances the efficiency of the SVM classifier by increasing precision and F-score as the selection is performed within the most significant features. This outcome highlights the efficiency gained by introducing BChimp2 as the features selection step before data classification and the superiority of SVM compared to KNN, DT, and KNN, which grantees high classification results in terms of accuracy, precision, recall, and F-score.

6.4.9 Discussion and Lessons Learned

Our contribution highlights the importance of selecting optimal and significant features before the classification phase as a lesson learned from the above sections. The selection should be based on optimization methods that come up with high classification accuracy, high convergence level, and

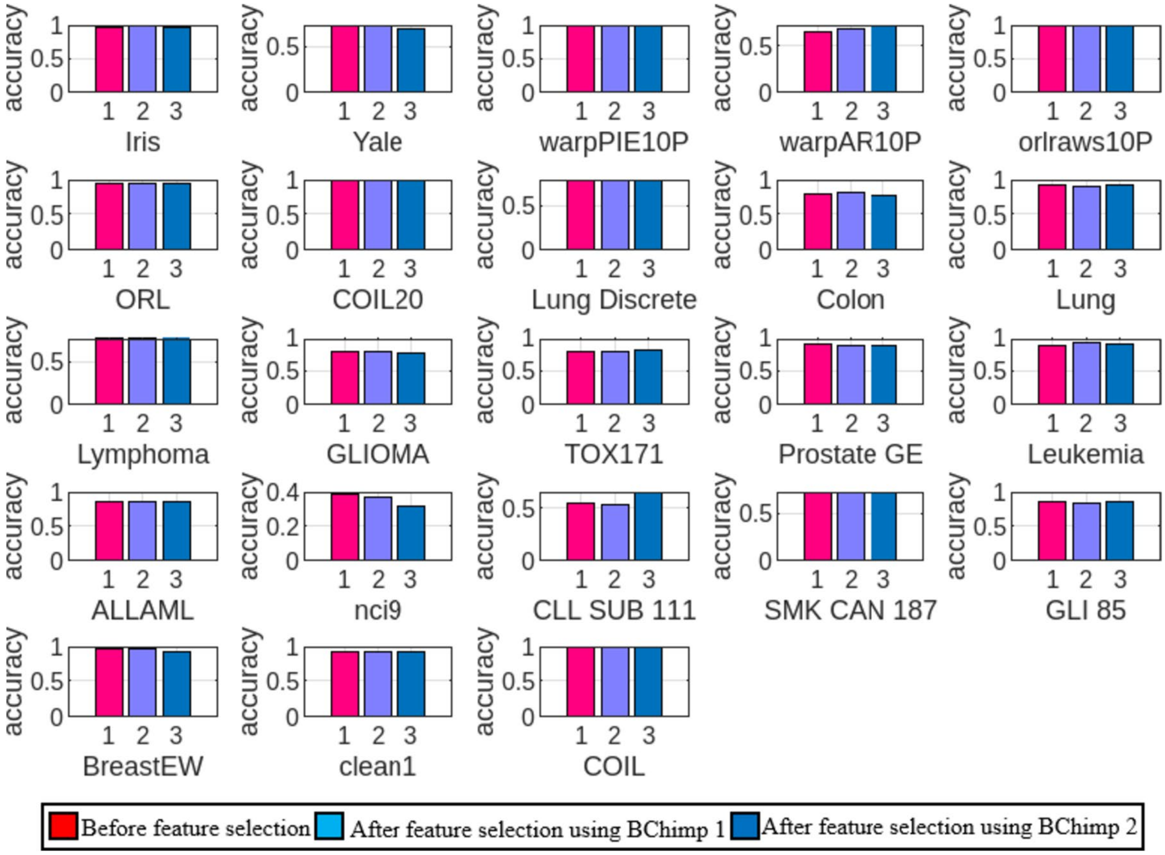


Fig. 12 Accuracy comparison of SVM classifier with/without binary chimpanzee methods

acceptable execution time which may be important criteria in high dimensional data.

As demonstrated in Figs. 10–15, we can conclude that the classification accuracy of six learning machines before and after feature selection using the two proposed algorithms BChimp 1 and BChimp 2 are very close to each other. This indicates that the two proposed algorithms BChimp 1 and BChimp 2 were capable to select the important attributes in all experimented databases. Note that the selection time of the proposed BChimp1 and BChimp1 are acceptable with an average of 10 times and 4 times respectively more than normal datasets before selection which is effective and which could be practicable. In addition, note that the accuracy is high in all used datasets for all six learning machines, which means that the tested machine learning classifiers with BChimp 1 or BChimp 2 are worthy to be practical with various domain applications. The proposed BChimp1 with ML (respectively BChimp2 with ML) method has outperformed the capability of classifiers without feature selection in achieving high accuracy on 6 cases (respectively 4 cases) in all 23 datasets. The comparison study highlighted the improved classification average accuracy with 82.02 % for BChimp2-ML and 83.83% for BChimp1-ML and reduced data size with 72.54% for BChimp2-ML and 42.77% for BChimp1-ML for all dataset, which proves the ability and the effectiveness of the suggested systems in the area of feature selection and categorization. Finally, from the detailed results and discussion, it is assumed that the QDA with 90.18% classifier model has ranked first by effectual performance over other classifiers on 23 datasets followed by the MLP with 89.98% classifier and RF with 81.19 % in the third place. However, the proposed methods suffer from some shortcomings such as:

- The convergence levels are not perfect in some databases for MLP and QDA because we noted when designing these classifiers on the MATLAB platform, some errors appeared since these classifiers have high computational complexity influence outstandingly on relevant features selectivity and data of different natures in some databases as depicted in Table 6.

