A Modified Binary Jaya Optimization Algorithm and its Application in Feature Selection

IJIMSR, Vol.1, No.1, (2023) 23.1.1.002

Manpreet Singh
Department: Computer Engineering Department
Institution: National Institute of Technology,
City Postal Code, Country: Kurukshetra, 136119, India
Clear and concise description of the reported work: Methodology and Formal Analysis.

Rajesh Ranjan
Department: Computer Engineering Department
Institution: National Institute of Technology
City Postal Code, Country: Kurukshetra, 136119, India
Clear and concise description of the reported work: Idea and Conceptualization, Writing - original draft.

Dr. Jitender Kumar Chhabra*
Department: Computer Engineering Department
Institution: National Institute of Technology
City Postal Code, Country: Kurukshetra, 136119, India
Clear and concise description of the reported work: Validation, Supervision, Writing - review & editing.
Email: jitenderchhabra@gmail.com

*Corresponding author

Received August 7th, 2023; Accepted August 22nd, 2023

Keywords: Feature Selection, Machine Learning, Filter, Wrapper, Jaya Optimization

A Modified Binary Jaya Optimization Algorithm and its Application in Feature Selection

Manpreet Singh
Department: Computer Engineering Department
Institution: National Institute of Technology,
City Postal Code, Country: Kurukshetra, 136119, India
Clear and concise description of the reported work: Methodology and Formal Analysis.

Rajesh Ranjan
Department: Computer Engineering Department
Institution: National Institute of Technology
City Postal Code, Country: Kurukshetra, 136119, India
Clear and concise description of the reported work: Idea and Conceptualization, Writing - original draft.

Dr. Jitender Kumar Chhabra*
Department: Computer Engineering Department
Institution: National Institute of Technology
City Postal Code, Country: Kurukshetra, 136119, India
Clear and concise description of the reported work: Validation, Supervision, Writing - review & editing.
Email: jitenderchhabra@gmail.com

ABSTRACT

In recent years rapid advancement of modern technologies has produced enormous and varied data which needs to be pre-processed before applying various machine learning techniques to gain valuable insight from the data. Feature Selection is an indispensable pre-processing step that helps to remove undesirable features which deteriorate the desired output from the various machine learning techniques. Further, it helps to wane the overall execution time. Metaheuristic algorithms have successfully applied as a wrapper approach for selecting those features which boost the overall outcome of machine learning techniques either in supervised or unsupervised form. The present work proposes a Modified Binary Jaya Optimization Algorithm as a wrapper for selecting the feature sub-set using K-NN as a classifier in a supervised Machine Learning task. In the proposed work, a unique initialization technique using Mutual information Coefficient as a Filter has been applied along with the Lévy Flight-based update mechanism, and a variable Mutation function is activated as the algorithm gets trapped in a locally optimal solution. The proposed work has been applied to ten significant benchmark classification datasets. The results show substantial improvement when compared with Binary Jaya Optimization Algorithm regarding average accuracy, precision, recall, F1-score, and feature size.

Keywords: Feature Selection, Machine Learning, Filter, Wrapper, Jaya Optimization

Received August 7th, 2023; Accepted August 22nd, 2023

1. INTRODUCTION

Recent technological advances have led to massive data generation from various sources. These data must be appropriately analyzed, pre-processed, and applied to various problems to extract valuable information. Usually, these data have several features which are optional for all the data-mining tasks. Some features are irrelevant, noisy, and redundant, which must be removed to obtain reliable and robust results. These unwanted features have been found to degrade the desired outcome and increase the computational cost of the task associated with the data [1]. Considering these two crucial issues, dimensionality reduction plays a significant role in various tasks related to the data-mining domain, such as classification, image and text categorization, clustering, and other pattern recognition task [2].

In the classification task under supervised machine learning, which involves training a model to classify the data into their appropriate classes, a vast set of features usually takes significant time to train the model. Hence, a relevant feature subset that excludes the noisy and duplicated feature is generally desirable to reduce the computational time and increase the overall accuracy level of the model [3].

According to various researchers, the dimensionality reduction technique can be broadly classified into Feature Extraction (FE) and Feature Selection (FS) [4]. Feature extraction transforms the original feature set into a new form with reduced dimension to avoid irrelevant, noisy, and duplicated features. Some of the important FE methods are Principal Component Analysis [5], Linear Discriminant Analysis [6], and Independent Component Analysis [7]. FE has been successfully applied to various classification tasks [8]. In the case of Feature Selection, a subset of the original feature set is chosen to improve the overall accuracy level and reduce the execution time. Feature Selection can be classified into Filter, Wrapper, and Embedded approach [9].

The filter approach selects the relevant features based on the inherent property of the data, such as statistical-based properties. This method usually scores each feature based on specific statistical or information-theoretic methods. Usually, they are less time-consuming, and most filter methods are univariate. In the case of the wrapper approach, a search mechanism such as backward elimination, forward selection, or recursive elimination methods is employed. During each successive iteration, the classification model is trained. The obtained accuracy level is evaluated for each unique feature subset; among them, the feature subset corresponding to the best accuracy level is selected as the optimum subset of features. The wrapper approach is comparatively slower than the Filter approach but gives better results than the previous. The embedded approach utilizes the goodness of both the filter and wrapper methods. In this approach, the feature selection is integrated into the learning algorithm itself. It avoids the over-fitting of the model but is computationally more expensive than the wrapper approach [10][11].

In recent times meta-heuristic-based algorithms have captivated the attention of several researchers from the machine learning domain for various optimization tasks. Feature selection is considered one of the complex optimization tasks as the number of feature sizes increases, and hence these nature-inspired algorithms are applied as a wrapper method for the feature selection task [12]. In the past, several such nature-inspired algorithms such as Particle Swarm Optimization (PSO) [13], genetic Algorithm (GA) [14], Ant Colony Optimization (ACO) [15], and Artificial Bee Colony (ABC)[16] have been applied for feature selection task in which the searching capability of these algorithms along with the classifier is utilized as a wrapper for the feature selection task. Recently several meta-heuristic approaches have been introduced by researchers for various optimization tasks. This newly introduced nature-inspired algorithm in its original form or hybridizing with other algorithms with different working principles is utilized as a wrapper for feature selection tasks [17]. It has been found that this hybridized approach usually produces better and more robust results than individual algorithms [18][19].

Jaya Optimization Algorithm (Jaya) is a recently proposed metaheuristic algorithm that gradually approaches the best solution obtained and simultaneously avoids the worst solution obtained during successive iterations [20]. In this way, the Jaya algorithm subsequently searches for new solution space, which is better in terms of fitness value than the previous solution obtained. As with another similar meat-heuristic approach, the Jaya algorithm also has specific parameters involved, which have been previously tuned according to the problem by various researchers [21][22]. In the present work, a Modified Binary Jaya Optimization Algorithm (MBJOA) has been proposed and applied as a wrapper for the feature selection problem. The proposed work is applied over ten benchmark datasets from the UCI repository with varying features, instances, and classes [23]. The results show that the Modified Binary Jaya Optimization algorithm produces better results than the Binary Jaya Optimization Algorithm (BJOA). The K-NN is applied as a classifier to obtain the accuracy of the selected feature subset [24].

