https://doi.org/10.48048/tis.2022.4588

A Nascent Technique for Cultivated Feature Selection using Evolutionary Computation Algorithms

Sachin Minocha[*] and Birmohan Singh

Department of Computer Science and Engineering, Sant Longowal Institute of Engineering and Technology, Punjab, India

([*]Corresponding author's e-mail: sachin0111@gmail.com)

Received: 26 December 2020, Revised: 15 May 2021, Accepted: 22 May 2021

Abstract

Evolutionary computation algorithms are in recent trends for feature selection due to their efficient results as compared to traditional algorithms. The performance of the evolutionary computational algorithm completely depends upon the parameter (like population, number of generations) setting. Existing parameter setting techniques have been developed for specific algorithms. This paper designs a generic pre-processing technique that removes the redundant as well as irrelevant features from the initial given feature set. This work uses Kendall tau to find the non-redundant feature and the Kruskal Wallis for the relevant features. If Kendall tau is not able to select any non-redundant feature then mutual information gain is used to select significant features followed by the Kruskal Wallis to select relevant features. The designed technique has been analyzed on 4 evolutionary computation algorithms, Modified Binary Particle Swarm Optimization, Non-dominated Sorting Genetic Algorithm, Binary Grey Wolf Optimization, and Binary Whale Optimization Algorithm over 6 datasets by using accuracy, the number of features in selected subset, sensitivity, and specificity. The present technique improves the performance of each algorithm in terms of accuracy, sensitivity, and specificity by 1 % on average with a 10 % reduction in the selected feature subset proves the significance of the proposed technique.

Keywords: Feature selection, Evolutionary computation, Metaheuristic algorithm, Binary whale optimization algorithm, Preprocessing

Introduction

In classification for a given dataset D with a set of vectors (instances) along with class labels C (C_1, C_2, …. C_{noc}), we have to identify the correct class for each instance. Each instance consists of different features F (f_1, f_2, ….. f_{nof}), describing any specific event. The preliminary task of the classifier is to predict the class label of different instances accurately [1]. The performance of the classifier is directly related to the number and behavior of the feature. The performance optimization of any classifier can be done by selecting relevant and non-redundant features [2]. A feature is said to be relevant if and only if the feature appears in every Boolean function used to determine the class label. In other words, a feature is relevant if $P(C|f_i,F) \neq P(C|F)$ where $F = \{f - f_i\}$ and $P(C|F)$ is the probability distribution of different classes with features F [3]. A feature is said to be redundant if it is highly correlated with the other feature(s) of a dataset. Moreover, the redundancy is 0 (minimum) when 2 features are orthogonal and it increases with a decrease in orthogonality [4].

Feature selection is an active field with numerous applications in areas like data mining, pattern recognition, machine learning, and statistics. It has proven its significance by enhancing accuracy and reducing complexity [5]. Suppose S is a subset of selected features from F features in dataset D. Then the feature selection can be formalized as Eq. (1);

$$P(C|S) \geq P(C|P_s - S) \quad (1)$$

where P_s is the power set of F features. Eq. (1) shows that the classification performance by using subset S of feature is higher as compared to the classification performance by other sets of features in P_s. The feature selection is a process to enhance the classification performance by selecting a subset of features from all the possible combinations of features [6]. The feature selection problem is proven to be NP-

complete as the number of possible subsets increases exhaustively with an increase in the dimensionality of the dataset [7].

Evolutionary Computation (EC) algorithms have the capabilities to perform the global search efficiently. The ability of EC algorithms to convergence towards global optima makes it suitable for feature selection. A few EC algorithms used for the feature selection are genetic algorithm (GA) [8], particle swarm optimization (PSO) [9], differential evolution (DE) [10]. These are the most popular EC algorithms as more than 500 papers had been published on these in the past few years [11]. Non-dominated sorting genetic algorithm (NSGA) is a popular multi-objective EC algorithm that is a modification of genetic algorithm. This algorithm builds the parent population for the next generation by sorting all parents and offspring into a different level of non-dominated solutions [12]. This algorithm also preserves diversity based on the crowding distance [13]. Modified binary particle swarm optimization (MBPSO) is designed to deal with the premature convergence of binary particle swarm optimization. It manages the particle inconsistency using the velocity and the resemblance between best particle solutions [14]. Differential evolution using a wheel-based search strategy ($DEFS_w$) modifies the DE to reduce the search space. The $DEFS_w$ reduces the search space by defining the upper limit to the feature subset size [15]. Binary grey wolf optimization (BGWO) simulates the hunting process of grey wolves [16]. The binary whale algorithm for optimization (BWAO) uses the hunting behavior of the whale to select the relevant features [17]. The performances of various EC algorithms depend upon different parameter settings; the wrong selection of such parameters for any EC algorithm can result in a non-optimal solution. The vital parameters that need to be fixed are the size of the population, the number of generations or iterations, the number of independent runs of an EC algorithm. The population size is a crucial factor for the performance determination of an EC algorithm for feature selection. If the initial population of an EC algorithm includes the optimal subset of features, then the EC algorithm performs well otherwise the performance of the EC algorithm depends upon other parameters [18].

