Applied Soft Computing 130 (2022) 109699

Contents lists available at ScienceDirect

Applied Soft Computing

journal homepage: www.elsevier.com/locate/asoc

Automatic frequency-based feature selection using discrete weighted evolution strategy

Hossein Nematzadeh [a,c,d,*], José García-Nieto [a,b,c], Ismael Navas-Delgado [a,b,c], José F. Aldana-Montes [a,b,c]

[a] ITIS Software, Universidad de Málaga, Arquitecto Francisco Peñalosa 18, Málaga, 29071, Spain
[b] Biomedical Research Institute of Málaga (IBIMA), Universidad de Málaga, Málaga, Spain
[c] Departamento de Lenguajes y Ciencias de la Computación, Universidad de Málaga, Málaga, Spain
[d] Department of Computer Engineering, Sari Branch, Islamic Azad University, Sari, Iran

ARTICLE INFO

Article history:
Received 31 January 2022
Received in revised form 5 September 2022
Accepted 2 October 2022
Available online 10 October 2022

Keywords:
Curse of dimensionality
Automatic hybrid feature selection
Filter
Wrapper
High dimensional medical datasets
Covid-19

ABSTRACT

High dimensional datasets usually suffer from curse of dimensionality which may increase the classification time and decrease the classification accuracy beyond a certain dimensionality. Thus, feature selection is used to discard redundant features for improving classification. Nonetheless, there is not a single feature selection method which could deal with all datasets. Thus, this paper proposes an automatic hybrid feature selection incorporating both filter and wrapper methods called Extended Mutual Congestion-Discrete Weighted Evolution Strategy (EMC-DWES). First, Extended Mutual Congestion (EMC) is proposed as a frequency-based filter ranker to discard irrelevant and redundant features using intrinsic statistics of features. Second, Discrete Weighted Evolution Strategy (DWES) is applied on the remaining features selected by EMC to perform the final automatic feature selection within a wrapper method. DWES clusters the features and applies mutation both to select the most relevant feature in each cluster at a time and to avoid selecting redundant features simultaneously through assigning greater weights to most informative clusters. The performance of EMC-DWES (in maximizing classification accuracy and minimizing the selected subset length) is investigated using benchmark high dimensional medical datasets including Covid-19. Likewise, the superiority of EMC-DWES in comparison with state-of-the-art is also evaluated in all datasets. The implementation of EMC-DWES is available on https://github.com/KhaosResearch/EMC-DWES.

1. Introduction

Feature selection is an approach of selecting the most effective features (predict variables) within a dataset [1–3]. Feature selection methods limit the number of features to speed up model training and improve the accuracy specially in high dimensional datasets through reducing redundant and less informative features. High dimensional datasets refer to the datasets in which the number of predict variables staggeringly exceeds the number of observations (samples) such as microarray, gene expression, biological, and most of the medical datasets [4]. Feature selection methods are categorized into five major groups, namely filter [5,6], wrapper [7,8], embedded [9,10], ensemble [11,12], and hybrid [13–15]. Filter methods utilize intrinsic properties of the predict variables (such as correlation) to develop heuristic or metaheuristic methods. However, wrapper methods generally evaluate all combination of features using a machine learning algorithm. Wrappers use classification accuracy for evaluation and thus usually have higher accuracies than filters. Unlike the filter methods which might fail to recognize the best subset of features in some situations, wrapper methods mostly have better convergence. Nevertheless, the most important weakness of wrapper methods is low performance (high execution time). Embedded methods perform variable selection along with classification simultaneously. Lasso (L1 regularization) [16] and Decision Tree [17] are among the most famous embedded feature selection methods. Ensemble methods combine the output of multiple feature rankers to find the final set of selected features instead [18]. Hybrid methods refer to any combination of wrapper, filter, embedded and ensemble methods. Feature selection methods can also be categorized into either manual or automatic. Unlike automatic methods, manual methods need a predefined threshold to determine the length of the selected subset of features [11,13]. This paper is a direct improvement of the previous research [13]

[*] Corresponding author at: Departamento de Lenguajes y Ciencias de la Computación, Universidad de Málaga, Málaga, Spain.
E-mail addresses: hnematzadeh@uma.es (H. Nematzadeh), jnieto@uma.es (J. García-Nieto), ismael@uma.es (I. Navas-Delgado), jfaldana@uma.es (J.F. Aldana-Montes).

https://doi.org/10.1016/j.asoc.2022.109699
1568-4946/

in which a novel automatic hybrid method is proposed so that the largest number of features is discarded using the proposed filter method called Extended Mutual Congestion (EMC) firstly. Then, (1 + 1) Discrete Weighted Evolution Strategy (DWES) is proposed and used as a metaheuristic algorithm within a wrapper method to select the final feature subset automatically. In fact, the proposed method is a hybrid of filter (EMC) and wrapper (DWES) methods. Briefly the contribution of paper is as follows:

1- To improve Mutual Congestion (MC) via proposing Extended Mutual Congestion (EMC) as a filter ranker both to enhance the accuracy of ranking and to deal with multi-label datasets.
2- To propose (1 + 1) Discrete Weighted Evolution Strategy (DWES) as a wrapper method to automatically select the best subset of features.