The rest of the paper is designed below: section 2 discusses the literature review of the previous works, and the detailed methodology of MBJOA applied as a wrapper approach for feature selection is addressed in section 3; in section 4, obtained results are presented, and compared with BJOA, the overall outcome is concluded in section-5.

2. LITERATURE REVIEW

Researchers have utilized a meta-heuristic approach as a wrapper or in hybrid form with a suitable filter approach to select the optimal feature subset. Various evolutionary methods like GA, PSO, ACO, ABC, and other recently proposed meta-heuristic algorithms have been applied as feature selection wrappers. Some of the recent and notable works are discussed in this section.

Dong et al. [25] have proposed three strategies utilizing Binary Genetic Algorithm for selecting relevant features. In the first phase, granular computing theory is combined with a Genetic Algorithm to select essential features; after that, with the help of neighborhood rough set theory, best subset of features is chosen. Finally, in the third phase, optimal granularity parameters are selected. The classification accuracy was taken as a fitness measure to appraise the selected subset of features. Gokulnath and Shantharajah [26] applied Binary Genetic Algorithm using Support Vector Machine as a classification model to select the relevant features for classification. In this approach, a single-point crossover along with a mutation operator has been applied. The proposed work has produced better results than appropriate filter approaches such as Relief, Chi-Square, and Information Gain (IG). Abasabadi et al. [27] proposed a hybrid feature selection approach, which works in two phases. In the first phase, 99 % of the irrelevant features were removed using the sorted-label interface method. In the second phase, the GA-based wrapper approach is utilized to optimize the solution obtained during the first phase.

Jain et al. [28] have proposed a hybrid approach for feature selection incorporating a Correlation-based filter approach with Improved Binary PSO. The proposed methodology utilized the Naïve Bayes method as a classification model. At first, with the help of the filter method, the extraneous and superfluous genes are removed, and the missing values are imputed with their mean value. Finally, through a ten-fold cross-validation approach, the Binary PSO is used as a wrapper for selecting the best feature subset. The proposed work is tested over 11 cancer micro-array data and has been compared with several similar and filter approaches. Langeveld, and Engelbrecht [29] presented Set-based PSO as a wrapper for feature selection. In the presented work, KNN is applied as a classifier. In this work particle's position and velocity are defined as a mathematical set. SBPSO has produced better results when compared with BPSO, Catfish Binary PSO, and Probability PSO over thirty benchmark datasets.

Ghosh et al. [30] proposed a wrapper-filter hybrid approach for feature selection using ACO. The proposed work uses a similarity-based filter approach which first selects relevant features, and the quality of selected features is further assessed through a wrapper-based system. Based on pheromone density, the candidate feature is added to an eclectic subset of features. Additionally, the given approach uses memory to maintain the best-selected feature set. The work has been compared with several binary variants of ACO, ABC, and PSO over ten benchmark datasets for classification. Manoj et al. [31] proposed Artificial Neural Network (ANN) based ACO for text feature selection. The ACO-ANN approach is a hybrid technique used for feature selection in text classification. The ACO algorithm demonstrated proficient exploration abilities within the problem domain, resulting in better convergence and successful finding of the feature subset. The methodology described in this study utilized ACO to evaluate the selection process. Additionally, ANN is employed to identify the optimal subset from the given subsets.

Shunmugapriya and Kanmani [32] proposed a novel method for feature selection by hybridizing ABC with ACO. By utilizing the best characteristics of both algorithms, the proposed hybrid wrapper approach has produced better classification accuracy, feature-subset size, and convergence rate. Hancer et al. [33] proposed a multi-objective enhanced artificial bee colony algorithm that utilizes genetic operators for feature selection tasks. The authors also created two variations of the algorithm: a binary version and a continuous version. The experimental findings indicate that the binary version performs better than the continuous version in most cases.

Numerous recently developed meta-heuristic techniques have been implemented as a wrapper approach for feature selection tasks in recent works. These approaches have shown that they can tackle difficult optimization problems, which led to their application in these works. Chaudhary and Sahu [34] proposed a Binary Jaya optimization algorithm utilizing a time-varying transfer function for feature selection in microarray data utilizing five different filter methods having varying working principles. The obtained result over ten micro-array gene selection datasets has surpassed other similar works in accuracy and convergence rate. In another prominent work, Chaudhary and Sahu [35] proposed three variants of the Binary Jaya algorithm for feature selection problems as a wrapper method in another well-known work. The authors applied sigmoid, hyperbolic (tanh), and transfer functions based on Jaccard similarity. The Naïve Bayes classifier is used to authenticate the selected feature subset. The proposed work is applied over 18 benchmark datasets from the UCI repository and compared with several binary-formed meta-heuristic approaches.

Das et al. [36] proposed FSJaya for feature selection using the Jaya optimization algorithm. The proposed approach improves classification accuracy by selecting a relevant set of features and reducing the feature size. The FSJaya approach was evaluated using four classifiers, NB, KNN, LDA, and RT, on various benchmark datasets with varying dimensions. The experiment results demonstrate that the FSJaya method can successfully remove unnecessary features and achieve superior performance compared to the FSGA, FSPSO, and FSDE methods. Baliarsingh et al. [37] proposed a hybrid approach for feature selection by combining Forest Optimization Algorithm (FOA) with Enhanced Jaya Optimization Algorithm (EJaya) over micro-array data. At first, the ANOVA filter is applied to remove the irrelevant and noisy dataset. After that, a hybrid metaheuristic approach is used as a wrapper over the selected features to obtain a better subset of features by applying Support Vector Machine (SVM) classifier.

3. METHODOLOGY

Jaya optimization algorithm is a recently proposed metaheuristic approach applied in several real-life optimization problems from various domains [38][39]. This work proposes a Modified Binary Jaya Optimization algorithm, applied as a hybrid approach for feature selection task. Due to the ever-increasing data size, selecting the most relevant feature subset from the original feature set is challenging. Besides this, a small-sized feature subset has better accuracy and is more efficient regarding time complexity.

The proposed work proposes a hybrid approach by first utilizing a filter method in the initialization process and then using the modified version of the Jaya optimization algorithm as a wrapper for selecting the relevant features. The Maximal Information Coefficient (MIC) is used as a filter method to initialize the particles involved in the proposed work partially. Details of MIC are given in the following sub-section.

3.1. Filter Measure:

Maximal Information Coefficient (MIC): The Maximal Information Coefficient [40] is a statistical measure used to assess the level of correlation between two variables. This measure is founded on the presumption that a grid can be drawn on the scatterplot of the two variables in question, which can then be used to partition the data and capture any relationship between the variables. The MIC is both comprehensive and equitable. Additionally, it can detect significant correlations between two variables with a wide range of values.
Mathematically, it can be defined as follows:

$$MIC(A,B) = \max_{n_a \times n_b \leq B(n,\alpha)} \left\{ \frac{\max_G \left(I_G(A,B) \right)}{\log_2 min(n_a, n_b)} \right\} \quad (1)$$

Here numerator represents the Mutual Information (MI) over grid size G represented by $n_a \times n_b$ and $B(n,\alpha)$ is a function of instance size n and is given as n^{α} where α is taken as 0.6.