Different authors have worked to control the initial size of the population to select the optimal subset of features. The work of few such authors has been discussed here. Hussien and Amin [19] has designed an Improved Harris Hawks Optimization (IHHO) by improving the Harris Hawks Optimization (HHO) algorithm through opposition-based learning, chaotic local search algorithm, and the self-adaptive strategy. Ooi *et al.* [20] has proposed the Self-Tune Linear Adaptive Genetic Algorithm (STLA-GA) that adapts the parameters like maximum generation, mutation probability, convergence threshold, and population size of genetic algorithm. Lobo and Lima [21]; Bielza *et al.* [22]; Toğan and Daloğlu [23] have given different schemes applied to the genetic algorithm to control the population parameter. Different preprocessing techniques have been used by Yu *et al.* [24]; Brest and Sepesy [25] for population size reduction on DE. Teng *et al.* [26] has used the correlation matrics along with V-shape transfer function-based binary particle swarm optimization for feature subset selection. Viharos *et al.* [27] have proposed an adaptive feature selection algorithm that selects the best possible feature based on the given problem. This algorithm iterates over several algorithms for a particular dataset to select the best features. Beuren and Anzanello [28] removed the redundant features using mutual information and rank the variables by using non-parametric tests to select the optimal subset of features. Any of the stated algorithms do not apply to different EC algorithms; each algorithm is specific to the particular EC algorithm.

This paper develops a generic technique that can be applied to any EC algorithm as pre-processing for controlling the population parameter. The designed technique reduces the initial population size by removing the redundant features which don't contribute to the group of selected features. This leads to enhanced probability for the selection of optimal subset. The proposed technique doesn't remove the relevant features that result in the improved accuracy of the classification algorithm. The main contribution of this work is as follows.

1) This work removes the redundant features from the initial population using the Kendall tau and mutual information gain to reduce the initial population for any EC algorithm.

2) The relevant features that are highly correlated with class are added back to the initial population to maintain the optimal initial population.

3) The performance validation has been done using 4 EC computation algorithms to show the significance of the work.

Materials and methods

This section describes the proposed preprocessing method along with the various useful techniques including Kendall tau, Kruskal Wallis, and theorems to find redundant features.

Similarity of distribution

The similarity of distribution between 2 variables is determined by the similarity score between 2 features evaluated using Kendall tau or Kruskal Wallis. The Kendall tau and Kruskal Wallis Kruskal Wallis are explained in the next subsections.

Kendall tau

Kendall's tau is a rank correlation-based measure used to estimate the association between 2 variables. It is a robust measure of association due to its vigorousness to the nature of distribution and insensitivity to the outliers. Suppose, 2 random variables X_i and X_j then Kendall tau for 2 observations (no ties) is given by Eq. (2);

$$\tau = pr\left(X_{i1} > X_{j1}\ and\ X_{i2} > X_{j2}\right) - pr(X_{i1} > X_{j1}\ and\ X_{i2} < X_{j2}) \quad (2)$$

where pr is the probability. It is preferred to make this definition in terms of the dominance variable given in Eq. (3);

$$\tau = \frac{score}{max\_score} \quad (3)$$

where the score is estimated by enumerating the concordance between 2 variables in a regular fashion. The score can have a value of 1 or −1 depending upon the same order or opposite order of 2 variables respectively. It is given by Eq. (4);

$$score = sign\left(X_{j1} - X_{i1}\right) \times sign\left(X_{j2} - X_{i2}\right) \quad (4)$$

The max_score is evaluated as shown in Eq. (5);

$$\max\_score = {n \times (n-1)}/{2} \quad (5)$$

where n is the number of observations [29-31].

Kruskal Wallis test

It is a non-parametric test used to evaluate whether 2 samples have been generated from the same population. It tests the null hypothesis that each sample belongs to the same population $H_o: \overline{S_e} = \frac{(n+1)}{2}$. In this test, firstly the output is altered to rank in ascending order. The test statistics of the samples with ties for the mean rank can be calculated by Eq. (6);

$$\widehat{H} = \frac{12}{n(n+1)} \sum_{e=1}^{C} \frac{{S_e}^2}{N_e} - 3(n+1) \quad (6)$$

where, $n = \sum_{e=1}^{C} n_e$ is the number of samples within the population, n_e is the number of data elements in any particular (e^{th}) group/class, S_e is the rank sum of the particular group. The test statistics in the presence of equal values (ties) as given by Eq. (7);

$$\widehat{H} = \frac{1}{P}\left(\sum_{e=1}^{C} \frac{{S_e}^2}{n_e} - \frac{n(n+1)^2}{4}\right) \quad (7)$$

where P is given by Eq. (8);

$$P = \frac{1}{n-1}\left(\sum S^2 - \frac{n(n+1)^2}{4}\right) \quad (8)$$

where S is the mean rank of each element in *D*. The null hypothesis is rejected if $\widehat{H} > \tau$, here τ is the tabular value or the experimental threshold. Some special table can be found to test the significance of the $\widehat{H}$, but the χ^2 distribution is more practical [32,33].

Redundant features from dataset

This section proves that the features with high similarity distribution are redundant. Moreover, this section also determines the redundant features of a dataset.

Theorem 1: If x, y are different variables and z is a decision variable such that z = f(x, y). If x and y have high similar distribution then z = F(x).

Solution:

$$z = f(x, y) \tag{9}$$

The similarity between the 2 variables is given as [34];

$$S(x, y) = \frac{1}{2}\left[1 + \sum_i (x_i . y_i)^{1/2}\right] \tag{10}$$

where $\sum_i (x_i . y_i)^{1/2}$ is the Bhattacharyya coefficient, which increases with the increase in distribution similarity.

Using Eq. (10) x and y are highly similar so;

$$y = h(x) \tag{11}$$

Using Eqs. (9) and (11);

$$z = f(x, h(x)) \tag{12}$$

⇨ z = F(x). This proves the given statement.

Theorem 2: If 2 different variables or features have a high similarity distribution, then any of these features can be used to provide the class label without loss of generality.