    2.1. To apply hierarchical clustering for feature selection considering minimum redundancy.
    2.2. To propose the concept of weight for clusters to improve DWES to find the most optimal subset containing informative non-redundant features automatically.

3. To combine EMC with DWES to construct a hybrid feature selection method.

The rest of this paper is organized as follows: Section 2 describes the related works and backgrounds. Section 3 briefly explains MC ranker, hierarchical clustering, and (1 + 1) evolution strategy. Section 4 shows how MC-measure can be improved to EMC and further be combined with the proposed DWES. Section 5 presents the results and discussions on benchmark high dimensional medical datasets. Finally, Section 6 concludes the paper with final remarks.

2. Related works

This section investigates the related works purposefully. First, the work regarding MC ranker is introduced. Second, the recent metaheuristic-based feature selection methods are explained. Recently, Nematzadeh et al. [13] introduced Mutual Congestion (MC) as a frequency-based filter ranker and combined it with Whale Optimization Algorithm (WOA) to propose a hybrid filter feature selection method (WOA-MC). The process started by recognizing the irrelevant features by WOA with a new defined fitness function so that half of the dimension was reduced firstly. Second, MC ranked the rest of the features and selected the best 10 features to construct the final feature subset. Finally, majority voting was applied using forward feature selection approach on the final feature subset. Measurement criteria (including accuracy, specificity, sensitivity, and Matthews Correlation Coefficient (MCC)) showed that the combination of WOA and MC improved the prediction of high dimensional binary medical datasets. Further analysis on WOA (including box plot analysis and convergence analysis) also justified the usage of WOA. In another parallel research, Alirezanejad et al. [19] proposed a filter method for feature selection of medical datasets called Xvariance (It is read as cross variance) and showed that the combination of MC and Xvariance did not any improve the prediction in comparison with individual MC or Xvariance. Furthermore, the results on binary medical datasets indicated that Xvariance had better results on binary datasets in which the number of samples exceeded the number of features. In contrast, MC outperformed Xvariance when the number of features was considerably greater than the number of samples. In addition, ROC analysis also confirmed both Xvariance and MC had acceptable results individually. Briefly explaining, Xvariance focused on calculating the distance between feature values of two distinguished labels for each dataset feature using variance. The rest of this section dedicates to study the current feature selection works utilizing metaheuristic algorithms. Vafaee et al. [20] proposed a two-phase approach to select the smallest subset of features to have the best possible classifier performance called Cellular Learning Automata-Ant Colony Optimization Feature Selection (CLACOFS). First, features were ranked using a Fisher ratio and the features with small Fisher ratio were discarded (filter method). The paper experimentally proved and justified that Fisher ratio had the highest performance among T test, Information Gain, and Z score. Then, the hybrid of Cellular Learning Automata and Ant Colony Optimization was used aligned with ROC curve to select the final subset (wrapper method). Azadifar et al. [21] proposed a hybrid method called MaPSOGS which used Fisher ratio to discard irrelevant features. The remaining features were then grouped into several clusters via graph clustering. Next, Many-Objective PSO was used with a newly defined repair operator to improve the solutions by selecting the genes from different clusters automatically. Sadeghian et al. [14] proposed a three-stage hybrid feature selection method called Ensemble Information Theory-based binary Butterfly Optimization Algorithm (EIT-bBOA) in which Minimal Redundancy-Maximal New Classification Information (MR-MNCI) was used in the first phase to discard the 80% of irrelevant features. Second, the best feature subset was selected using Information Gain binary Butterfly Optimization Algorithm (IG-bBOA) which was a developed version of S-shaped binary Butterfly Optimization Algorithm (S-bBOA) that typically ignored the redundancy and relevancy of features. Finally, a similarity based ranking method was used to select the final feature subset manually with threshold $\tau = 30$ using the ensemble of reliefF and Fisher Score. Likewise, further analysis revealed that Information Theory-based binary Butterfly Optimization Algorithm (IT-bBOA) which was the combination of MR-MNCI and IG-bBOA outperformed binary Whale Optimization Algorithm (bWOA), binary Crow Search Algorithm (bCSA) and binary Gray Wolf Optimization (bGWO) in terms of convergence within 100 iterations. Amini et al. [22] proposed a hybrid method including two stages. First, Genetic Algorithm was used in a wrapper way to reduce the dimensionality and number of predictors. Then, Elastic Net (EN) was used as a famous embedded method to increase the optimality of solutions generated by GA. The proposed model with tuned hyper-parameters was evaluated through 3-fold cross validation and the performance was compared in terms of relative RMSE with four different scenarios. Abasabadi et al. [11] proposed a three-stage Automatic Ensemble Feature Selection method (ATFS). First, three rankers were applied to the input dataset to obtain multiple rankings. These rankers were MC which was proposed by Nematzadeh et al. [13] along with a newly proposed filter ranker called Sorted Label Interference (SLI), and existing reliefF recalling that SLI was a frequency-based ranker inspired from MC. Then, the concept of fast non-dominated sorting was used with automatic thresholding capability to combine the output of rankers (non-dominated sorting is basically used in the selection operator of multi-objective metaheuristic algorithms to sort the solutions according to the Pareto dominance principle). Finally, the smaller ranked feature subsets obtained from fast non-dominated sorting were used to generate final feature subsets. The ensemble ATFS was proposed for binary datasets and the feature selection process was automatic. Abasabadi et al. [23] also proposed another hybrid method using genetic algorithm called $GA_{rank\&rand}$ so that the newly proposed filter method SLI-γ (inspired from MC) was combined with genetic algorithm that exploited Artificial Neural Network (ANN) and K-Nearest Neighbors (KNN)