3.2. Binary Jaya Optimization Algorithm

Feature selection using the wrapper approach requires a search algorithm that searches the optimal feature subset. Researchers have proposed several search techniques, such as exhaustive search, random search, and heuristic search. As the number of features increases, these search algorithms become more and more computationally expensive. Considering the complexity of feature selection as an optimization problem, nature inspired algorithms known as the meta-heuristic approach have been frequently applied in recent times for selecting the feature sub-set from the original set of features. The present work proposes a modified Binary Jaya algorithm for selecting relevant feature subsets, improving the overall accuracy compared to the binary Jaya algorithm.

First, the Jaya optimization algorithm is converted into binary form. So, to alter the actual encoding to binary form, the famous S-shaped sigmoid function is applied [41]. The range of the sigmoid function is between 0 and 1. And with the help of the threshold function, a randomly generated number between 0 and 1, the real encoded crow's position is converted into binary form. The mathematical definition of the sigmoid function is given below:

$$S\left(Z_{i,d}^{iter+1}\right) = \frac{1}{1+e^{-Z_{i,d}^{iter+1}}} \quad (2)$$

$$Z_{i,d}^{iter+1}(Binary) = \begin{cases} 1, & if\ S\left(Z_{i,d}^{iter+1}\right) > \sigma \\ 0, & otherwise \end{cases} \quad (3)$$

Where, σ is a random number between [0,1]. $Z_{i,d}^{iter+1}$, $Z_{i,d}^{iter+1}(Binary)$, is the i^{th} particle in the continuous and in the binary form having size d at iteration number "*iter*+1". Each particle in the Binary Jaya algorithm is randomly initialized to binary from i.e., either in 1 or 0 form. The size of the particle is kept equal to the feature size of the dataset. Hence for each individual particle the 1 denotes the feature is selected whereas 0 represents that the feature is not selected.

3.3 Modified Binary Jaya Optimization Algorithm

Jaya optimization algorithm has only a few parameters, and in the case of discrete optimization problems like Feature selection, it fails to give the best optimum solution. Hence, a modified version of the Binary Jaya optimization algorithm is proposed in the present work. The modifications proposed by us in the Jaya optimization algorithm are as follows:

1. Around 20% of the total population is initialized through the filter technique.
2. Instead of using random numbers while updating the particles, a Lévy-flight based updating mechanism has been implemented.
3. After regular intervals to bring diversity to the solution, a mutation function is applied to some particles as described below in Modified Binary Jaya Optimization Algorithm (Algorithm -1).

The flow diagram and the algorithm of the proposed work are given below. As seen from the algorithm, apart from the usual particle update, a mutation operator is applied when there is no significant improvement in the best particle for four consecutive iterations, and it is checked after every five iterations. The mutation rate of the operator varies with each successive iteration. The primary purpose of applying a mutation operator is to bring diversity to the solution and avoid the local optimal solution. Besides this, a hybrid initialization approach is used, which initializes 20% of the total population through a filter technique, and the remaining 80% is initialized randomly.

4. RESULTS AND DISCUSSION

4.1 Parameters Setting

The work is simulated in Python 3.6 with an i7 processor and 16 GB RAM on an Ubuntu 20.04 operating system. The parameters of the Binary Jaya Optimization Algorithm and Modified Binary Jaya Optimization Algorithm are kept similar for an unbiased comparison. The number of particles is kept at 50, and the Maximum iteration is kept at 100. K-NN and its four other variants, i.e., Local Mean Vector KNN (LMKNN) [42], Local Mean Pseudo Nearest Neighbor (LMPNN) [43], Generalized Mean Distance Vector KNN (GMDKNN) [44] and Harmonic Mean Distance Based KNN (MLMKHNN) [45] are used as classifiers in the present work. The training and testing ratio is kept at 70:30, and during the training phase, 5-fold cross-validation is performed.