Solution:

Suppose f1 and f2 are 2 features with c as a class label. Then c can be determined using f1 and f2 as stated by Eq. (13);

$$c = h(f1, f2) \tag{13}$$

where 'h' is any function to compute the relation between c and f1,f2. In 2010, Auffarth *et al.* [35] has given Eq. (14) to find redundancy;

$$Red(f1, f2, c) = \frac{1}{n}\sum_{i=1}^{n} \Delta([f1|c = c_i], [f2|c = c_i]) \tag{14}$$

where, $[f1|c = c_i]$ shows the distribution of feature f1 when the class is i, n is the number of the class label, and Δ shows the similarity distribution.

Using Eq. (14), for high values of Δ the *Red* is high then;

$$f1 = p(f2) \tag{15}$$

where *p* is any function.

Using Theorem 1 with Eqs. (13) and (15) gives Eq. (16);

$$c = P(f1) \tag{16}$$

This shows that class c can be computed by using any feature that proves the theorem.

Proposed work

A generic pre-processing technique is designed by selecting the features which are highly correlated with the class and mutually exclusive to each other. Suppose a dataset *D* has *nof* features $F = (f_1, f_2, \ldots \ldots f_{nof})$ and *noc* classes $C = (c_1, c_2, c_3, \ldots c_{noc})$ with *n* instances of each. Then the process to select the *sf* subset of features applies Kendall tau to determine the association of each attribute with the other attribute. This association is determined in terms of tau(τ) which can have any integer value depending upon score and max_score as specified in previous subsections is given by Eq. (17);

$$\forall_{i=1}^{nof} \forall_{j=1}^{nof} \tau_{ij} = \frac{\sum_{k=1}^{n-1} \text{sign}(f_{jk} - f_{ik}) \times \text{sign}(f_{jk+1} - f_{ik+1})}{n \times (n-1)/2} \tag{17}$$

The minimum and the maximum value of τ for each feature are computed by Eqs. (18) - (19), respectively.

$$\forall_{i=1}^{nof} \min_i = minimum\left(\forall_{j=1}^{nof} \tau_{ij}\right) \tag{18}$$

$$\forall_{i=1}^{nof} \max_i = maximum\left(\forall_{j=1}^{nof} \tau_{ij}\right) \tag{19}$$

where min_i, max_i gives the minimum and maximum value of τ for ith feature respectively. τ_{ij} shows the association (τ value) of the ith feature with the jth feature. To compare the value of τ for each attribute, this value is normalized between 0 and 1 by using Eq. (20);

$$\forall_{i=1}^{nof} \forall_{j=1}^{nof} \tau_{ij} = \frac{(\tau_{ij} - \min_i)}{(\max_i - \min_i)} \tag{20}$$

A feature with a similarity score higher than the average similarity score is redundant so the average similarity score is calculated to determine the redundant features by Eqs. (21) - (22);

$$\forall_{i=1}^{nof} \tau_{mi} = \frac{1}{nof} \sum_{j=1}^{nof} \tau_{ij} \tag{21}$$

$$\tau_M = \frac{1}{nof} \sum_{i=1}^{nof} \tau_{mi} \tag{22}$$

where Eq. (21) computes the average similarity score for each feature τ_{mi} by dividing the sum of the score for that particular feature by the number of features. While Eq. (12) average the score of each feature to compute the overall similarity score τ_M.

Using Theorem 2, the features with high similarity i.e. $\tau_M < \tau_{mi}$ are redundant so these features can be dropped to select the non-redundant subset of features given by Eq. (23);

$$sf = \forall_{i=1}^{nof} \bigcup_{\tau_{mi} < \tau_M} f_i \tag{23}$$

Eq. (23) union the features with a similarity score less than the overall similarity score to give a non-redundant subset of features.

If there exists no feature whose similarity score is less than the τ_M then the feature with high information gain as compared to average information gain is selected to form a subset of features. The information gain for a particular feature can be computed by (24);

$$\forall_{i=1}^{nof} infogain_i = H(C) - H(C|f_i) \tag{24}$$

where, $H(C)$ and $H(C|f_i)$ is the entropy and entropy when f_i is included given by Eqs. (25) - (26), respectively.

$$H(C) = -\sum_{c \in C} p(c) \log \mathrm{p(c)} \tag{25}$$

$$\forall_{i=1}^{nof} H(C|f_i) = -\sum_{c \in C} p(c|f_i) \log \mathrm{p}\big((\mathrm{c})\big|\mathrm{f_i}\big) \tag{26}$$

where $p(c)$ is the probability for class c. The average information gain, say $infogain_m$ is calculated by averaging the information gain by each feature as given by Eq. (27);

$$infogain_m = \frac{1}{nof} \sum_{i=1}^{nof} infogain_i \tag{27}$$

Proposed Algorithm (D,F,C)

Initiate $sf = \{\emptyset\}$,th = 0.4
Calculate similarity score for each feature i.e. $\forall_{i=1}^{nof} \tau_{mi}$ by using Eq. (21)
Compute the overall average similarity score τ_M by using Eq. (22)
Add Non-redundant Features to sf by using Eq. (23)
if length(sf)==0
    Compute information gain for each feature i.e. $\forall_{i=1}^{nof} infogain_i$ using Eq. (24)
    Compute average information gain i.e. $infogain_m$ by using Eq. (27)
    Add features to sf by using Eq. (28)
Endif
Compute mean feature rank $\widehat{H}$ using Eq. (6)
Add relevant features to the sf using Eq. (29)

Figure 1 Proposed technique for preprocessing.