Table 1
Summary of related works.

Method	Evolutionary concept	Type	Selection procedure	Problem domain	Response variable	Year
WOA-MC [13]	Whale Optimization Algorithm	Hybrid Filter + Filter	Manual	Classification	Binary	2019
ATFS [11]	Non-dominated sorting	Ensemble of three filters	Automatic	Classification	Binary	2021
$GA_{rank\&rand}$ [23]	Genetic algorithm	Hybrid filter + Wrapper	Automatic	Classification	Binary	2022
EIT-bBOA [14]	Butterfly Optimization Algorithm	Hybrid Filter + Wrapper + Filter	Manual	Classification	Multi-label	2021
MPSONC [5]	Multi-Objective PSO	Filter	Automatic	Classification	Multi-label	2020
CLACOFS [20]	Ant Colony Optimization Algorithm	Hybrid Filter + Wrapper + Filter	Automatic	Classification	Multi-label	2016
GA-EN [22]	Genetic algorithm	Hybrid Wrapper + Embedded	Automatic	Regression	Continuous	2021
MaPSOGS [21]	Many-Objective PSO	Hybrid Filter + Wrapper	Automatic	Classification	Multi-label	2021
Xvariance [19]	Not Applicable	Filter	Manual	Classification	Binary	2020

in fitness function. $GA_{rank\&rand}$ only used 1% of the best features identified by SLI-γ for initial population generation in genetic algorithm. $GA_{rank\&rand}$ achieved good accuracy on binary medical datasets but the computation time was very expensive particularly when the fitness function was ANN. In another recent work, Rostami et al. [5] proposed an automatic filter feature selection called Multi-objective Particle Swarm Optimization algorithm and Node Centrality (MPSONC) through integrating of node centrality (calculated from graph representation) and multi-objective PSO search algorithm. The proposed method was targeted for medical datasets with minimum redundancy and highest relevance of selected features. Table 1 demonstrates the related works in a tabular format. Generally, metaheuristic algorithms are time consuming and increasing the number of iterations even lead to larger execution time and sometimes makes the metaheuristic algorithm non-practical. This usually happens because metaheuristics algorithms have too many operations to set up. In contrast, the proposed DWES is very fast because it only uses mutation and it provides more explorations within a reasonable time. All in all, this paper proposes a new filter ranker for multi-label datasets called Extended Mutual Congestion (EMC) and combines it with a wrapper (1 + 1) Discrete Weighted Evolution Strategy (DWES) to construct a hybrid automatic feature selection method (EMC-DWES). The reason for selecting evolution strategy is that it only uses one operation (mutation) which is sufficient for convergence theoretically. The EMC ranker in the proposed hybrid method discards the large number of irrelevant features and minimizes the dimensionality of datasets considerably. The small dataset constructed using EMC ranker increases the overall performance when combining by DWES.

3. Preliminaries

This section introduces MC ranker, hierarchical clustering, and (1 + 1) continuous evolution strategy. MC ranker is then enhanced in Section 4 to construct Extended Mutual Congestion (EMC). Likewise, (1 + 1) continuous evolution strategy is further improved in Section 4 to construct Discrete Weighted Evolution Strategy (DWES) using hierarchical clustering with weights for clusters. EMC and DWES are finally combined to construct the proposed method (EMC-DWES).