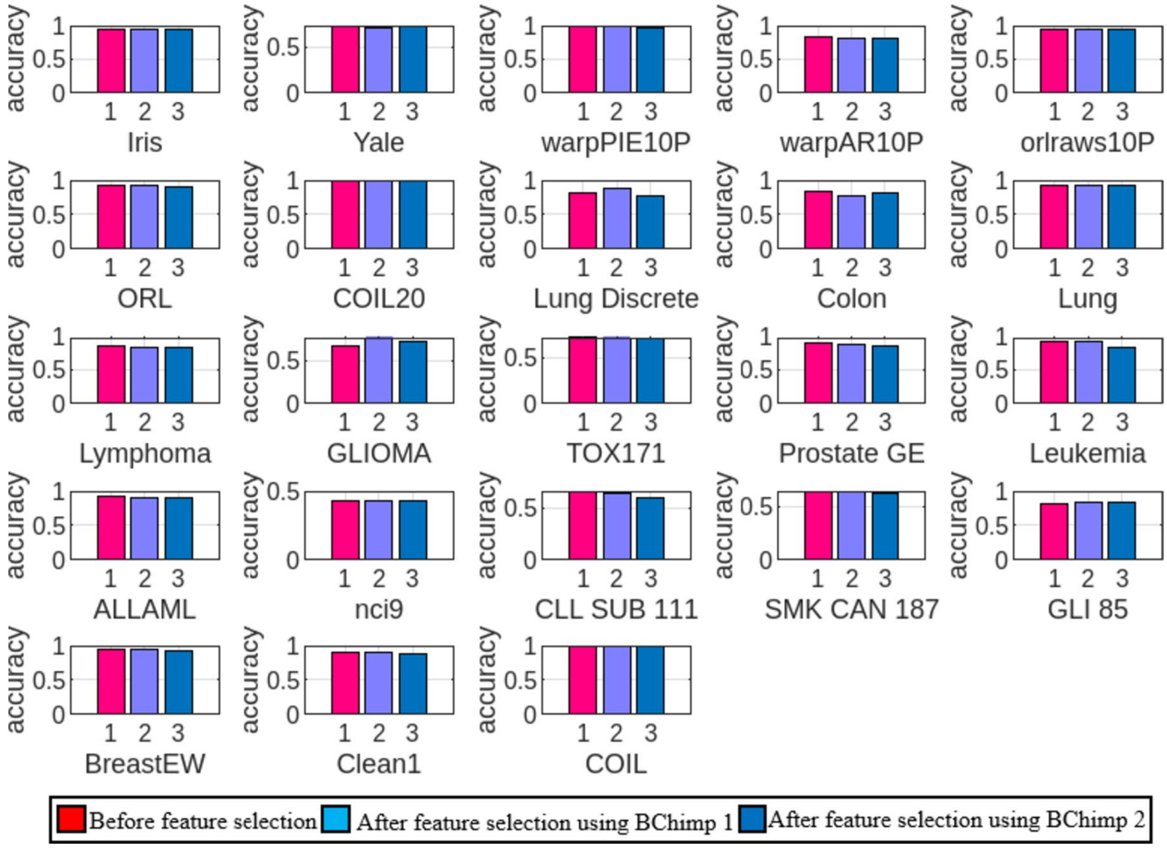


Fig. 13 Accuracy comparison of FR classifier with/without binary chimpanzee methods

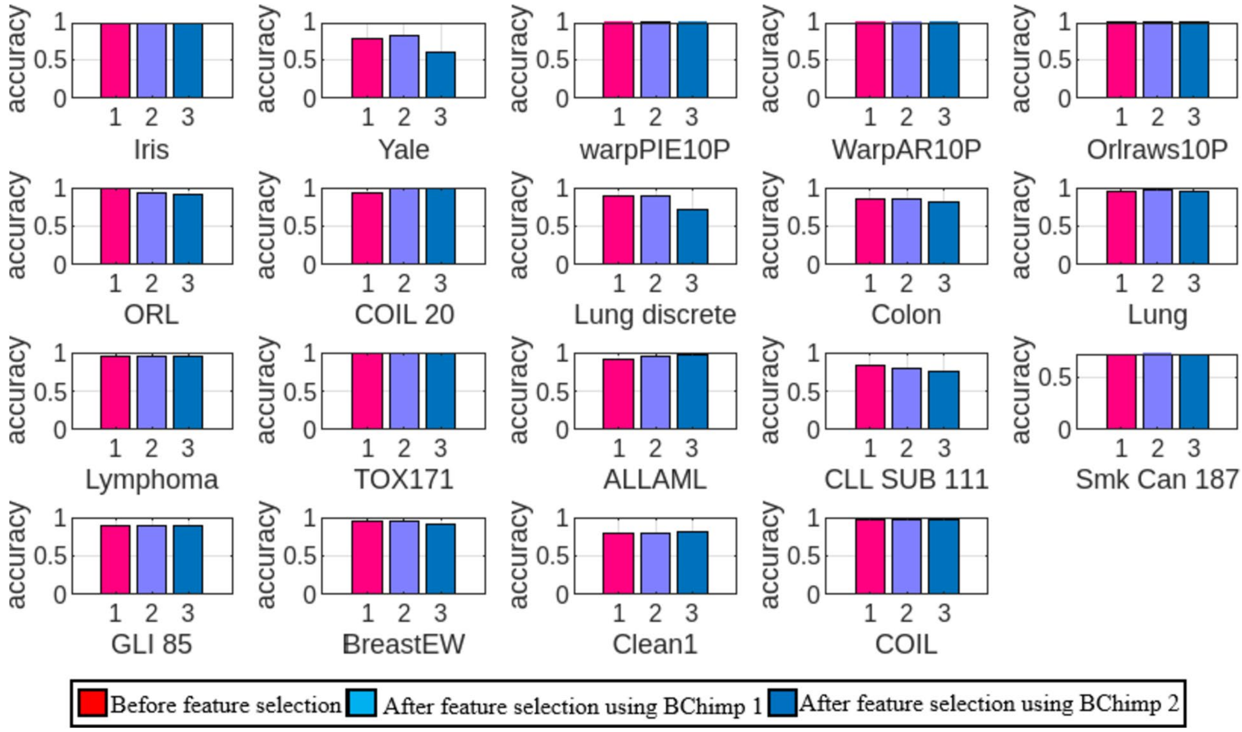


Fig. 14 Accuracy comparison of QDA classifier with/without binary chimpanzee methods

Table 6 Accuracy values before and after features selection and machine learning methods