The subset of features having information gain higher than average information gain is computed using Eq. (28);

$$sf = \forall_{i=1}^{nof} \cup_{infogain_{i > infogain_m}} f_i \tag{28}$$

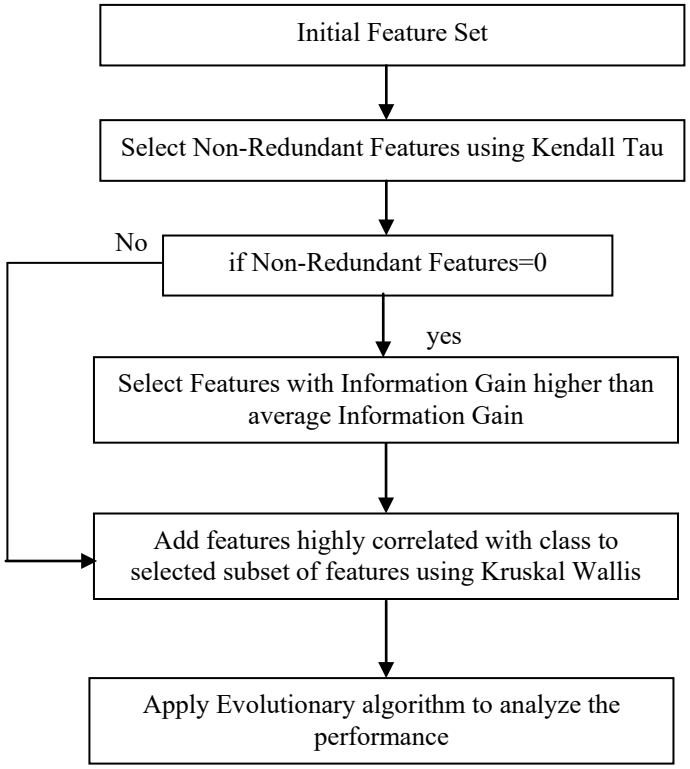

Figure 2 Block diagram for proposed technique.

The non-redundant features sf may remove some features which are highly correlated with the class, such features should be added back as these features make a significant contribution to the classification. That's why the Kruskal-Wallis test is used to determine whether any attribute removed has the same distribution as the distribution of the class, such features are added to the selected subset of features. The mean rank $\widehat{H}$ of each feature is computed by Eq. (6) if there is no tie otherwise Eq. (7) gives the mean rank of a feature. The relevant features are added to sf using Eq. (29) to give the selected subset of features.

$$sf = \forall_{f_i \in \{F-sf\}} \cup_{\widehat{H}<th}(sf, f_i) \tag{29}$$

where 'th' is the threshold value to select the relevant feature.

This subset produces non-redundant features that are highly correlated with the class. The process is also mentioned in the algorithm using **Figure 1**. The algorithm given in **Figure 1**, selects the mutual exclusive features which are highly correlated with the class. The block diagram corresponding to the technique is given in **Figure 2**.

Figure 2, summarizes the working of the proposed technique. The initial feature set consists of redundant as well as irrelevant features. Such features may lower the performance of the feature selection algorithms. This technique selects the non-redundant features from the initial feature set by using the Kendall tau as stated by Eq. (23). If the number of non-redundant features is 0, then the features having information gain higher than the average information gain are selected by using Eq. (28) as non-redundant features. Some redundant features may be highly correlated with the class, such features are added to the selected subset of features by using the Kruskal Wallis as stated in Eq. (29). These steps remove the redundant and irrelevant features from the initial feature set. The processed dataset is used as an input to an evolutionary computation-based feature selection algorithm. The implementation and the result analysis of the proposed technique have been given in the next section.

Results and discussion

This work has been analyzed on 6 datasets given in **Table 1**, extracted from the UCI repository [36]. The datasets selected for the analysis vary in terms of classes, instances as well as attributes as shown in **Table 1**. The number of instances varies from 208 to 10,299 while the number of attributes varies from 24 to 561. Such high variation in the size of datasets checks the scalability of the proposed work. Moreover, each dataset belongs to different application areas like financial, biology, and artificial, such diverse datasets are capable enough to explore the capabilities of the algorithm.

Table 1 Datasets used.

S. No.	Dataset name	Class	Instance	Attributes
1	SONAR	2	208	60
2	WDBC	2	569	30
3	ARRHYTHMIA	13	279	452
4	GERMAN	2	1,000	24
5	MADELON	2	2,000	500
6	HAR	6	10,299	561

The proposed preprocessing technique has been implemented on the 4 evolutionary algorithms, namely NSGA, MBPSO, BGWO and BWOA. These algorithms already have been discussed in the literature. The performance of these algorithms has been analyzed by using parameters; accuracy, average selection size (NOF), sensitivity, and specificity. The detail of these parameters is specified by Emary *et al.* [16]. This paper compares the performance of NSGA, MBPSO, BGWO, and BWOA algorithms with the NSGA-P, MBPSO-P, BGWO-P, and BWOA-P with the same parameter settings for each pair respectively. Here the NSGA-P represents the NSGA with the preprocessing technique to reduce the initial feature set. Similarly, in the MBPSO-P, BGWO-P, and BWOA-P preprocessing has been used before the MBPSO, BGWO, and BWOA respectively. Each algorithm is executed 10 times to take an average of the results. The optimal parameter setting has been done based on literature to get the optimized performance. The number of iterations in each algorithm is equal to 100. The number of search agents in the BWOA and BGWO algorithm is equal to the number of instances in the dataset, and the problem dimension is the number of features in the dataset. MBPSO algorithm works with lower bound as zero and upper bound as one with the constants c1 and c2 as 1.4962 and velocity range [−0.1 0.1]. The NSGA algorithm 0.1 mutation rate while 0.7 and 0.4 as a percentage of crossover and mutation respectively. The NSGA, MBPSO, and BGWO with 60 %, 40 % as the training and testing dataset respectively uses SVM classifier with RBF kernel ($\sigma = 15$). While BWOA uses KNN as a classifier with 10 fold cross-validation. The performance comparison of NSGA, MBPSO, BGWO, and BWOA with NSGA-P, MBPSO-P, BGWO-P, and BWOA-P, respectively is shown in the following **Table 2** in terms of Accuracy and average selection size (NOF).