3.1. Mutual congestion

According to Nematzadeh et al. [13] the values of each feature within a binary dataset should be sorted ascendingly in Mutual Congestion (MC) firstly. Then, the response variable (labels) also should be sorted according to each ascendingly sorted feature. Next, the algorithm proceeds by calculating the frequency in which two class labels interfere for each feature so that MC $\in$ [0, 1]. The less the MC measure is, the better that feature is for classification. MC measure is calculated as in Eq. (1) so that n_2^+ and n_3^- are the number of positive and negative labels within interference region respectively. Likewise, n_1^+ and n_4^- are the number of positive and negative labels out of the interference region respectively. Fig. 1 illustrates the best, general, and worst cases for a dataset of size $n \times m$ so that the purple line shows the interference region of two class labels. Fig. 1 also shows if the values of an arbitrary feature are sorted ascendingly in a binary dataset and then the labels are sorted accordingly, three possible sortings are generated in which the most common sorting is the general case. Assuming $L_1^+, L_2^+, \ldots, L_s^+$ are the sorted positive labels with the size of s (showed by red in Fig. 1) and $L_1^-, L_2^-, \ldots, L_p^-$ are the sorted negative labels with the size of p (showed by blue in Fig. 1) so that $s + p = n$. As such, the general case in Fig. 1 shows that the list of sorted labels (including positive and negative labels) starts with consecutive positive labels (red color) and continues by the mixture of positive and negative labels (purple color) and finally ends with consecutive negative labels (blue color). Therefore, n_2^+ and n_3^- are the number of positive and negative labels in purple zone. Likewise, n_1^+ and n_4^- are the number of positive and negative labels in red and blue zones respectively. Based on Eq. (1) and Fig. 1, the MC values of 0 and 1 show that the feature is completely separable (best case) or totally non-separable (worst case) respectively. However, in practice most of the features within a dataset have a degree of separability in [0,1] (general case) according to the calculated MC measure. Regarding time complexity of MC, the initial sorting can be done using quicksort algorithm to save time which has the time complexity of O ($n \log n$) in the best and average case and O (n^2) in the worst case. All these steps are repeated for all m features; thus, Mutual Congestion has O (mn^2) computation complexity in the worst case recalling that identifying samples with their respective positive or negative labels and interference calculation between labels is done in O (n).

$$MC = \frac{n_2^+ + n_3^-}{n_1^+ + n_2^+ + n_3^- + n_4^-} \tag{1}$$

3.2. Hierarchical clustering

Hierarchical clustering [24] is an agglomerative bottom-up approach that starts with n clusters (n is the number of observations). At each step, the closest pair of clusters is merged until

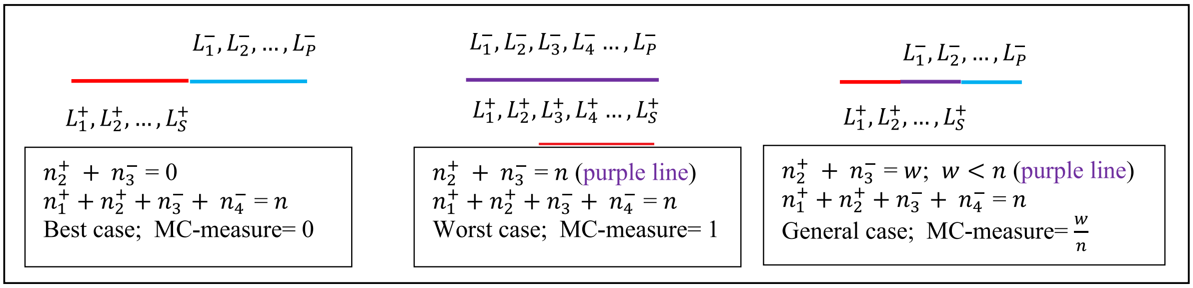
$L_1^-, L_2^-, \dots, L_P^-$
$L_1^+, L_2^+, \dots, L_S^+$
$n_2^+ + n_3^- = 0$
$n_1^+ + n_2^+ + n_3^- + n_4^- = n$
Best case; MC-measure= 0
$L_1^-, L_2^-, L_3^-, L_4^- \dots, L_P^-$
$L_1^+, L_2^+, L_3^+, L_4^+ \dots, L_S^+$
$n_2^+ + n_3^- = n$ (purple line)
$n_1^+ + n_2^+ + n_3^- + n_4^- = n$
Worst case; MC-measure= 1
$L_1^-, L_2^-, \dots, L_P^-$
$L_1^+, L_2^+, \dots, L_S^+$
$n_2^+ + n_3^- = w;\ w < n$ (purple line)
$n_1^+ + n_2^+ + n_3^- + n_4^- = n$
General case; MC-measure= $\frac{w}{n}$

Fig. 1. MC-measure in three cases.

reaching only one cluster finally. Thus, cluster merging is done based on Eq. (2) where K is the entire number of clusters.

$$merge\ C_i\ and\ C_j\ if\ d\left(C_i, C_j\right) = \min_{i,j}\left[d\left(C_i, C_j\right) : i \in K, j \in K\right] \tag{2}$$

Hierarchical clustering has three main methods for calculation of distance between two clusters called, single-linkage Eq. (3), average-linkage Eq. (4), and complete-linkage Eq. (5) so that s and t are cluster members inside C_i and C_j respectively.

$$d\left(C_i, C_j\right) = min_{s,t}\left[d\left(s, t\right) : s \in C_i, t \in C_j\right] \tag{3}$$

$$d\left(C_i, C_j\right) = \frac{\sum t \in C_j \sum s \in C_i d(s, t)}{n_i n_j} \quad ;$$

$$n_i : number\ of\ members\ in\ C_i \tag{4}$$

$$d\left(C_i, C_j\right) = max_{s,t}\left[d\left(s, t\right) : s \in C_i, t \in C_j\right] \tag{5}$$

Single linkage results in clusters with different densities. Complete linkage tends to have clusters with same densities like Kmeans clustering algorithm. However, complete linkage is robust to noise unlike Kmeans which is noise sensitive. Average linkage comes between single linkage and complete linkage.