Dataset name		Machine learning methods					
		KNN	DT	SVM	RF	QDA	MLP
Iris	Before Selection	96.00	92.67	96.00	94.00	**98.00**	**98.00**
	BChimp1	96.00	96.00	98.00	95.33	97.33	97.33
	BChimp2	96.00	94.00	96.67	94.67	**98.00**	**98.00**
Yale	Before Selection	58.05	50.33	72.83	72.87	78.27	66.58
	BChimp1	58.16	52.68	71.62	70.96	81.91	87.21
	BChimp2	60.74	52.32	68.38	72.79	60.70	**90.96**
warpPIE10P	Before Selection	91.43	77.14	99.05	98.10	**100.00**	90.48
	BChimp1	90.00	81.43	99.05	98.10	99.52	**100.00**
	BChimp2	92.38	76.67	100	96.19	**100.00**	**100.00**
WarpAR10P	Before Selection	54.62	74.62	64.62	83.08	**100.00**	96.15
	BChimp1	56.92	73.85	67.69	80.00	**100.00**	94.61
	BChimp2	56.38	74.62	71.54	81.54	**100.00**	44.62
Orlraws10P	Before Selection	56.38	74.62	71.54	81.54	**100.00**	44.62
	BChimp1	90.00	73.00	99.00	95.00	99.00	99.00
	BChimp2	91.00	76.00	99.00	94.00	99.00	**100.0**
ORL	Before Selection	88.25	60.00	93.75	92.00	98.50	89.75
	BChimp1	87.00	61.75	94.25	92.25	92.25	90.75
	BChimp2	88.25	57.50	**94.50**	90.50	90.50	79.00
COIL20	Before Selection	98.33	90.14	100	99.58	93.96	98.54
	BChimp1	98.13	90.56	99.93	99.44	99.44	96.39
	BChimp2	98.68	89.79	**99.86**	99.44	99.46	94.38
Lung Discrete	Before Selection	83.99	62.14	79.46	81.96	89.11	93.33
	BChimp1	86.43	62.86	79.46	89.11	89.11	**97.32**
	BChimp2	84.82	48.39	79.82	78.04	72.50	96.07
Colon	Before Selection	75.24	71.19	78.81	82.83	**85.71**	–
	BChimp1	79.29	74.29	80.71	77.38	85.44	–
	BChimp2	73.33	69.29	78.10	81.90	82.33	–
Lung	Before Selection	96.07	90.19	92.17	92.12	96.02	–
	BChimp1	96.07	88.19	91.17	92.64	**97.52**	–
	BChimp2	95.60	85.74	92.12	92.71	96.05	–
Lymphoma	Before Selection	90.78	54.89	78.11	84.56	**95.89**	–
	BChimp1	89.78	60.78	78.00	83.44	**95.89**	–
	BChimp2	90.67	68.89	78.22	83.44	**95.89**	–
GLIOMA	Before Selection	82.00	72.00	80.00	66.00	–	**90.00**
	BChimp1	80.00	88.00	80.00	78.00	–	82.00
	BChimp2	86.00	76.00	76.00	72.00	–	68.00
TOX 171	Before Selection	64.31	63.10	79.58	72.45	98.30	**98.86**
	BChimp1	65.42	62.52	78.89	73.63	98.82	95.33
	BChimp2	74.35	67.29	81.34	70.78	98.24	98.27
Prostate_GE	Before Selection	86.09	75.45	89.09	89.18	–	**99.00**
	BChimp1	86.36	77.18	87.09	88.18	–	97.00
	BChimp2	88.18	71.82	88.00	84.55	–	96.09
Leukemia	Before Selection	88.75	95.71	87.68	93.21	–	–
	BChimp1	**96.61**	91.96	93.21	93.21	–	–
	BChimp2	87.68	87.50	90.36	84.29	–	–
ALLAML	Before Selection	82.14	83.39	86.07	91.79	91.14	**100**
	BChimp1	83.39	87.50	86.07	90.54	94.46	98.75
	BChimp2	85.00	89.46	86.07	90.18	97.32	95.89

Table 6 (continued)

Dataset name		Machine learning methods					
		KNN	DT	SVM	RF	QDA	MLP
nci9	Before Selection	**46.67**	26.67	38.33	43.33	–	–
	BChimp1	50.00	25.00	36.67	43.33	–	–
	BChimp2	45.00	33.33	31.67	43.33	–	–
CLL_SUB_111	Before Selection	51.36	58.64	54.92	66.59	83.71	92.73
	BChimp1	50.68	56.74	53.18	64.85	79.24	82.27
	BChimp2	63.71	66.52	65.68	59.62	74.70	**97.27**
SMK CAN 187	Before Selection	67.37	62.46	72.31	64.09	70.76	**88.22**
	BChimp1	66.49	61.49	71.55	64.68	73.19	87.57
	BChimp2	64.68	64.12	72.28	61.99	71.23	88.10
GLI 85	Before Selection	83.89	74.03	86.11	82.08	89.58	96.36
	BChimp1	83.89	82.22	84.31	83.89	89.58	**97.64**
	BChimp2	84.86	75.28	84.86	84.72	89.31	96.53
Breast1_EW	Before Selection	95.08	94.20	96.31	94.90	95.07	**97.54**
	BChimp1	95.25	94.02	96.31	95.08	94.54	97.18
	BChimp2	91.03	72.79	92.26	93.33	90.51	94.02
Clean 1	Before Selection	87.60	78.79	**92.86**	90.32	79.02	–
	BChimp1	89.09	89.49	92.86	89.49	78.78	–
	BChimp2	86.99	79.61	92.44	89.10	80.86	–
COIL	Before Selection	97.85	90.07	100.0	99.65	96.26	97.01
	BChimp1	97.99	90.21	99.93	99.86	96.32	98.06
	BChimp2	98.89	91.60	**99.79**	99.58	96.74	66.46

Bold represents the best results obtained through optimization method, for each database