Table 2 Comparison of accuracy and average selected size of features.

Dataset	BGWO		BGWO-P		NSGA		NSGA-P	
	Accuracy	NOF	Accuracy	NOF	Accuracy	NOF	Accuracy	NOF
SONAR	75.0000	31.3	76.3333	14.4	68.9285	20.4	75.3571	9.9
WDBC	95.6578	8.1	95.8684	4.4	96.7543	11.3	96.3157	8.8
ARRHYTHMIA	56.6850	188	58.2320	143.1	57.6243	112.2	58.2872	88

GERMAN	67.2500	12.5	67.1000	12.5	67.8500	9.9	68.2500	9.6
MADELON	58.4642	255.2	60.5500	73.3	57.3000	215.1	58.6500	222.3
HAR	92.6700	281.1	93.8700	277.8	89.8543	250.1	89.7500	170.6
Dataset	**MBPSO**		**MBPSO-P**		**BWOA**		**BWOA-P**	
	Accuracy	**NOF**	**Accuracy**	**NOF**	**Accuracy**	**NOF**	**Accuracy**	**NOF**
SONAR	71.7857	30.8	70.0000	22	81.683	50.2	82.019	33.3
WDBC	96.2719	17.6	97.0526	15.4	93.515	26.1	93.708	21.4
ARRHYTHMIA	55.8622	137.9	57.0165	114	85.504	221.9	86.050	158.1
GERMAN	67.2250	17.2	68.9375	14.6	69.620	21.4	70.050	20.9
MADELON	55.5625	252.5	55.5875	257	50.623	468.9	74.431	395.4
HAR	88.2437	280.3	90.0341	278.8	99.905	476.5	99.910	310.2

Table 2 shows the comparison of the accuracy and average selection size for each algorithm. It can be identified that the accuracy of BGWO-P and BWOA-P in each dataset has been increased with the decrease in the average selection size as compared to BGWO and BWOA respectively. It means less number of selected features is producing a higher accuracy on the same number of instances. This is due to the removal of redundant features by applying preprocessing on the BGWO and BWOA. While the accuracy of NSGA-P in each dataset has been increased with the decrease in the average selection size except for the WDBC dataset. In the WDBC dataset, the accuracy has been decreased minor with the reduced average selection size. This is due to the proposed preprocessing algorithm enhances the performance of the NSGA algorithm by selecting significant features. The accuracy of MBPSO-P in each dataset has been increased except in the SONAR dataset with the decrease in the average selection size. The SONAR dataset exhibit exception by having lower accuracy with reduced features. It is because of the proposed technique that enhances the performance of the MBPSO algorithm by selecting significant and non-redundant features. This comparison also has been shown graphically in **Figures 3** and **4**.

Figure 3 compares the performance of each algorithm with its modified version respectively in terms of accuracy. The increase in the accuracy over each dataset can be identified with an exception on the WDBC dataset in NSGA and GERMAN dataset in the BGWO algorithm. The improvement in the accuracy is due to a selection of relevant and non-redundant features in the initial dataset by the proposed algorithm.

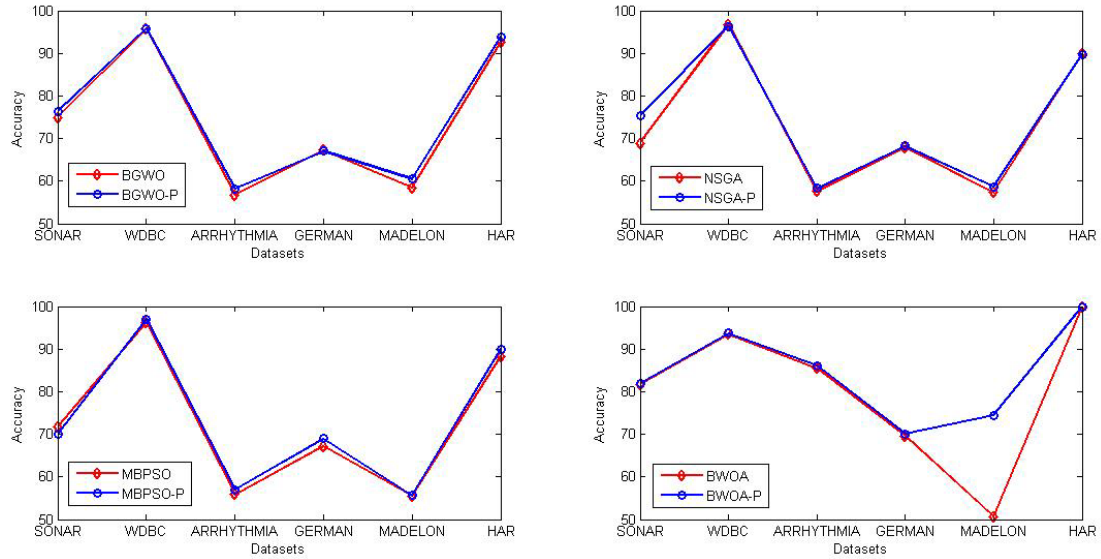

Figure 3 Comparison of accuracy.