3.3. (1 + 1) Evolution strategy

(1 + 1) continuous evolution strategy is the simplest evolution strategy which only uses mutation so that one parent generates one offspring by applying normally distributed mutation [25,26]. Algorithm 1 shows the steps for continuous (1 + 1) evolution strategy in which P and the respective feasible ranges are determined in line 1 initially. Second, the parent is constructed by selecting random values (from respective feasible ranges) for each parameter and the related fitness is also calculated in lines 2 and 3. Then, the parent is mutated by adding a normally distributed random variable α with mean of 0 and deviation of δ to construct the offspring and the related fitness is calculated as well in lines 5–7. Algorithm 1 continues by comparing the fitness of offspring and parent to select the better solution. Finally, the process of mutation and creating new offspring continues within a loop until stopping criterion is considered and the final solution is returned. The proposed method in this paper will introduce a new (1 + 1) Discrete Weighted Evolution Strategy using hierarchical clustering to determine P in step 1 of Algorithm 1.

4. Proposed method

The proposed method, Extended Mutual Congestion-Discrete Weighted Evolution Strategy (EMC-DWES), is initiated by discarding significant number of features of the input dataset using EMC as a filter ranker firstly. Then, feature scaling is done as a standardization step so that the values of independent features are normalized. Next, the newly constructed dataset with a reduced dimensionality is used within a wrapper feature selection using a Discrete Weighted Evolution Strategy (DWES). The idea is maximum feature reduction by EMC to get the most functionality from DWES. Fig. 2 clearly shows the general phases of the EMC-DWES so that the size of the input dataset considerably decreases after applying EMC (k ≪ m) and DWES (p ≪ k).

4.1. Extended mutual congestion

The basic idea of Extended Mutual Congestion (EMC) is finding the most informative features with best classification power. Assuming a three-label dataset with two features (F1, F2) in Fig. 3, the image of samples on F1 axis confirms that F1 has a better classification functionality in comparison with F2. However, if the response variable (label) is ascendingly sorted based on the individual feature values, a typical array in Fig. 3 may be achieved for an arbitrary feature in practice. This happens because the real labeling is not done basically by using just one feature specially in a high dimensional dataset. Thus, EMC tries to find the frequency of irrelevant labels based on an individual label for each ascendingly sorted feature. Irrelevant labels refer to those which are non-separable based on a certain label. As such, the labels specified by red and dashed brace are assumed as separable and non-separable labels respectively based on the red label. EMC calculates this frequency for each label and finally generates EMC ∈ [0,1] for each feature so that the smallest EMCs analogize the best features.

Algorithm 1. Continuous (1+1) evolution strategy

```
Input: $problem$: a problem
Output: $Parent$
1. Determine $\{X_{1min}, X_{1max}\}, \{X_{2min}, X_{2max}\}, \dots, \{X_{Pmin}, X_{Pmax}\}$
2. $Parent \leftarrow x_1, x_2, \dots, x_P$
3. $Z_1 \leftarrow Fitness(Parent)$
4. WHILE stopping criterion is not met DO
5.     $x_i' \leftarrow x_i + \alpha\,(0, \delta), \quad i = 1,2, \dots, P$
6.     $Offspring \leftarrow x_1', x_2', \dots, x_P'$
7.     $Z_2 \leftarrow Fitness(Offspring)$
8.     IF $Z_2 > Z_1$
9.         $Parent \leftarrow Offspring$
10.        $Z_1 \leftarrow Z_2$
11.    END IF
12.END WHILE
13. Return $Parent$
```

Assuming X is a high dimensional dataset of size $n \times m$ ($n \ll m$) as illustrated in Table 2 and Eq. (6) with labels in Eq. (7). Thus, any instance of X can be defined based on feature values as shown in Eq. (8). Likewise, each feature vector is defined accordingly in Eq. (9).

$$X = (x_1, x_2, \dots, x_n)^t \tag{6}$$

$$L = (l_1, l_2, \dots, l_k) \tag{7}$$

$$x_i = (f_{i1}, f_{i2}, \dots, f_{im}) \quad i = 1, 2, \dots, n \tag{8}$$

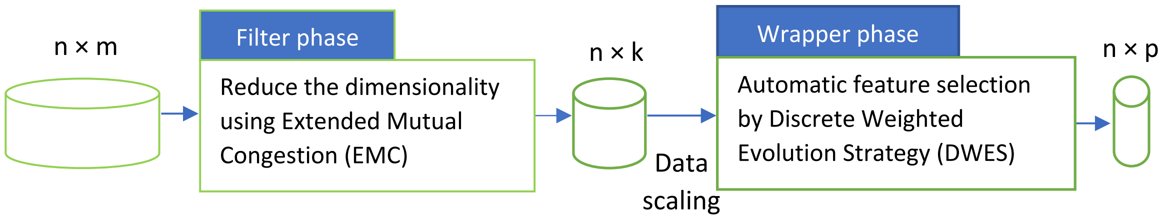
n × m
Filter phase
Reduce the dimensionality using Extended Mutual Congestion (EMC)
n × k
Data scaling
Wrapper phase
Automatic feature selection by Discrete Weighted Evolution Strategy (DWES)
n × p

Fig. 2. EMC-DWES in abstract view.