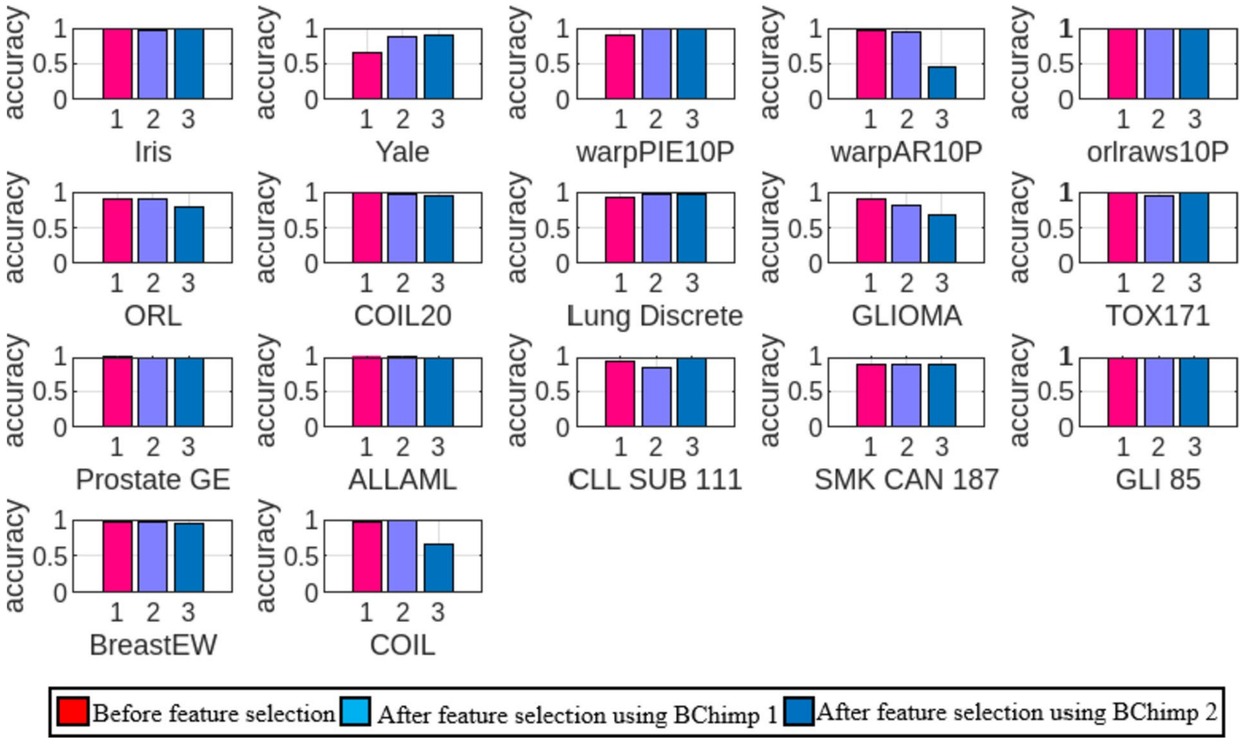


Fig. 15 Accuracy comparison of MLP classifier with/without Binary chimpanzee methods