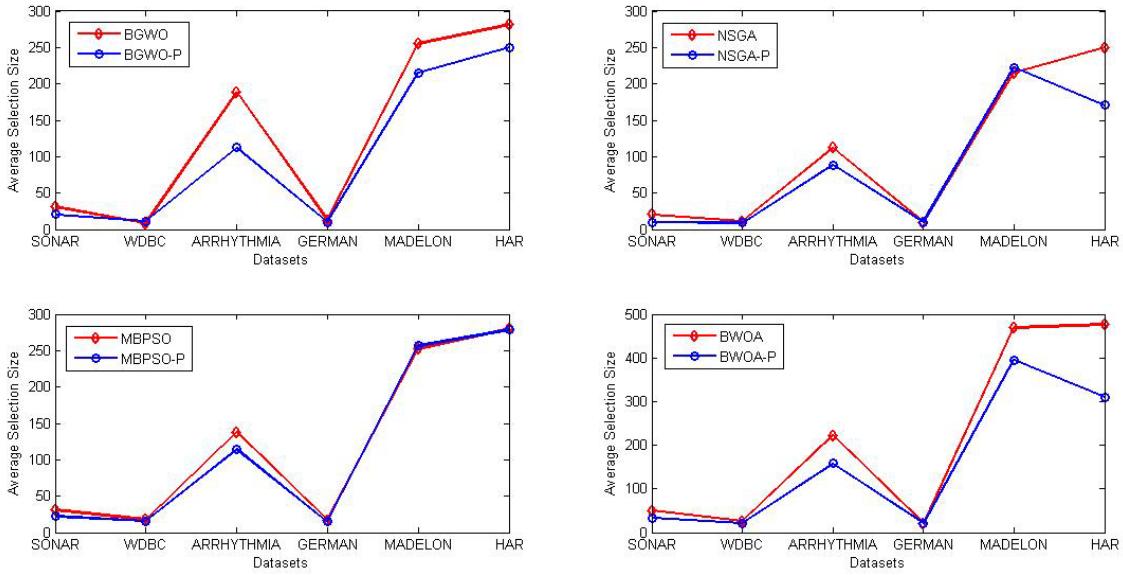

Figure 4 Comparison of average selection size of features.

Figure 4 compares the average selection size of features (NOF) for each algorithm. NOF has been decreased in each dataset over each algorithm due to the reduction in the initial population as per the proposed algorithm.

Table 3 Specificity and Sensitivity analysis for BGWO & BGWO-P.

Dataset	BGWO		BGWO-P		NSGA		NSGA-P	
	Sensitivity	**Specificity**	**Sensitivity**	**Specificity**	**Sensitivity**	**Specificity**	**Sensitivity**	**Specificity**
SONAR	0.6657	0.8102	0.6885	0.8367	0.7685	0.6326	0.7900	0.7775
WDBC	0.9654	0.9537	0.9690	0.9585	0.9527	0.9722	0.9545	0.9658
ARRHYTHMIA	0.8200	0.4109	0.8466	0.4252	0.8488	0.4428	0.8511	0.4857
GERMAN	0.7560	0.6353	0.7585	0.6321	0.6731	0.6808	0.7439	0.6852
MADELON	0.6090	0.5603	0.6278	0.5832	0.5869	0.5591	0.5990	0.5639
HAR	0.9754	0.9612	0.9798	0.9702	0.9846	0.9800	0.9839	0.9753
Dataset	**MBSO**		**MBPSO-P**		**BWOA**		**BWOA-P**	
SONAR	0.6771	0.7469	0.6514	0.7346	0.8603	0.7670	0.8637	0.7783
WDBC	0.9527	0.9658	0.9572	0.9747	0.8830	0.9661	0.8840	0.9689
ARRHYTHMIA	0.8231	0.4114	0.8444	0.4198	0.9608	0.2443	0.9651	0.2496
GERMAN	0.7154	0.6530	0.7306	0.6710	0.3003	0.8658	0.3160	0.8668
MADELON	0.5789	0.5324	0.5638	0.5400	0.5615	0.4509	0.8019	0.6866
HAR	0.6132	0.5864	0.6432	0.5931	0.9991	0.9989	0.9991	0.9989

Table 3 compares the sensitivity and specificity of each algorithm. The BGWO-P and BWOA-P exhibit higher specificity on each dataset due removal of noisy features. The enhanced specificity is also observed except in the GERMAN dataset due to the selection of mutually exclusive features. While the NSGA-P shows the higher specificity and sensitivity on each dataset except the HAR dataset because of less noisy features and mutually exclusive features. The MBPSO-P shows the higher specificity and sensitivity on each dataset except the SONAR dataset because of less noisy features and mutually exclusive features.

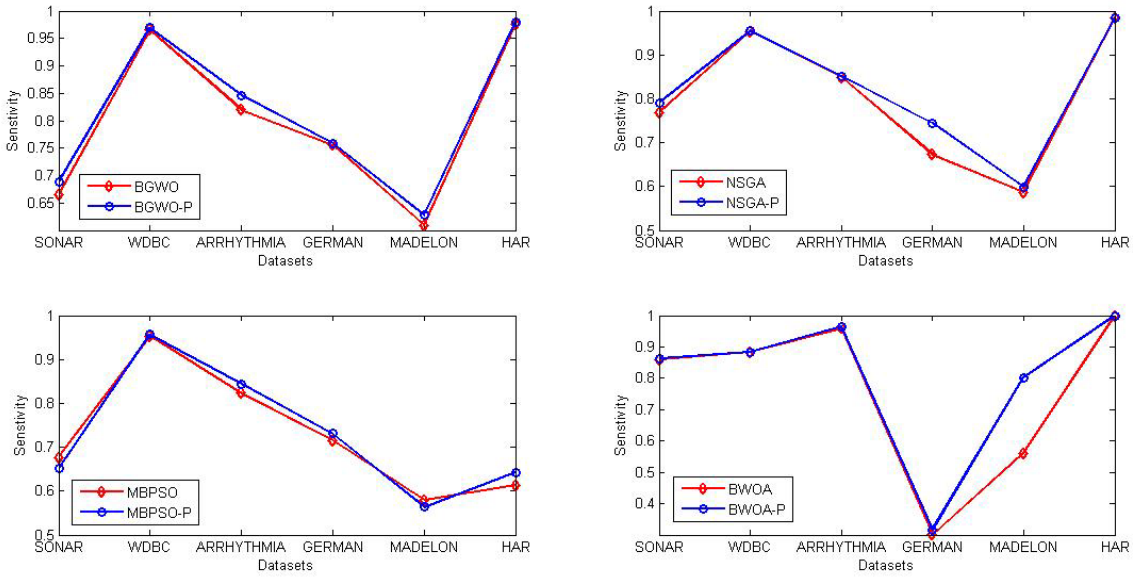

Figure 5 A comparison of sensitivity.