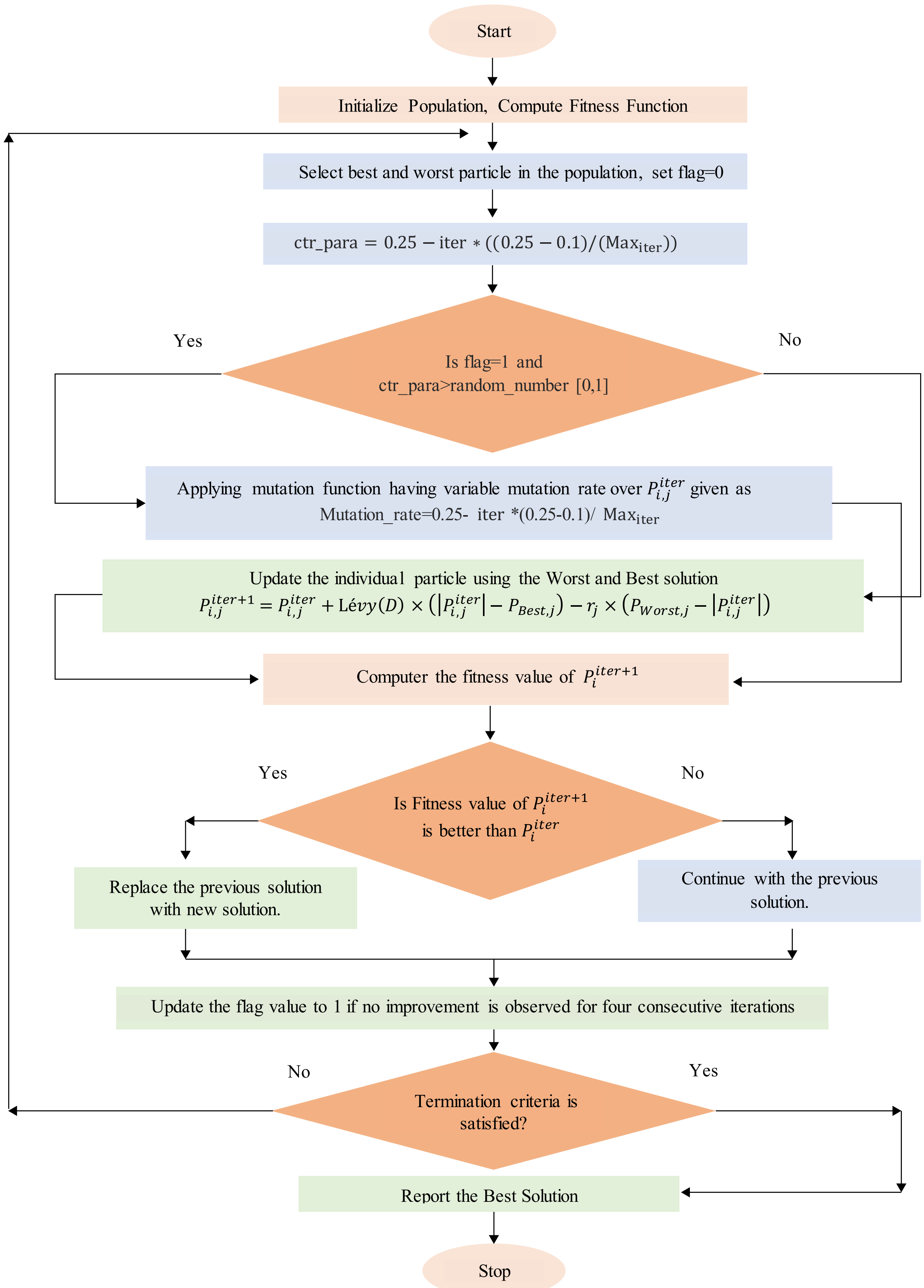

Figure-1. Modfied Binary Jaya Optimization Algorithm

$\boldsymbol{Algorithm\ 1: Modified\ Binary\ Jaya\ Optimization\ Algorithm (MBJOA)}$

$Parametes:$
$Max_{iter}: Maximum\ iterations$
$N: Number\ of\ particles$
$D: Size\ of\ Particle$
$Set\ iter = 0$
$Initialize\ through\ Hybrid\ Initialization\ Algorithm\ (Algorithm\ 2)$

$while\ iter < Max_{iter}$
$flag = 0$
$Compute\ the\ fitness\ function\ of\ all\ the\ N\ Particles$
$Find\ the\ Best\ and\ Worst\ Particle\ in\ terms\ of\ fitness\ values$
$ctr\_para = 0.25 - iter * ((0.25 - 0.1)/(Max_{iter}))$
$for\ i = 1: N$
$if (flag == 1\ and\ ctr\_para > random\_number[0,1])$
$P_i^{iter+1} = Mutation\ Function\left(P_i^{iter}\right)$
$else$
$r_j = random\_uniform(0,1, D)$
$for\ j = 1: D$
$P_{i,j}^{iter+1} = P_{i,j}^{iter} + \text{Lévy}(D) \times \left(\left|P_{i,j}^{iter}\right| - P_{Best,j}\right) - r_j \times \left(P_{Worst,j} - \left|P_{i,j}^{iter}\right|\right)$
$//Replace\ the\ current\ solution\ in\ case\ of\ improvement$
$if\left(fitness\left(P_i^{iter+1}\right) \le fitness\left(P_i^{iter}\right)\right)$
$P_i^{iter} = P_i^{iter+1}$
$if (iter > 5\ and\ iter\%5 == 0)$
$if (Best[iter - 1] == Best[iter - 2]\ and\ Best[iter - 2] == Best[iter - 3])$
$flag = 1$
$else$
$flag = 0$
$End\ while$
$Return\ the\ Best\ solution$

Algorithm -1. Modified Binary Jaya Optimization Algorithm

$\boldsymbol{Algorithm\ 2: Hybrid\ Initialization\ Algorithm}$

$Parametes:$
$N: Number\ of\ particles$
$D: Size\ of\ Particle$
$Rank: Ranking\ of\ Features\ based\ on\ MIC\ value$

$for\ i = 1: 0.8 * N$
$for\ j = 1: D$
$P_{i,j} = Random\_Choice[0,1]$
$for\ i = 0.8 * N: N$
$count = int\left(D * random\_uniform(0.35,0.75)\right)$
$final\_rank = rank[0: count]$
$for\ j = 1: D$
$if\ (j\ in\ final\_rank)$
$P_{i,j} = 1$
$else$
$P_{i,j} = 0$
$return\ N$

Algorithm -2. Hybrid Initialization Algorithm

4.2 Dataset Details

To analyze the performance of the MBJOA with BJOA, the author has experimented with ten real datasets obtained from the UCI repository [23]. Datasets details are given in Table-1.

Table-1. Dataset Description

Dataset	Features	Instances	Class
Sonar	60	208	2
Dermatology	34	366	6
Ionosphere	34	351	2
Musk	168	476	2
Pima	9	768	2
ParkinsonC	753	755	2
WDBC	30	569	2
Wine	13	178	3
Vehicle	18	846	4
SPECTF	43	266	2

4.3 Simulation Results

Table-2 compares the accuracy of selected datasets-based Feature Selection (FS) using Binary Jaya Optimization Algorithm (BJOA) vs. Without Feature Selection (WFS). Best results are shown in boldface letters, and the standard deviation of each outcome is shown with a ‘ ’ sign. Five K-NN variants are applied to calculate the results. The results in the table are the average accuracy of 20 runs with the standard deviation of each machine learning algorithm after proper parameter tuning.

As shown in Table-2, MLMKHNN gives the best performance over average results without feature selection; GMDKNN gives the best on average performance with feature selection.

Table-2: Accuracy comparison of selected features with unselected features using five K-NN variants

Datasets	KNN		LMKNN		MLMKHNN		LMPNN		GMDKNN	
	WFS	FS	WFS	FS	WFS	FS	WFS	FS	WFS	FS
Sonar	0.85144 ±0.0385	0.81777 ±0.0213	0.85217 ±0.0403	0.85523 ±0.0203	**0.89130** ±0.0392	0.86507 ±0.0174	0.87753 ±0.0384	0.86206 ±0.0216	0.87898 ±0.0381	0.87444 ±0.0186
Dermatology	0.96074 ±0.0175	0.96845 ±0.0084	**0.97479** ±0.0095	0.96136 ±0.0052	0.97190 ±0.0112	0.96090 ±0.0072	0.96611 ±0.0147	0.95909 ±0.0081	0.97314 ±0.0156	0.96009 ±0.0075
Ionosphere	0.85560 ±0.0311	0.8756 ±0.0117	0.88232 ±0.0244	0.88556 ±0.0175	0.8875 ±0.0327	0.895 ±0.0144	0.89137 ±0.0222	0.89396 ±0.0141	0.88103 ±0.0240	**0.90235** ±0.0130
Musk	0.84493 ±0.0271	0.83307 ±0.0168	0.88132 ±0.0221	**0.89377** ±0.0165	0.88417 ±0.0252	0.89349 ±0.0109	0.89113 ±0.0245	0.88167 ±0.0176	0.88765 ±0.0257	0.88937 ±0.0115
Pima	**0.73838** ±0.0257	0.72965 ±0.0125	0.73070 ±0.0256	0.7071 ±0.0140	0.72185 ±0.0171	0.70121 ±0.0191	0.72421 ±0.0304	0.70956 ±0.0135	0.72618 ±0.0247	0.7101 ±0.0109
Parkinson	0.92000 ±0.0208	0.86929 ±0.0067	0.90579 ±0.0145	0.90277 ±0.0095	0.9314 ±0.0186	0.92585 ±0.0091	0.9314 ±0.0151	0.92497 ±0.0092	**0.93780** ±0.0154	0.92585 ±0.0085
WDBC	0.96728 ±0.0109	0.96608 ±0.0049	**0.96861** ±0.0082	0.95836 ±0.0059	0.96728 ±0.0104	0.95684 ±0.0057	0.95957 ±0.0134	0.95573 ±0.0072	**0.96861** ±0.0127	0.95736 ±0.0055
Vehicle	0.68696 ±0.0277	0.71842 ±0.0138	0.72982 ±0.0204	**0.73779** ±0.0167	0.71839 ±0.0176	0.73543 ±0.0182	0.71017 ±0.0233	0.72393 ±0.0148	0.71446 ±0.0226	0.72681 ±0.0165
Wine	**0.97542** ±0.0173	0.96944 ±0.0111	0.96694 ±0.0173	0.96481 ±0.0109	0.96949 ±0.0182	0.96388 ±0.0149	0.97457 ±0.0173	0.96351 ±0.0107	0.95932 ±0.0229	0.96555 ±0.0124
SPECTF	0.75112 ±0.0176	0.77062 ±0.0228	0.76250 ±0.0150	0.773 ±0.0248	0.75024 ±0.0162	0.76112 ±0.0244	0.73425 ±0.0197	0.75987 ±0.0260	0.74450 ±0.0199	**0.77125** ±0.0233
Average	0.85518	0.85183	0.86549	0.86397	**0.86935**	0.86587	0.86603	0.86343	0.86716	0.86831