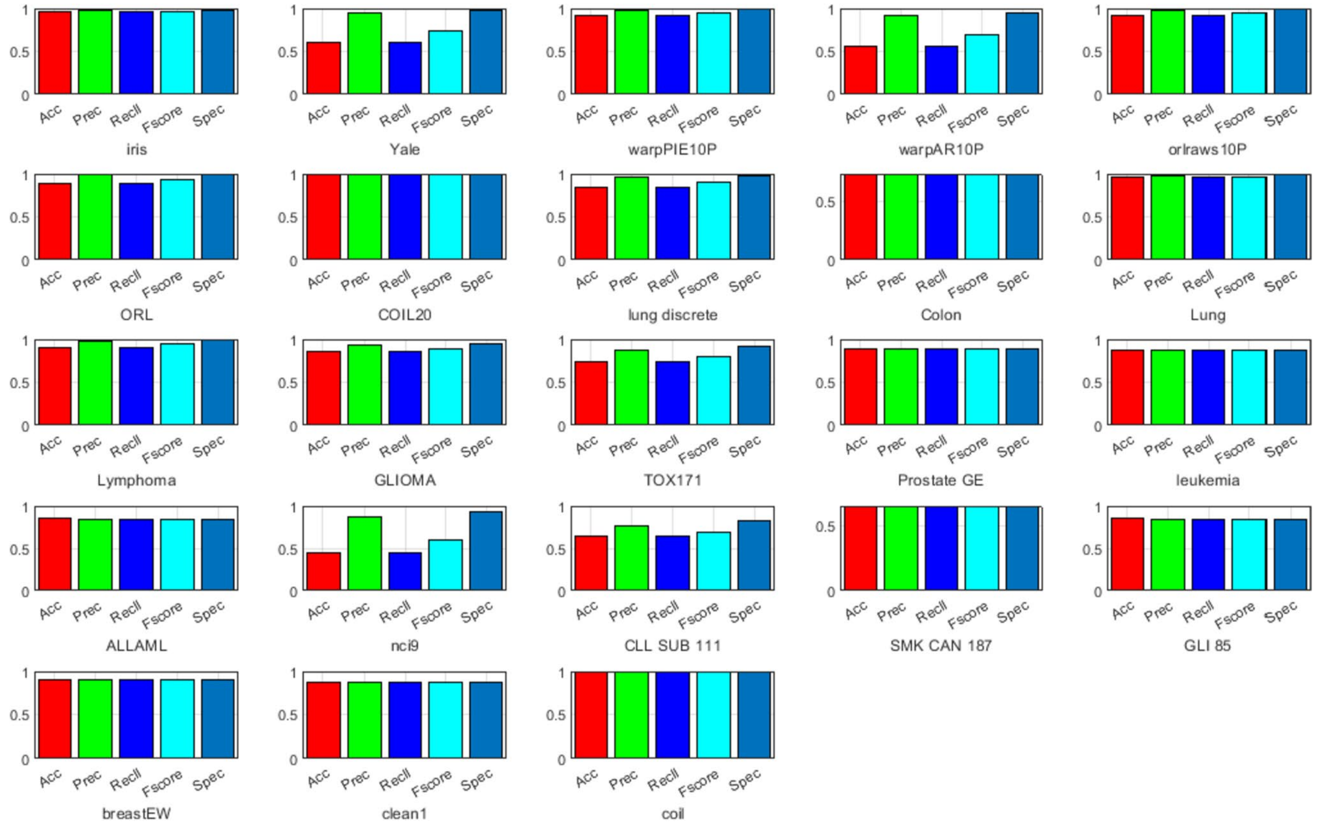


Fig. 16 Average classification results of KNN classifier for 23 datasets

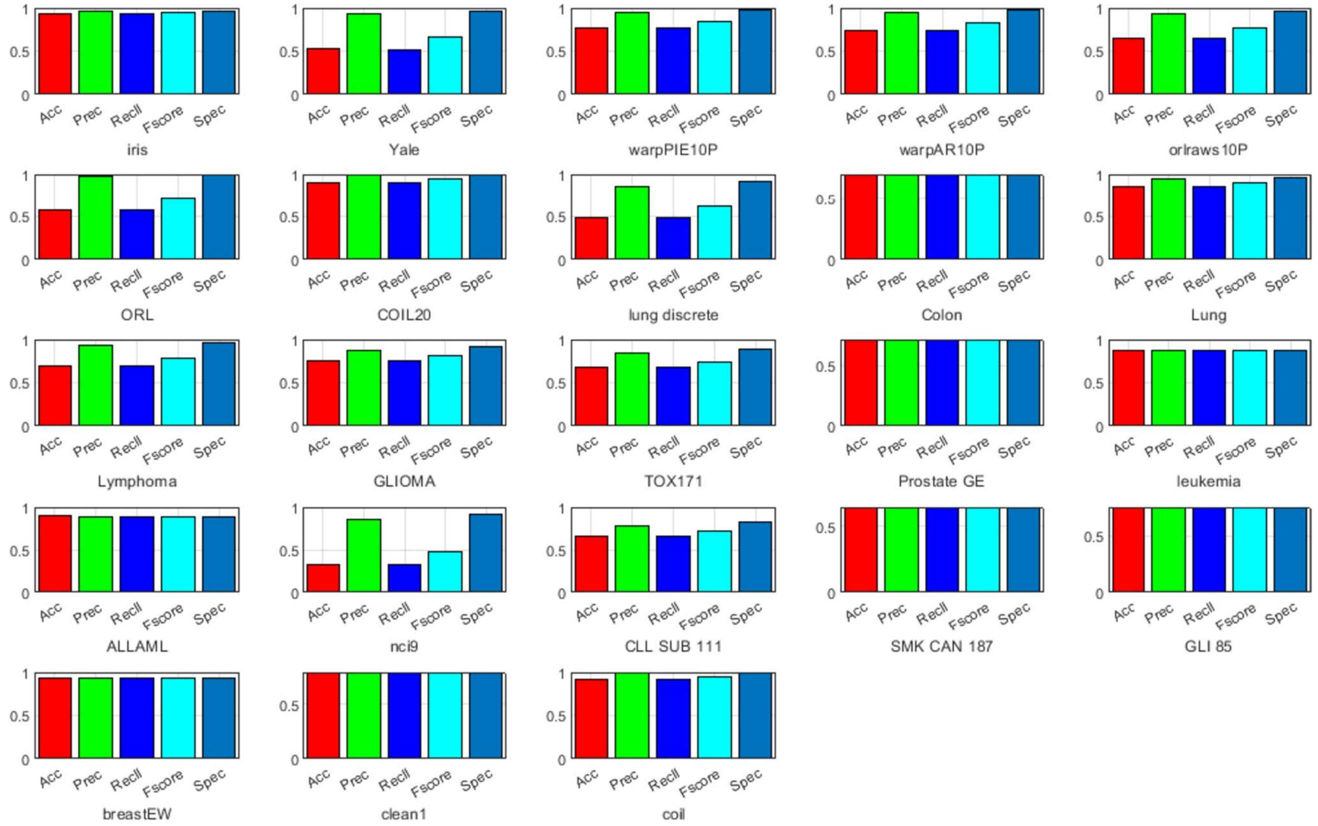


Fig. 17 Average classification results of DT classifier for 23 datasets

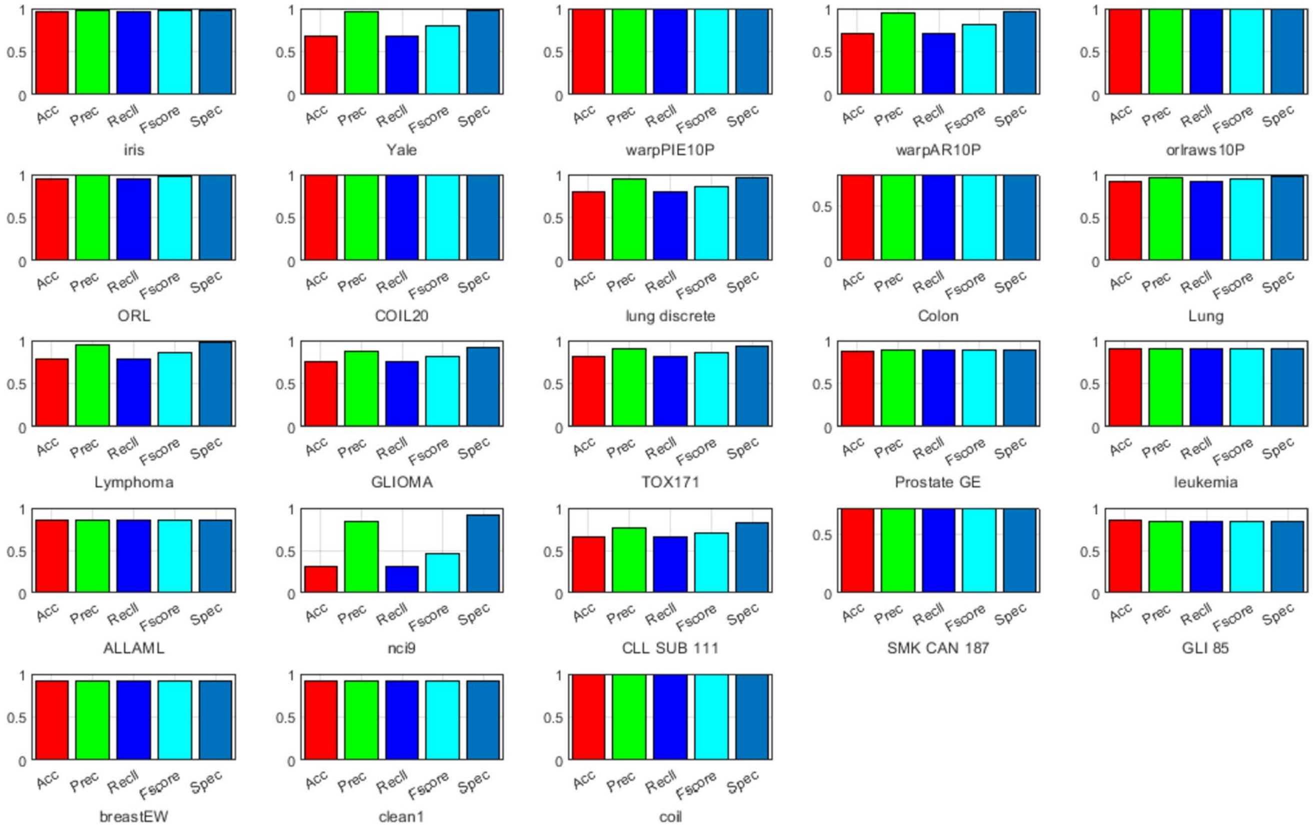


Fig. 18 Average classification results of SVM classifier for 23 datasets

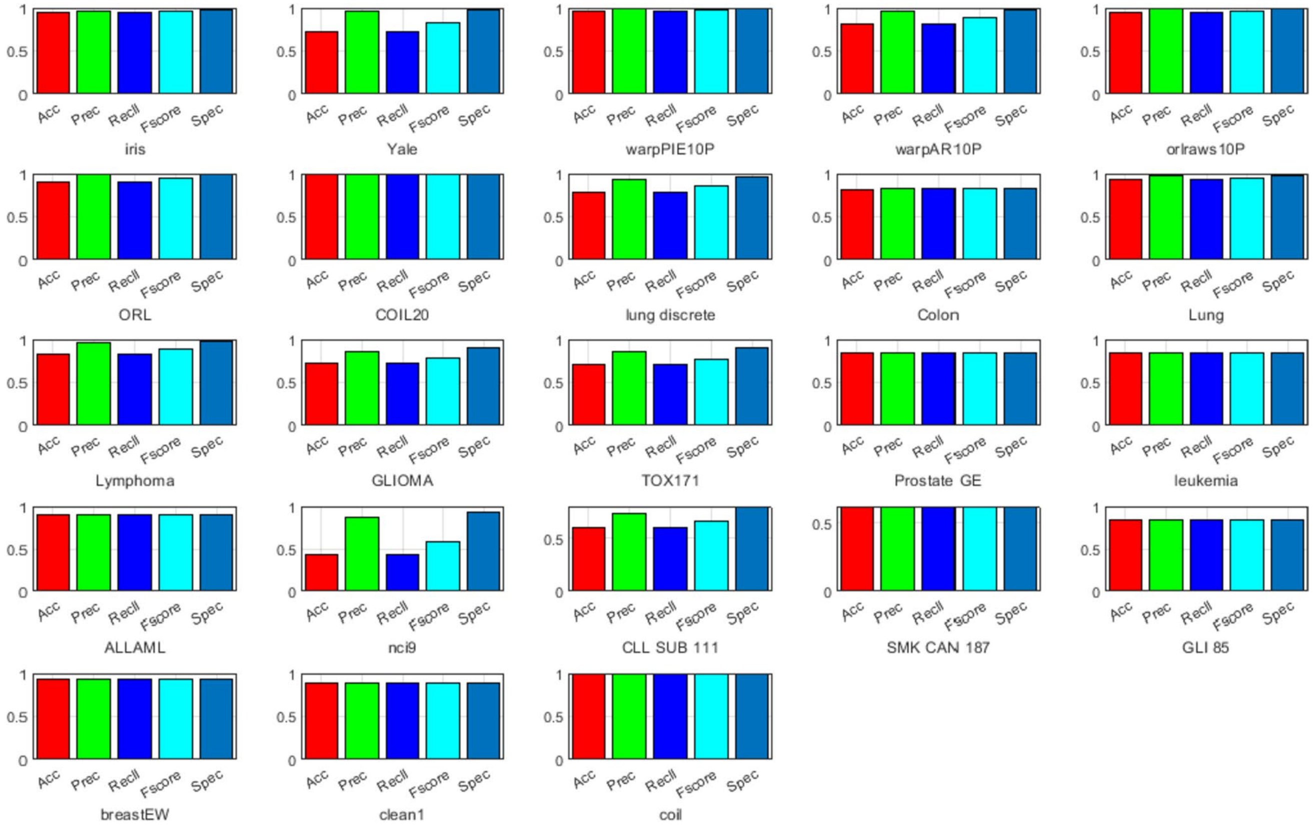


Fig. 19 Average classification metrics of RF classifier for 23 datasets

- The precision is high but not optimal since the used features for training and testing phases which is limited by the selected features by BChimp 1 and BChimp 2.
- No mechanism is investigated too deeply select features in unbalanced input datasets, particularly if the positive sample becomes highly sparse in high dimensional problems. during their selection, training, and testing phases. This is very important in high-dimensional data.

Future work will address these limitations.

7 Conclusion

This paper introduces a new hybrid approach in the feature selection field, which is based on enhanced binary chimp optimization algorithms (BChimp1 and BChimp2) combined with six standard classifiers. The approach was evaluated by using different types of datasets to select relevant features that help ensure highly accurate results of machine learning classifiers. Additionally, we compared the proposed approach to four widely used and efficient feature selection algorithms (GA, PSO, GWO, and WOA). The evaluation of the obtained results was achieved through well-known metrics such as dimensionality reduction and convergence rate. The results obtained are very encouraging and validate the efficiency of our approach. To confirm that the proposed feature selection process extracted the most significant and relevant attributes, we have applied the most standard classifiers, such as KNN, DT, SVM, DA, and MLP before and after the feature selection process to 23 benchmarks. In almost all cases, the classifier's accuracy is very similar before and after selecting features. In other cases, the accuracy of the classifier after feature selection is higher than the classifier before feature selection which means the relevance and significance of the proposed approach. It suffers from non-convergence of the recognition process for some datasets for both QDA and MLP classifiers. In general, the two proposed algorithms BChimp1-ML and BChimp2-ML give acceptable convergence time compared to the other techniques on 23 databases for all six learning machine classifiers. In future research, we aim to test BChimp1 and BChimp2 on medical datasets and provide multi-criteria dynamic optimization problems.