Figure 5 shows the comparison of sensitivity for BGWO, NSGA, MBPSO, and BWOA with BGWO-P, NSGA-P, MBPSO-P, and BWOA-P respectively. The higher sensitivity in each algorithm over each dataset proves that the proposed technique improves the performance of the evolutionary algorithm by selecting mutually exclusive features.

Figure 6 compares the specificity of BGWO, NSGA, MBPSO, and BWOA with BGWO-P, NSGA-P, MBPSO-P, and BWOA-P respectively. The higher specificity in each algorithm over each dataset proves that the proposed technique enhances the performance of the evolutionary algorithm by selecting highly correlated features with the class.

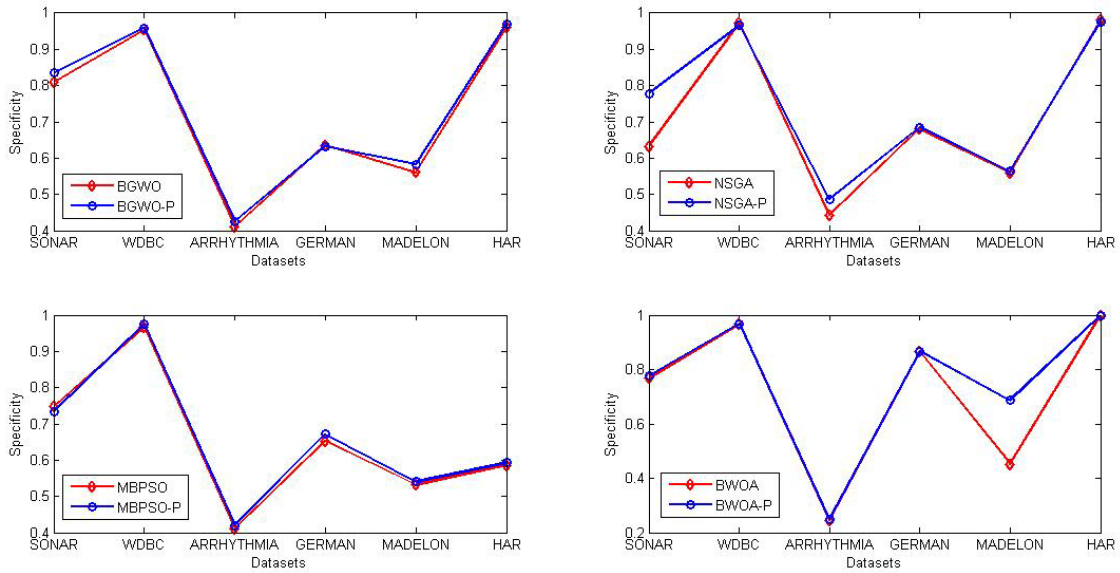

Figure 6 Comparison of specificity

Conclusions

This paper designs a generic preprocessing technique that can be applied to evolutionary computation algorithms to improve their performance. The presented technique selects the mutual exclusive features which are highly correlated with the class. The mutually exclusive features have been selected by using the Kendall tau. If Kendall tau isn't able to select any feature, then information gain is used to select the significant feature. Then, the features which are highly correlated with the class are selected by using the Kruskal-Wallis test. This leads to the removal of the noisy features from the dataset that improves the performance of the evolutionary algorithm in terms of classification accuracy with a reduced subset of features. The performance analysis has been done with 4 evolutionary algorithms MBPSO, NSGA, BGWO, and BWOA over 6 datasets of varying size. The improvement in accuracy with reduced features and high sensitivity and specificity proves the significance of the algorithm. In the future, the algorithm can be implied to other application areas like medical, social to select appropriate features.

References

[1] S Gu, R Cheng and Y Jin. Feature selection for high-dimensional classification using a competitive swarm optimizer. *Soft Comput.* 2018; **22**, 811-22.

[2] H Chantar, M Mafarja, H Alsawalqah, AA Heidari, I Aljarah and H Faris. Feature selection using binary grey wolf optimizer with elite-based crossover for Arabic text classification. *Neural Comput. Appl.* 2020; **32**, 12201-20.

[3] L Yu and H Liu. Efficient feature selection via analysis of relevance and redundancy. *J. Mach. Learn. Res.* 2004; **5**, 1205-24.

[4] D Luo, F Wang, J Sun, M Markatou, J Hu and S Ebadollahi. SOR: Scalable orthogonal regression for non-redundant feature selection and its healthcare applications. *In*: Proceedings of the SIAM