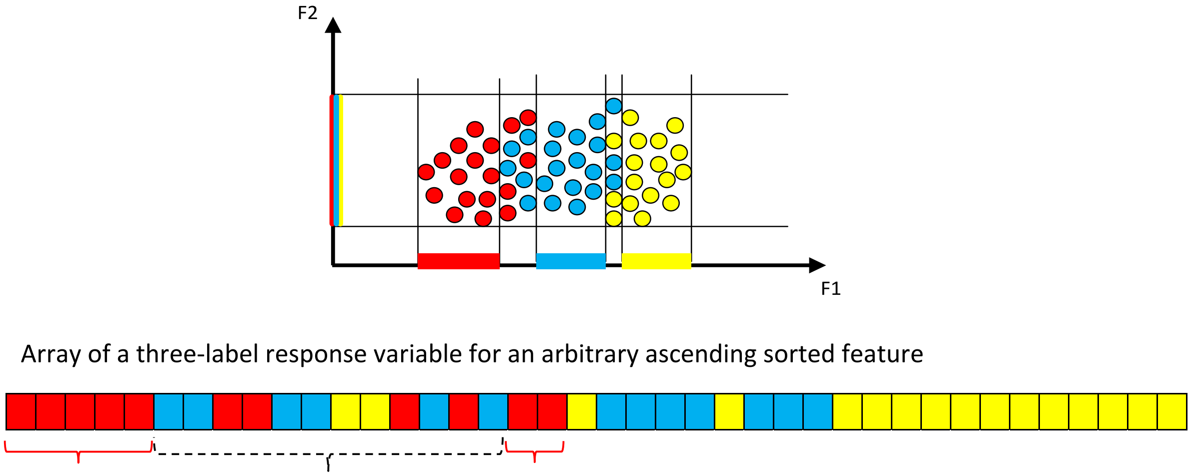
F2
F1
Array of a three-label response variable for an arbitrary ascending sorted feature

Fig. 3. A three-labeled dataset with two features (F1,F2) and an array of response variable based on an ascending sorted values of an arbitrary feature.

Table 2
A sample high dimensional dataset X of size $n \times m$.

F_1	F_2	…	F_m	*Label*
f_{11}	f_{12}	…	f_{1m}	L
f_{21}	f_{22}	…	f_{2m}	L
…	…	…	…	…
f_{n1}	f_{n2}	…	f_{nm}	L

$$F_j = \left(f_{1j}, f_{2j}, \ldots, f_{nj}\right)^t \qquad j = 1, 2, \ldots, m \tag{9}$$

Assuming L is the set of response variables (labels) in Eq. (7), then y^{lp} is the set of instances with similar labels (lp) in Eq. (10) subsequently.

$$y^{lp} = \left(x_1^{lp}, x_2^{lp}, \ldots, x_{n_p}^{lp}\right) = \left\{x_j^{lp}\right\}_{j=1}^{n_p} \qquad p = 1, 2, \ldots, k \tag{10}$$

where

$$\cup_{p=1}^{k} y^{lp} = X, \qquad \sum_{p=1}^{k} n_p = n$$

The definition of y^{lp} in terms of features is also shown in Eq. (11)

$$y^{lp} \equiv \left(f_1^{lp}, f_2^{lp}, \ldots, f_{n_p}^{lp}\right)^t \tag{11}$$

In this phase, each feature F_j is sorted ascendingly and the respective vector of labels will be sorted accordingly. Thus, for each ascendingly sorted F_j, there exists a sorted vector of labels. As a result, y^{lp} corresponds to Eq. (12) which is a permutation of y^{lp} in Eq. (11). However, the response variable is generally sorted as shown in Fig. 3 in practice in accordance with its respective sorted feature values. Fig. 4 is a subset of sorted response variable for label r_i. The more the length of the blue section is in Fig. 4, the less that feature is good for classifying label r_i. The blue section in Fig. 4 is a region with non-separable labels.

$$y^{lp} \equiv \left(g_1^{lp}, g_2^{lp}, \ldots, g_{n_p}^{lp}\right)^t \tag{12}$$

Finally, the separability of each feature is calculated using Eq. (13) so that the less EMC is, the better classifier that feature is. m_{r_i} is the number of instances with non-separable labels for r_i (blue section in Fig. 4). θ_{r_i} is the sum of the number of instances with separable labels (red sections in Fig. 4) and number of instances with non-separable labels (blue section in Fig. 4) for label r_i.

$$EMC(j) = \left\{\frac{\sum_{i=1}^{k} m_{r_i}}{\sum_{i=1}^{k} \theta_{r_i}}\right\}_{j=1}^{m} \tag{13}$$