MLMKHNN without feature selection provides 86.93% average accuracy, and GMDKNN with feature selection provides 86.83% average accuracy. Feature selection slightly degrades the performance as the difference between average accuracy is 0.01%, which is negligible. Based on Table-2, feature selection can make K-NN faster without affecting accuracy. Based on the average accuracy value, we cannot say that feature selection does not affect accuracy. So, we are comparing it based on box plots. Figure-2 shows the comparison of feature selection without feature selection based on five variants of K-NN.

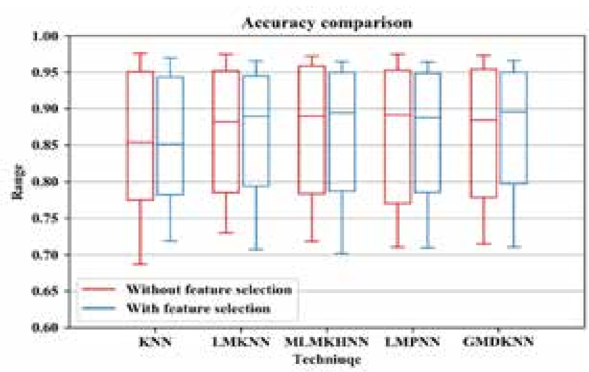

Figure-2: Accuracy comparison of K-NN variants

As shown in Figure-2, in the case of GMDKNN, MLMKHNN, and LMKNN, feature selection improves the prediction accuracy. GMDKNN provides the best results with feature selection based on Figure-2. The median value of blue coloured box plot of GMDKNN is higher than all other box plots. The accuracy of all datasets is equally distributed, as no box plot contains outliers.

Two optimizers Binary Jaya Optimization Algorithm (BJOA) and the Modified Binary Jaya Optimization Algorithm (MBJOA), are used for feature selection in the present work, and Table-3 shows the accuracy comparison of BJOA with MBJOA. MBJOA improves the prediction accuracy of all K-NN variants proved by the average column of Table-3. The best improvement is achieved in the case of GMDKNN by feature selection based on MBJOA. Out of 10 datasets, GMDKNN gives the best results in 6 datasets, K-NN and LMKNN provide the best results in 1 dataset each, and MLMKHNN gives the best results in the case of 2 datasets.

Table-3: Accuracy comparison based on selected features using BJOA and MBJOA

Datasets	KNN		LMKNN		MLMKHNN		LMPNN		GMDKNN	
	MBJOA	BJOA	MBJOA	BJOA	MBJOA	BJOA	MBJOA	BJOA	MBJOA	BJOA
Sonar	0.86231 ±0.0305	0.81777 ±0.0213	0.84492 ±0.0380	0.85523 ±0.0203	0.86884 ±0.0379	0.86507 ±0.0174	0.86739 ±0.0349	0.86206 ±0.0216	0.87463 ±0.0334	0.87444 ±0.0186
Dermatology	0.96570 ±0.0138	0.96845 ±0.0084	0.97190 ±0.0112	0.96136 ±0.0052	0.97148 ±0.0121	0.96090 ±0.0072	0.97231 ±0.0091	0.95909 ±0.0081	0.97272 ±0.0125	0.96009 ±0.0075
Ionosphere	0.88879 ±0.0341	0.8756 ±0.0117	0.89568 ±0.0207	0.88556 ±0.0175	0.90431 ±0.0262	0.895 ±0.0144	0.90000 ±0.0211	0.89396 ±0.0141	0.90991 ±0.0257	0.90235 ±0.0130
Musk	0.85474 ±0.0345	0.83307 ±0.0168	0.89398 ±0.0232	0.89377 ±0.0165	0.90822 ±0.0194	0.89349 ±0.0109	0.88544 ±0.0193	0.88167 ±0.0176	0.89462 ±0.0183	0.88937 ±0.0115
Pima	0.74881 ±0.0237	0.72965 ±0.0125	0.71259 ±0.0197	0.7071 ±0.0140	0.71574 ±0.0281	0.70121 ±0.0191	0.70669 ±0.0204	0.70956 ±0.0135	0.73011 ±0.0217	0.7101 ±0.0109
Parkinson	0.92000 ±0.0152	0.86929 ±0.0067	0.90960 ±0.0144	0.90277 ±0.0095	0.93260 ±0.0159	0.92585 ±0.0091	0.93600 ±0.0181	0.92497 ±0.0092	0.93279 ±0.0175	0.92585 ±0.0085
WDBC	0.96170 ±0.0174	0.96608 ±0.0049	0.96648 ±0.0135	0.95836 ±0.0059	0.96622 ±0.0150	0.95684 ±0.0057	0.96170 ±0.0104	0.95573 ±0.0072	0.97127 ±0.0104	0.95736 ±0.0055
Vehicle	0.71446 ±0.0271	0.71842 ±0.0138	0.74285 ±0.0327	0.73779 ±0.0167	0.74785 ±0.0245	0.73543 ±0.0182	0.73285 ±0.0323	0.72393 ±0.0148	0.73374 ±0.0264	0.72681 ±0.0165
Wine	0.96864 ±0.0229	0.96944 ±0.0111	0.97203 ±0.0279	0.96481 ±0.0109	0.96864 ±0.0202	0.96388 ±0.0149	0.97118 ±0.0234	0.96351 ±0.0107	0.97796 ±0.0161	0.96555 ±0.0124
SPECTF	0.76987 ±0.0244	0.77062 ±0.0228	0.78175 ±0.0216	0.773 ±0.0248	0.77312 ±0.0237	0.76112 ±0.0244	0.77087 ±0.0289	0.75987 ±0.0260	0.77362 ±0.0238	0.77125 ±0.0233
Average	0.86550	0.85183	0.86917	0.86397	0.87570	0.86587	0.87044	0.86343	0.87713	0.86831

For all ten datasets, selected features using MBJOA provide the best results, and GMDKNN shows the best improvement, which is more than 1%. Figure- 3 compares the performance of two feature selection techniques based on five K-NN variants in terms of bar-graph. From the figure it is clear that the proposed MBJOA provides better results than the existing optimizer.