International Conference on Data Mining, Anaheim, California, USA. 2012, p. 576-87.
[5] M Mafarja, I Aljarah, H Faris, AI Hammouri, AM Al-Zoubi and S Mirjalili. Binary grasshopper optimisation algorithm approaches for feature selection problems. *Expert Syst. Appl.* 2019; **117**, 267-86.
[6] X Gu, J Guo, L Xiao, T Ming and C Li. A feature selection algorithm based on equal interval division and minimal-redundancy - maximal-relevance. *Neural Process. Lett.* 2020; **51**, 1237-63.
[7] G Ditzler, R Polikar and G Rosen. A sequential learning approach for scaling up filter-based feature subset selection. *IEEE Trans. Neural Networks Learn.* 2018; **29**, 2530-44.
[8] DE Goldberg and JH Holland. Genetic algorithms and machine learning. *Mach. Learn.* 1988; **3**, 95-9.
[9] R Eberhart and J Kennedy. A new optimizer using particle swarm theory. *In*: Proceedings of the MHS'95. Proceedings of the 6th International Symposium on Micro Machine and Human Science, Nagoya, Japan. 1995, p. 39-43.
[10] R Storn and K Price. Differential evolution - a simple and efficient heuristic for global optimization over continuous spaces. *J. Glob. Optim.* 1997; **11**, 341-59.
[11] B Xue, M Zhang, WN Browne and X Yao. A survey on evolutionary computation approaches to feature selection. *IEEE Trans. Evol. Comput.* 2016; **20**, 606-26.
[12] Y Yuan, H Xu and B Wang. An improved NSGA-III procedure for evolutionary many-objective optimization. *In*: Proceedings of the Annual Conference on Genetic and Evolutionary Computation, Vancouver BC, Canada. 2014, p. 661-8.
[13] H Bach, N Bing, X Ivy, L Peter and A Mengjie. New mechanism for archive maintenance in PSO-based multi-objective feature selection. *Soft Comput.* 2016; **20**, 3927-46.
[14] SM Vieira, LF Mendonça, GJ Farinha and JMC Sousa. Modified binary PSO for feature selection using SVM applied to mortality prediction of septic patients. *Appl. Soft Comput. J.* 2013; **13**, 3494-504.
[15] A Al-Ani, A Alsukker and RN Khushaba. Feature subset selection using differential evolution and a wheel based search strategy. *Swarm Evol. Comput.* 2013; **9**, 15-26.
[16] E Emary, HM Zawbaa and AE Hassanien. Binary grey wolf optimization approaches for feature selection. *Neurocomputing* 2016; **172**, 371-81.
[17] AG Hussien, D Oliva, EH Houssein, AA Juan and X Yu. Binary whale optimization algorithm for dimensionality reduction. *Mathematics* 2020; **8**, 1821.
[18] FJ Lobo, CF Lima and Z Michalewicz. *Parameter setting in evolutionary algorithms*. Springer, Heidelberg, Germany, 2007.
[19] AG Hussien and M Amin. A self-adaptive Harris Hawks optimization algorithm with opposition-based learning and chaotic local search strategy for global optimization and feature selection. *Int. J. Mach. Learn. Cybern.* 2022; **13**, 309-36.
[20] CS Ooi, MH Lim and MS Leong. Self-tune linear adaptive-genetic algorithm for feature selection. *IEEE Access*. 2019; **7**, 138211-32.
[21] FG Lobo and CF Lima. A review of adaptive population sizing schemes in genetic algorithms. *In*: Proceedings of the 7th Annual Workshop on Genetic and Evolutionary Computation, Washington DC, USA. 2007, p. 185-204.
[22] C Bielza, P Juan and P Larra. Parameter control of genetic algorithms by learning and simulation of Bayesian networks - a case study for the optimal ordering of tables. *J. Comput. Sci. Tech.* 2013; **28**, 720-31.
[23] V Toğan and AT Daloğlu. An improved genetic algorithm with initial population strategy and self-adaptive member grouping. *Comput. Struct.* 2008; **86**, 1204-18.
[24] T Yu, K Sastry and DE Goldberg. *Population sizing to go: Online adaptation using noise and substructural measurements. In*: FG Lobo, CF Lima, Z Michalewicz (Eds.). Parameter setting in evolutionary algorithms. Springer, Heidelberg, Germany, 2007, p. 205-23.
[25] J Brest and M Sepesy. Population size reduction for the differential evolution algorithm. *Appl. Intell.* 2008; **29**, 228-47.
[26] X Teng, H Dong and X Zhou. Adaptive feature selection using v-shaped binary particle swarm optimization. *PLoS One* 2017; **12**, e0173907.
[27] ZJ Viharos, KB Kis, Á Fodor and MI Büki. Adaptive, Hybrid Feature Selection (AHFS). *Pattern Recognit.* 2021; **116**, 107932.
[28] GM Beuren and MJ Anzanello. Variable selection using statistical non-parametric tests for classifying production batches into multiple classes. *Chemom. Intell. Lab. Syst.* 2019; **193**, 103830.
[29] AI Mcleod. *The Kendall package*. Western University, Ontario, Canada, 2005.

[30] MG Kendall. Biometrika trust a new measure of rank correlation. *Biometrika* 1938; **30**, 81-93.
[31] N Cliff and V Charlin. Variances and covariances of Kendall's tau and their estimation. *Multivariate Behav. Res.* 1991; **26**, 693-707.
[32] WH Kruskal and WA Wallis. Use of ranks in one-criterion variance analysis. *J. Am. Stat. Assoc.* 1952; **47**, 583-621.
[33] T Pohlert. The Pairwise Multiple Comparison of Mean Ranks Package (PMCMR), Available at: . http://CRAN.R-project.org/package=PMCMR, accessed June 2020.
[34] JK Chung, PL Kannappan, CT Ng and PK Sahoo. Measures of distance between probability distributions. *J. Math. Anal. Appl.* 1989; **138**, 280-92.
[35] B Auffarth, M López and J Cerquides. Comparison of redundancy and relevance measures for feature selection in tissue classification of CT images. *In*: P Perner (Ed.). Advances in data mining. applications and theoretical aspects. ICDM 2010. Lecture notes in computer science. Vol 6171. Springer, Heidelberg, Germany, 2010, p. 248-62.
[36] D Dheeru and E Karra Taniskidou. Machine Learning Repository, Available: https://archive.ics.uci.edu/ml/index.php, accessed June 2020.