In order to understand how *EMC* is calculated for each feature, the array in Fig. 3 is used in which the number of instances with non-separable labels for red, blue, and yellow labels are 12, 18, and 15 respectively. Likewise, the number of instances with separable labels for red, blue, and yellow labels are 7, 5, and 14 respectively. Thus, $EMC = \frac{12+18+15}{12+18+15+7+5+14} = \frac{45}{71} = 63\%$. The proposed method starts by applying EMC as a filter method to the high dimensional datasets initially. Next, features are ranked based on their respective EMC measures ascendingly and 95% of the least informative features are discarded. Finally, the features with the best EMC measures are retained to construct the new dataset X' as shown in Table 3 based on Fig. 2 and Eq. (14) (Recall that m is the dimensionality of the input dataset).

$$m' = 0.95 \times m \tag{14}$$

4.2. Discrete weighted evolution strategy

Prior to propose and apply Discrete Weighted Evolution Strategy (DWES) feature scaling should be done on X' which helps

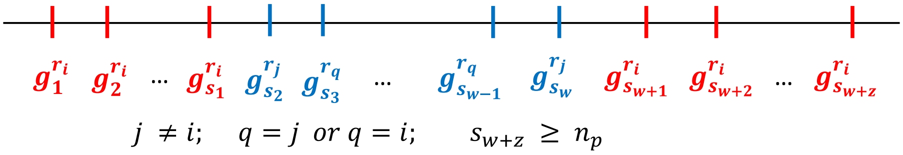

Fig. 4. Non-separable labels (blue) and separable labels (red) for label r_i.

Table 3
Dimensionality reduced dataset X' with size $n \times m'$.

F'_1	F'_2	…	$F'_{m'}$	*Label*
f_{11}	f_{12}	…	$f_{1m'}$	L
f_{21}	f_{22}	…	$f_{2m'}$	L
…	…	…	…	. . .
f_{n1}	f_{n2}	…	$f_{nm'}$	L

in speeding up the calculations as well as better convergence of classifiers. Assuming F_i is a feature vector to be scaled and μ_i and σ_i are the mean and standard deviation of F_i, data standardization is shown in Eq. (15). The entire features in the dataset X' should be scaled accordingly.

$$F_i = \frac{F_i - \mu_i}{\sigma_i} \qquad i = 1, 2, \ldots, m' \tag{15}$$

The Discrete Weighted Evolution Strategy (DWES) should be formulated to cluster the features of X'. Thus, the dataset X' should be transposed firstly. The steps of DWES are introduced as follows:

Step 1: First, q and the respective discrete feasible ranges in the proposed DWES are determined using hierarchical clustering as shown in Eq. (16). Hierarchical clustering exploits agglomerative clustering using single linkage, average linkage, and complete linkage. As a result, the features of Table 3 ($F' = \{F'_1, F'_2, \ldots, F'_{m'}\}$) are divided into q number of clusters in C^q and can be further expanded in Eq. (17). The algorithm is tuned so that each cluster c_i has an initial selection weight of W_{c_i} as shown in Eq. (18).

$$C^q = \text{Hierarchical clustering}\left(F', \mathrm{q}\right) \tag{16}$$

$$C^q = \{c_1, c_2, \ldots, c_q\} \tag{17}$$

where

$$c_i = \left\{F'^i_1, F'^i_2, \ldots, F'^i_{s_i}\right\} \; s_i = |c_i|\,,\; \cup_{i=1}^{q} c_i = F'$$

$$W_{C_i} = 0.5 \qquad i = 1, 2, \ldots, q \tag{18}$$

Step 2: To construct the parent, a feature from each cluster should be selected considering W_{c_i} so that $\mathrm{c_i}$ is selected with its respective W_{c_i} (which initially equals 0.5 in Eq. (18)) in Eq. (19).

$$\boldsymbol{Parent} = \boldsymbol{W}_{\boldsymbol{c_1}}\left(F'^{1}_{jc_1}\right), \boldsymbol{W}_{\boldsymbol{c_2}}\left(F'^{2}_{jc_2}\right), \ldots, \boldsymbol{W}_{\boldsymbol{c_q}}\left(F'^{q}_{jc_q}\right) \tag{19}$$

Step 3: The fitness of parent in Eq. (20) (which is the accuracy of classifier) shows how close the given subset of features is to the optimum solution. The proposed DWES uses Support Vector Machine (SVM) with linear kernel and C = 50 and Decision Tree (DT) regarding two-labeled and multi-label datasets respectively, though DWES is flexible to use any classifier.

$$Z_1 = accuracy\,(\boldsymbol{Parent}) \tag{20}$$

Step 4: The offspring is generated through exploring each cluster to select a new feature as shown in Eq. (21). In fact, a feature within a certain cluster is selected randomly if that cluster is selected based on its associated weight. As a result, each member of c_i is mutated within its range specified by hierarchical clustering.

$$\boldsymbol{Offspring} = \boldsymbol{W}_{\boldsymbol{c_1}}\left(F'^{1}_{gc_1}\right), \boldsymbol{W}_{\boldsymbol{c_2}}\left(F'^{2}_{gc_2}\right), \ldots, \boldsymbol{W}_{\boldsymbol{c_q}}\left(F'^{q}_{gc_q}\right) \tag{21}$$