Acknowledgements The researchers would like to thank the Deanship of Scientific Research, Qassim University for funding the publication of this project.

Author Contributions All authors contributed to this paper as follows: study conception and design: FA; draft manuscript preparation: FA, AA. Supervision: AA. All authors reviewed the results and approved the final version of the manuscript and agreed to the published version of the manuscript.

Funding This research received no external funding.

Data Availability All resources from this project are available at https://www.mathworks.com/matlabcentral/fileexchange/133267-binary-chimp-optimization-algorithm-for-features-selection.

Declarations

Conflict of Interest The authors declare no conflict of interest.

Ethical approval Not applicable.

Consent to participate Not applicable.

Consent for publication Not applicable.

References

1. Hancer E, Xue B, Zhang M, Karaboga D, Akay B. Pareto front feature selection based on artificial bee colony optimization. Inf Sci. 2018;422:462–79. https://doi.org/10.1016/j.ins.2017.09.028.
2. Zhang L, Shan L, Wang J. Optimal feature selection using distance-based discrete firefly algorithm with mutual information criterion. Neural Comput Appl. 2017;28(9):2795–808. https://doi.org/10.1007/s00521-016-2204-0.
3. Liu H, Yu L. Toward integrating feature selection algorithms for classification and clustering. IEEE Trans Knowl Data Eng. 2005;17(4):491–502.
4. Chandrashekar G, Sahin F. A survey on feature selection methods. Comput Electr Eng. 2014;40(1):16–28.
5. Hu Z, Bao Y, Xiong T, Chiong R. Hybrid filter–wrapper feature selection for short-term load forecasting. Eng Appl Artif Intell. 2015;40:17–27.
6. Al-Tashi Q, Abdulkadir SJ, Rais HM, Mirjalili S, Alhussian H, Ragab MG, Alqushaibi A. Binary multi-objective grey wolf optimizer for feature selection in classification. IEEE Access. 2020;8:106247–63.
7. Emary E, Zawbaa HM, Hassanien AE. Binary grey wolf optimization approaches for feature selection. Neurocomputing. 2016;172:371–81. https://doi.org/10.1016/j.neucom.2015.06.083.
8. Faris H, Mafarja MM, Heidari AA, et al. An efficient binary Salp Swarm Algorithm with crossover scheme for feature selection problems. Knowl Based Syst. 2018;154:43–67. https://doi.org/10.1016/j.knosys.2018.05.009.