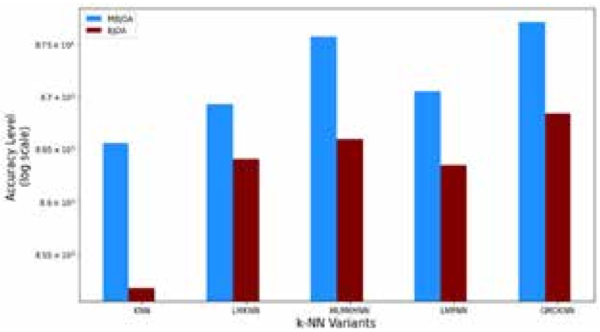

Figure-3: Average Accuracy level of k-NN variants

Based on Figure-3 and Table-3, it can be concluded that feature selection improves the prediction accuracy of the K-NN, and the proposed modification in the Jaya algorithm (MBJOA) selects the most relevant features.

However, datasets are imbalanced, so comparing other performance metrics such as precision, recall, and F1-score is necessary. Table-4 compares both feature selection techniques based on precision performance metric. The best results in Table-4 are shown in boldface letters.

In the case of 7 datasets, feature selection with the proposed optimizer provides better results, as shown in boldface letters. However, on average,

GMDKNN with Jaya gives the best results, and the difference is very slight. Table-5 compares both feature selection techniques based on recall performance metrics. The best results in Table-5 are shown in boldface letters. In the case of all ten datasets, feature selection with the proposed optimizer provides better results, as shown in boldface letters.

Table-4: Precision comparison of feature selection techniques

Datasets	KNN		LMKNN		MLMKHNN		LMPNN		GMDKNN	
	MBJOA	**BJOA**	**MBJOA**	**BJOA**	**MBJOA**	**BJOA**	**MBJOA**	**BJOA**	**MBJOA**	**BJOA**
Sonar	0.86821	0.82769	0.85280	0.87058	0.87491	0.88158	0.87156	0.87726	0.88050	**0.88933**
Dermatology	0.96869	0.97176	0.97389	0.97321	0.97317	0.97337	0.97409	0.97146	**0.97526**	0.97254
Ionosphere	0.90035	0.88766	0.89801	0.89874	0.90224	0.9103	0.90671	0.91069	0.91543	**0.91802**
Musk	0.86333	0.8513	0.89559	0.90542	**0.90997**	0.90623	0.89109	0.89751	0.89698	0.90271
Pima	**0.74390**	0.72531	0.71237	0.71882	0.71774	0.7113	0.70721	0.70886	0.72762	0.71888
Parkinson	0.92018	0.86773	0.90844	0.91256	0.93230	0.93565	**0.93635**	0.93490	0.93226	0.93614
WDBC	0.96261	0.96676	0.96699	0.96871	0.96656	0.96735	0.96214	0.96619	**0.97149**	0.96787
Vehicle	0.71345	0.71005	0.74106	0.74835	**0.74869**	0.74450	0.73061	0.73271	0.73092	0.73572
Wine	0.97133	0.97202	0.97364	0.97659	0.97052	0.97565	0.97278	0.97547	**0.97928**	0.97738
SPECT	0.77166	0.77028	0.76830	0.77061	0.76870	0.7649	**0.77309**	0.76808	0.76684	0.77265
Average	0.868371	0.85505	0.86910	0.87435	0.87648	0.87708	0.87256	0.87431	0.87765	**0.87912**

Table-5: Recall comparison of feature selection techniques

Datasets	KNN		LMKNN		MLMKHNN		LMPNN		GMDKNN	
	MBJOA	BJOA	MBJOA	BJOA	MBJOA	BJOA	MBJOA	BJOA	MBJOA	BJOA
Sonar	0.86231	0.81777	0.84492	0.85523	0.86884	0.86507	0.86739	0.86206	**0.87463**	0.87444
Dermatology	0.96570	0.96845	0.97190	0.96136	0.97148	0.96090	0.97231	0.95909	**0.97272**	0.96009
Ionosphere	0.88879	0.8756	0.89568	0.88556	0.90431	0.895	0.90000	0.89396	**0.90991**	0.90235
Musk	0.85474	0.83307	0.89398	0.89377	**0.90822**	0.89349	0.88544	0.88167	0.89462	0.88937
Pima	**0.74881**	0.72965	0.71259	0.7071	0.71574	0.70121	0.70669	0.70956	0.73011	0.7101
Parkinson	0.92000	0.86929	0.90960	0.90277	0.93260	0.92585	**0.93600**	0.92497	0.93279	0.92585
WDBC	0.96170	0.96608	0.96648	0.95836	0.96622	0.95684	0.96170	0.95573	**0.97127**	0.95736
Vehicle	0.71446	0.71842	0.74285	0.73779	**0.74785**	0.73543	0.73285	0.72393	0.73374	0.72681
Wine	0.96864	0.96944	0.97203	0.96481	0.96864	0.96388	0.97118	0.96351	**0.97796**	0.96555
SPECTF	0.76987	0.77062	**0.78175**	0.773	0.77312	0.76112	0.77087	0.75987	0.77362	0.77125
Average	0.86550	0.85183	0.86917	0.86397	0.87570	0.86587	0.87044	0.86343	**0.87713**	0.86831

On average, GMDKNN with modified Jaya (MBJOA) gives the best results. Table-6 compares both feature selection techniques based on the F1-score performance metric. The best results in Table-6 are shown in boldface letters. In the case of six datasets, feature selection with the proposed optimizer provides better results, as shown in boldface letters. On average, GMDKNN with Jaya gives the best results.

Table-7 presents the average feature size obtained through BJOA and MBJOA; the value in braces represents the average value obtained over 20 iterations, and values without braces represent the round-off value. As seen from the accepted value, both BJOA and MBJOA have obtained an almost similar number of features.

Table-6: F1-Score comparison of feature selection techniques

Datasets	KNN		LMKNN		MLMKHNN		LMPNN		GMDKNN	
	MBJOA	BJOA	MBJOA	BJOA	MBJOA	BJOA	MBJOA	BJOA	MBJOA	BJOA
Sonar	0.86117	0.81611	0.84445	0.86495	0.86826	0.87477	0.86663	0.87140	0.87385	**0.8840**
Dermatology	0.96586	0.96876	0.97189	0.97131	0.97164	0.97097	0.97234	0.96924	**0.97290**	0.97021
Ionosphere	0.88420	0.86973	0.89359	0.89355	0.90161	0.90243	0.89674	0.90078	0.90728	**0.90982**
Musk	0.85520	0.83382	0.89399	0.90364	**0.90839**	0.90369	0.88592	0.89208	0.89468	0.89961
Pima	0.73966	0.72425	0.71076	0.7167	0.71502	0.71022	0.70570	0.70786	**0.72774**	0.71791
Parkinson	0.91951	0.86192	0.90743	0.91047	0.93151	0.93467	0.93550	0.93411	0.93154	**0.93497**
WDBC	0.96144	0.96589	0.96638	0.96827	0.96612	0.96674	0.96170	0.9656	**0.97120**	0.96728
Vehicle	0.70347	0.70693	0.74032	0.74588	**0.74669**	0.74286	0.73027	0.73089	0.73028	0.73381
Wine	0.96852	0.96941	0.97203	0.97486	0.96869	0.97387	0.97118	0.97353	**0.97797**	0.97552
SPECTF	0.76588	0.76559	0.76880	0.77190	0.76609	0.76454	0.76874	0.76486	0.76636	**0.7727**
Average	0.86249	0.84824	0.86696	0.87215	0.87440	0.87447	0.86947	0.87103	0.87538	**0.87658**

Table-7: Feature Size Comparison

Dataset	Features	BJOA	MBJOA
Sonar	60	34 (34)	34 (34.45)
Dermatology	34	21 (21.2)	21 (21.04)
Ionosphere	34	15 (15.4)	15 (15.2)
Musk	168	104 (103.85)	102 (101.8)
Pima	9	4 (4.2)	4 (4.4)
ParkinsonC	753	486 (485.95)	484 (484.4)
WDBC	30	17 (17.3)	17 (17.45)
Wine	13	6 (6.25)	6 (6.5)
Vehicle	18	10 (10.45)	11 (10.95)
SPECTF	43	25 (25)	25 (25.2)

5. CONCLUSION AND FUTURE SCOPE

The current study introduces a Modified Binary Jaya Optimization algorithm incorporating a Lévy-flight-based update mechanism and a Mutation Function to enhance the overall accuracy level compared to the Binary Jaya Optimization Algorithm (BJOA). Initially, a comparison is made between the BJOA and a standard K-NN classifier and its four variants, utilizing all available features. The analysis indicates that using all features results in slightly better accuracy than BJOA as a wrapper method. However, BJOA significantly reduces the feature size, leading to a reduction in training time. Subsequently, a comparison was made between Binary Jaya Optimization Algorithm (BJOA) and the Modified Binary Jaya Optimization Algorithm (MBJOA) across ten benchmark datasets. The results obtained indicate that MBJOA outperforms BJOA. In addition to the mean accuracy, MBJOA has exhibited superior performance compared to BJOA in other performance metrics, including Precision, Recall, and F1-score. The current study focuses on utilizing a wrapper method for feature selection tasks, specifically in the form of a binary optimization problem. However, this method can potentially be applied to other intricate continuous optimization problems. In addition to K-NN, MBJOA, and BJOA algorithms' performance exhibits slight variations. Therefore, there is potential for further optimization to achieve significantly improved results.

REFERENCES:

1. García S, Ramírez-Gallego S, Luengo, J, Benítez, JM, Herrera, F (2016) Big data preprocessing: methods and prospects. Big Data Analytics, 1(1), 1-22. https://doi.org/10.1186/s41044-016-0014-0.
2. Ayesha S, Hanif MK, Talib R (2020) Overview and comparative study of dimensionality reduction techniques for high dimensional data. Information Fusion, 59, 44-58. https://doi.org/10.1016/j.inffus.2020.01.005.
3. Siblini W, Kuntz P, Meyer F (2019) A review on dimensionality reduction for multi-label classification. IEEE Transactions on Knowledge and Data Engineering, 33(3), 839-857. https://doi.org/10.1109/TKDE.2019.2940014.
4. Ghosh M, Guha R, Sarkar R, Abraham A (2020) A wrapper-filter feature selection technique based on ant colony optimization. Neural Computing and Applications, 32, 7839-7857. https://doi.org/10.1007/s00521-019-04171-3.
5. Wold S, Esbensen K, Geladi P (1987) Principal component analysis. Chemometrics and intelligent laboratory systems, 2(1-3), 37-52. https://doi.org/10.1016/0169-7439(87)80084-9.
6. Balakrishnama S, Ganapathiraju A (1998) Linear discriminant analysis-a brief tutorial. Institute for Signal and information Processing.
7. Hyvärinen A, Oja E (2000) Independent component analysis: algorithms and applications. Neural networks, 13(4-5), 411-430. https://doi.org/10.1016/S0893-6080(00)00026-5.
8. Martis RJ, Acharya UR, Min LC (2013) ECG beat classification using PCA, LDA, ICA

and discrete wavelet transform. Biomedical Signal Processing and Control, 8(5), 437-448. https://doi.org/10.1016/j.bspc.2013.01.005.

9. Dash M, Liu H (1997) Feature selection for classification. Intelligent data analysis, 1(1-4), 131-156. https://doi.org/10.1016/S1088-467X(97)00008-5.
10. Liu H, Motoda H (2012). Feature selection for knowledge discovery and data mining. Springer Science & Business Media.
11. Kohavi R, John GH (1997) Wrappers for feature subset selection. Artificial intelligence, 97(1-2), 273-324. https://doi.org/10.1016/S0004-3702(97)00043-X.
12. Sadeghian Z, Akbari E, Nematzadeh H, Motameni H (2023) A review of feature selection methods based on meta-heuristic algorithms. Journal of Experimental & Theoretical Artificial Intelligence, 1-51. https://doi.org/10.1080/0952813X.2023.2183267.
13. Chuang LY, Chang HW, Tu CJ, Yang CH (2008) Improved binary PSO for feature selection using gene expression data. Computational Biology and Chemistry, 32(1), 29-38. https://doi.org/10.1016/j.compbiolchem.2007.09.005.
14. Oh IS, Lee JS, Moon BR (2004) Hybrid genetic algorithms for feature selection. IEEE Transactions on pattern analysis and machine intelligence, 26(11), 1424-1437. https://doi.org/10.1109/TPAMI.2004.105.
15. Kabir MM, Shahjahan M, Murase K (2012) A new hybrid ant colony optimization algorithm for feature selection. Expert Systems with Applications, 39(3), 3747-3763. https://doi.org/10.1016/j.eswa.2011.09.073.
16. Zorarpacı E, Özel SA (2016) A hybrid approach of differential evolution and artificial bee colony for feature selection. Expert Systems with Applications, 62, 91-103. https://doi.org/10.1016/j.eswa.2016.06.004.
17. Mafarja MM, Mirjalili S (2017) Hybrid whale optimization algorithm with simulated annealing for feature selection. Neurocomputing, 260, 302-312.https://doi.org/10.1016/j.neucom.2017.04.053.
18. Faris H, Mafarja MM, Heidari AA, Aljarah I, Ala'm AZ, Mirjalili S, Fujita H (2018) An efficient binary salp swarm algorithm with crossover scheme for feature selection problems. Knowledge-Based Systems, 154, 43-67. https://doi.org/10.1016/j.knosys.2018.05.009.
19. Dokeroglu T, Deniz A, Kiziloz HE (2022) A comprehensive survey on recent metaheuristics for feature selection. Neurocomputing. https://doi.org/10.1016/j.neucom.2022.04.083.
20. Rao R (2016) Jaya: A simple and new optimization algorithm for solving constrained and unconstrained optimization problems. International Journal of Industrial Engineering Computations, 7(1), 19-34. https://doi.org/10.5267/j.ijiec.2015.8.004.
21. Zitar RA, Al-Betar MA, Awadallah MA, Doush IA, Assaleh K (2022) An intensive and comprehensive overview of JAYA algorithm, its versions and applications. Archives of Computational Methods in Engineering, 29(2), 763-792. https://doi.org/10.1007/s11831-021-09585-8.
22. da Silva LSA, Lúcio YLS, Coelho LDS, Mariani VC, Rao RV (2022) A comprehensive review on Jaya optimization algorithm. Artificial Intelligence Review, 1-33. https://doi.org/10.1007/s10462-022-10234-0.
23. Frank A (2010) UCI machine learning repository [http://archive. ics. uci. edu/ml].
24. Kashef S, Nezamabadi□pour H, Nikpour B (2018) Multilabel feature selection: A comprehensive review and guiding experiments. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, 8(2), e1240. https://doi.org/10.1002/widm.1240.
25. Dong H, Li T, Ding R, Sun J (2018) A novel hybrid genetic algorithm with