9. Oh IS, Lee JS, Moon BR. Hybrid genetic algorithms for feature selection. IEEE Trans Pattern Anal Mach Intell. 2004;26(11):1424–37.
10. Kennedy J. Particle swarm optimization. MA: Encyclopedia of machine learning Springer; 2011. p. 760–6.
11. Karaboga D, Basturk B. A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm. J Glob Optim. 2007;39:459–71. https://doi.org/10.1007/s10898-007-9149-x.
12. Dorigo M, Birattari M. Ant colony optimization. Boston, MA: Encyclopedia of machine learning. Springer; 2011. p. 36–9.
13. Too J, Mirjalili S. A hyperlearning binary dragonfly algorithm for feature selection: A COVID-19 case study. Knowl-Based Syst. 2021;212:106553.
14. Khishe M, Mosavi MR. Chimp optimization algorithm. Expert Syst Appl. 2020;149:113338.
15. Jia H, Sun K, Zhang W, Leng X. An enhanced chimp optimization algorithm for continuous optimization domains. Complex Intell Syst. 2022;8(1):65–82.
16. Wang D, Zhang H, Liu R, Lv W, Wang D. t-Test feature selection approach based on term frequency for text categorization. Pattern Recognit Lett. 2014;45:1–10.
17. Gao W, Hu L, Zhang P, He J. Feature selection considering the composition of feature relevancy. Pattern Recognit Lett. 2018;112:70–4. https://doi.org/10.1016/j.patrec.2018.06.005.
18. Labani M, Moradi P, Ahmadizar F, Jalili M. A novel multivariate filter method for feature selection in text classification problems. Eng Appl Artif Intell. 2018;70:25–37. https://doi.org/10.1016/j.engappai.2017.12.014.
19. Faramarzi A, Heidarinejad M, Mirjalili S, Gandomi AH. Marine Predators Algorithm: A nature-inspired metaheuristic. Expert Syst Appl. 2020;152:113377. https://doi.org/10.1016/j.eswa.2020.113377.
20. Krömer P, Platoš J, Nowaková J, Snášel V. Optimal column subset selection for image classification by genetic algorithms. Ann Oper Res. 2018;265(2):205–22.
21. Huang CL, Wang CJ. A GA-based feature selection and parameters optimization for support vector machines. Expert Syst Appl. 2006;31(2):231–40. https://doi.org/10.1016/j.eswa.2005.09.024.
22. Jiang S, Chin KS, Wang L, Qu G, Tsui KL. Modified genetic algorithm-based feature selection combined with pre-trained deep neural network for demand forecasting in outpatient department. Expert Syst Appl. 2017;82:216–30. https://doi.org/10.1016/j.eswa.2017.04.017.
23. Siedlecki W, Sklansky J. A note on genetic algorithms for large-scale feature selection. Pattern Recognit Lett. 1989;10(5):335–47. https://doi.org/10.1016/0167-8655(89)90037-8.
24. Nemati S, Basiri ME, Ghasem-Aghaee N, Aghdam MH. A novel ACO–GA hybrid algorithm for feature selection in protein function prediction. Expert Syst Appl. 2009;36(10):12086–94. https://doi.org/10.1016/j.eswa.2009.04.023.
25. De Stefano C, Fontanella F, Marrocco C, Di Freca AS. A GA-based feature selection approach with an application to handwritten character recognition. Pattern Recognit Lett. 2014;35:130–41. https://doi.org/10.1016/j.patrec.2013.01.026.
26. Rejer I. Genetic algorithm with aggressive mutation for feature selection in BCI feature space. Pattern Anal Appl. 2015;18(3):485–92. https://doi.org/10.1007/s10044-014-0425-3.
27. Too J, Abdullah AR. A new and fast rival genetic algorithm for feature selection. J Supercomput. 2021;77(3):2844–74. https://doi.org/10.1007/s11227-020-03378-9.
28. Poli R, Kennedy J, Blackwell T. Particle swarm optimization. Swarm Intell. 2007;1(1):33–57. https://doi.org/10.1109/ICNN.1995.488968.
29. Chuang LY, Chang HW, Tu CJ, Yang CH. Improved binary PSO for feature selection using gene expression data. Comput Biol Chem. 2008;32(1):29–38.
30. Unler A, Murat A. A discrete particle swarm optimization method for feature selection in binary classification problems. Eur J Oper Res. 2010;206(3):528–39.
31. Ayeche F, Alti A. HDG and HDGG: an extensible feature extraction descriptor for effective face and facial expressions recognition. Pattern Anal Appl. 2021;24:1095–110.
32. Tan TY, Zhang L, Neoh SC, Lim CP. Intelligent skin cancer detection using enhanced particle swarm optimization. Knowl-based Syst. 2018;158:118–35. https://doi.org/10.1016/j.knosys.2018.05.042.
33. Too J, Abdullah AR, Mohd Saad N. A new co-evolution binary particle swarm optimization with multiple inertia weight strategy for feature selection. Informatics. 2019;6(2):21.
34. Xue Y, Tang T, Pang W, Liu AX. Self-adaptive parameter and strategy-based particle swarm optimization for large-scale feature selection problems with multiple classifiers. Appl Soft Comput. 2020;88:106031. https://doi.org/10.1016/j.asoc.2019.106031.
35. Too J, Abdullah AR, Mohd Saad N. A new quadratic binary harris hawk optimization for feature selection. Electronics. 2019;8(10):1130. https://doi.org/10.3390/electronics8101130.
36. Nouri-Moghaddam B, Ghazanfari M, Fathian M. A novel multi-objective forest optimization algorithm for wrapper feature selection. Expert Syst Appl. 2021;175:114737.
37. Hegazy AE, Makhlouf MA, El-Tawel GS. Feature selection using chaotic salp swarm algorithm for data classification. Arab J Sci Eng. 2019;44(4):3801–16.
38. Niu B, Yi W, Tan L, Geng S, Wang H. A multi-objective feature selection method based on bacterial foraging optimization. Nat Comput. 2021;20(1):63–76. https://doi.org/10.1007/s11047-019-09754-6.
39. Piri, J., Mohapatra, P., Dey, R. (2021). Multi-objective ant lion optimization-based feature retrieval methodology for investigation of fetal wellbeing. In *2021 Third international conference on inventive research in computing applications (ICIRCA)* (pp. 1732-1737). IEEE.
40. Hashim FA, Houssein EH, Mabrouk MS, Al-Atabany W, Mirjalili S. Henry gas solubility optimization: A novel physics-based algorithm. Future Gener Comput Syst. 2019;101:646–67.
41. Kaveh A, Talatahari S. A novel heuristic optimization method: charged system search. Acta Mech. 2010;213(3):267–89.
42. Desuky AS, Hussain S, Kausar S, Islam MA, El Bakrawy LM. EAOA: an enhanced archimedes optimization algorithm for feature selection in classification. IEEE Access. 2021;9:120795–814.
43. Xue B, Zhang M, Browne WN. Particle swarm optimization for feature selection in classification: a multi-objective approach. IEEE Trans Cybern. 2013;43:1656–71. https://doi.org/10.1109/TSMCB.2012.2227469.
44. Xue B, Zhang M, Browne WN. Particle swarm optimization for feature selection in classification: novel initialization and updating mechanisms. Appl Soft Comput. 2014;18:261–76.
45. Alweshah M, Khalaileh SA, Gupta BB, et al. The monarch butterfly optimization algorithm for solving feature selection problems. Neural Comput Appl. 2020. https://doi.org/10.1007/s00521020-05210-0.
46. El-Hasnony IM, Barakat SI, El-Hasnony M, Mostafa RR. Improved feature selection model for big data analytics. IEEE Access. 2020;8:66989–7004. https://doi.org/10.1109/ACCESS.2020.2986232.
47. Datasets | Feature Selection @ ASU. http://featureselection.asu.edu/datasets.php . Accessed from 29 Oct 2022.49.
48. Abdel-Basset M, Abdle-Fatah L, Sangaiah AK. An improved Le´vy based whale optimization algorithm for Band width-efficient virtual

machine placement in cloud computing environment. Clust Comput. 2018. https://doi.org/10.1007/s10586-018-1769-z.

49. Piri J, Mohapatra P, Pradhan MR, Acharya B, Patra TK. A Binary Multi-Objective Chimp Optimizer with Dual Archive for Feature Selection in the Healthcare Domain. IEEE Access. 2022;10:1756–74. https://doi.org/10.1109/ACCESS.2021.3138403.
50. Pashaei E, Pashaei E. An efficient binary chimp optimization algorithm for feature selection in biomedical data classification. Neural Comput Appl. 2022;34:6427–51. https://doi.org/10.1007/s00521-021-06775-0.
51. Khishe M, Azar OP, Hashemzadeh E. Variable-length CNNs evolved by digitized chimp optimization algorithm for deep learning applications. Multimed Tools Appl. 2023. https://doi.org/10.1007/s11042-023-15411-z.
52. Cai C, Gou B, Khishe M, Mohammadi M, Rashidi S, Moradpour R, Mirjalili S. Improved deep convolutional neural networks using Chimp optimization algorithm for Covid19 diagnosis from the X-ray images. Expert Syst Appl. 2023;213:119206. https://doi.org/10.1016/j.eswa.2022.119206.
53. Qiuyu B, Wuqun C, Mohammad K. Evolving chimp optimization algorithm by weighted opposition-based technique and greedy search for multimodal engineering problems. Appl Soft Comput. 2023. https://doi.org/10.1016/j.asoc.2022.109869.
54. Khishe M, Orouji N, Mosavi MR. Multi-Objective chimp Optimizer: An innovative algorithm for Multi-Objective problems. Expert Syst Appl. 2023. https://doi.org/10.1016/j.eswa.2022.118734.
55. Saffari A, Khishe M, Zahiri SH. Fuzzy-ChOA an improved chimp optimization algorithm for marine mammal classification using artificial neural network. Analog Integr Circ Sig Process. 2022;111:403–17. https://doi.org/10.1007/s10470-022-02014-1.

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.