granular information for feature selection and optimization. Applied Soft Computing, 65, 33-46. https://doi.org/10.1016/j.asoc.2017.12.048.

26. Gokulnath CB, Shantharajah SP (2019) An optimized feature selection based on genetic approach and support vector machine for heart disease. Cluster Computing, 22, 14777-14787. https://doi.org/10.1007/s10586-018-2416-4.
27. Abasabadi S, Nematzadeh H, Motameni H, Akbari E (2022) Hybrid feature selection based on SLI and genetic algorithm for microarray datasets. The Journal of Supercomputing, 78(18), 19725-19753. https://doi.org/10.1007/s11227-022-04650-w.
28. Jain I, Jain VK, Jain R (2018) Correlation feature selection based improved-binary particle swarm optimization for gene selection and cancer classification. Applied Soft Computing, 62, 203-215. https://doi.org/10.1016/j.asoc.2017.09.038.
29. Langeveld J, Engelbrecht AP (2012) Set-based particle swarm optimization applied to the multidimensional knapsack problem. Swarm Intelligence, 6, 297-342. https://doi.org/10.1007/s11721-012-0073-4.
30. Guha R, Ghosh M, Kapri S, Shaw S, Mutsuddi S, Bhateja V, Sarkar R (2021) Deluge based genetic algorithm for feature selection. Evolutionary intelligence, 14, 357-367. https://doi.org/10.1007/s12065-019-00218-5.
31. Joseph Manoj R, Anto Praveena MD, Vijayakumar K (2019) An ACO–ANN based feature selection algorithm for big data. Cluster Computing, 22, 3953-3960. https://doi.org/10.1007/s10586-018-2550-z.
32. Shunmugapriya P, Kanmani S (2017) A hybrid algorithm using ant and bee colony optimization for feature selection and classification (AC-ABC Hybrid). Swarm and evolutionary computation, 36, 27-36. https://doi.org/10.1016/j.swevo.2017.04.002.
33. Hancer E, Xue B, Zhang M, Karaboga D, Aka B (2018) Pareto front feature selection based on artificial bee colony optimization. Information Sciences, 422, 462-479. https://doi.org/10.1016/j.ins.2017.09.028.
34. Chaudhuri A, Sahu TP (2021) A hybrid feature selection method based on Binary Jaya algorithm for micro-array data classification. Computers & Electrical Engineering, 90, 106963. https://doi.org/10.1016/j.compeleceng.2020.106963.
35. Chaudhuri A, Sahu TP (2021) Binary Jaya algorithm based on binary similarity measure for feature selection. Journal of Ambient Intelligence and Humanized Computing, 1-18. https://doi.org/10.1007/s12652-021-03226-5.
36. Das H, Naik B, Behera HS (2022) A Jaya algorithm based wrapper method for optimal feature selection in supervised classification. Journal of King Saud University-Computer and Information Sciences, 34(6), 3851-3863. https://doi.org/10.1016/j.jksuci.2020.05.002.
37. Baliarsingh SK, Vipsita S, Dash B (2020) A new optimal gene selection approach for cancer classification using enhanced Jaya-based forest optimization algorithm. Neural Computing and Applications, 32, 8599-8616. https://doi.org/10.1007/s00521-019-04355-x.
38. Rao RV, Saroj A (2017) A self-adaptive multi-population based Jaya algorithm for engineering optimization. Swarm and Evolutionary computation, 37, 1-26. https://doi.org/10.1016/j.swevo.2017.04.008.
39. Basak S, Bhattacharyya B, Dey B (2022) Combined economic emission dispatch on dynamic systems using hybrid CSA-JAYA Algorithm. International journal of system Assurance Engineering and Management, 13(5),2269-2290. https://doi.org/10.1007/s13198-022-01635-z.

40. Han J, Moraga C (1995) The influence of the sigmoid function parameters on the speed of backpropagation learning. In From Natural to Artificial Neural Computation: International Workshop on Artificial Neural Networks Malaga-Torremolinos, Spain, June 7–9, 1995 Proceedings 3 (pp. 195-201). Springer Berlin Heidelberg. https://doi.org/10.1007/3-540-59497-3_175.
41. Sun G, Li J, Dai J, Song Z, Lang F (2018) Feature selection for IoT based on maximal information coefficient. Future Generation Computer Systems, 89, 606-616. https://doi.org/10.1016/j.future.2018.05.060.
42. Zeng Y, Yang Y, Zhao L (2009) Pseudo nearest neighbor rule for pattern classification. Expert Systems with Applications, 36(2), 3587-3595. https://doi.org/10.1016/j.eswa.2008.02.003.
43. Gou J, Ma H, Ou W, Zeng S, Rao Y, Yang H (2019) A generalized mean distance-based k-nearest neighbor classifier. Expert Systems with Applications, 115, 356-372. https://doi.org/10.1016/j.eswa.2018.08.021.
44. Gou J, Zhan Y, Rao Y, Shen X, Wang X, He W (2014) Improved pseudo nearest neighbor classification. Knowledge-Based Systems, 70, 361-375. https://doi.org/10.1016/j.knosys.2014.07.020.
45. Pan Z, Wang Y, Ku W (2017) A new k-harmonic nearest neighbor classifier based on the multi-local means. Expert Systems with Applications, 67, 115-125. https://doi.org/10.1016/j.eswa.2016.09.031.