Step 5: To evaluate the goodness of the offspring, the fitness of the offspring should be calculated. Like Eq. (20) the fitness of the offspring is calculated as in Eq. (22).

$$Z_2 = accuracy\,(\boldsymbol{Offspring}) \tag{22}$$

Step 6: If ($Z_2 > Z_1$**),** then parent should be substituted by offspring for next generation. In addition, by selecting the offspring the selection weight of the respective clusters ($\mathrm{W}_{\mathrm{c_i}}$) should be updated according to Eq. (23) considering that W_{c_i}s do not exceed 1. The initial value of $W_{\mathrm{C_i}} = 0.5$ gives equal chance to clusters for being selected at the beginning of DWES under the assumption that the weight of each cluster does not exceed 1. If ($Z_2 < Z_1$), the parent subset as well as selection weight of clusters remain unchanged. The parameter α in Eq. (23) is experimentally set to 0.1.

$$W_{\mathrm{C_i}} \leftarrow \mathrm{W}_{\mathrm{C_i}} + \alpha\left(1 - \mathrm{W}_{\mathrm{C_i}}\right) \tag{23}$$

Step 7: Go to Step 4 until specified number of generations is considered.

Algorithm 2 clearly shows the steps of EMC-DWES in which lines 1–4 initialize the algorithm, lines 11–23 generate an offspring for each iteration, lines 25–34 updates the cluster weights in case of fitness improvement. It is noteworthy mentioning that DWES also considers the substitution of parent by the offspring with same accuracy, but shorter length in implementation as well. The stopping condition in DWES is the maximum iteration which is set to 200 iterations.

5. Experimental results

This section introduces the datasets firstly. Second, the used classifiers are introduced following the respective experimental setups. Finally, the measurement criteria are introduced.

5.1. Datasets

The benchmark high dimensional gene expression datasets are introduced in Table 4 including 5 binary and 4 multi-label datasets. A brief description of each used dataset is given in the following. Colon is a binary dataset of colon cancer patients with negative and positive predictions. CNS is a central nervous system embryonal tumor binary dataset containing two classes of survivors and failures. Survivors are patients who are alive after treatment, but the failures are those who succumbed to their disease. GLI is a dataset with transcriptional profiling of gliomas. The aim is to predict whether an initial tumor is diagnosed as Grade III or IV glioma of any histologic type on initial surgical treatment. SMK is a binary dataset with gene expression data from smokers with and without lung cancer. Leukemia is a binary dataset consists of bone marrow samples to distinguish Acute Lymphoblastic Leukemia (ALL) and Acute Myeloid Leukemia (AML) classes.

Leukemia-II is a multi-label dataset which distinguishes three classes of AML, T-cell and B-Cell. Covid-19 is a gene expression dataset of three disease states including observations with no virus, observations infected with virus but this virus is not Covid, and observations infected with Covid virus. SRBCT contains 4 different childhood tumors namely Ewing's family of tumors (EWS), Neuroblastoma, Non-Hodgkin lymphoma (Burkitt's lymphoma, BL) and Rhabdomyosarcoma (RMS). MLL is a multi-label leukemia dataset including three types of leukemias namely ALL, AML, and Mixed-Lineage Leukemia (MLL).

Algorithm 2. EMC-DWES

```
Input: $F' = \{F'_1, F'_2, \dots, F'_{m'}\}$
Output: Best_subset
q ← Number of clusters
C ← $\{c_1, c_2 \dots, c_q\}$  // clustering of features (F')
$W_{c_i} \leftarrow 0.5$  ,  for  i = 1,2, ..., q
Best_subset ← NULL
WHILE stopping condition is not met DO
  Random_subset ← Generate a random subset
  IF Best_subset = NULL  THEN
        Best_subset ← Random_subset
  END IF
  Subset[1..q] ← Random_subset
  counter ← 0
  FOR i ← 1 to q
    IF rand(0,1) ≤ $Wc_i$ THEN
        counter ← counter + 1
        Feature ← Select a member in $c_i$
        Subset[counter] ← Feature
        Position_C[counter] ← $i$
        Position_M[counter] ← Determine position in $c_i$
        corresponds to Feature
    END IF
  END FOR
  IF counter = = 0 THEN
     counter ← q
  END IF
  Fitness ← Calculate accuracy of Subset [1..counter]
  IF Fitness > fitness (Best_subset) THEN
    Best_subset ← Subset
    FOR i ← 1 to counter
        index ← Position_C[i]
        $Wc_{index} \leftarrow Wc_{index} + \alpha(1 - Wc_{index})$
        IF $Wc_{index} > 1$ THEN
           $Wc_{index} \leftarrow 1$
        END IF
    END FOR
  END IF
END WHILE
Return Best_subset
```

5.2. Measurement criteria

The following measurement criteria [11,13,14,27] are used to investigate the strength and success of EMC-DWES in feature selection recalling that True Positive (TP) and True Negative (TN) refer to correctly prediction of positive and negative classes respectively. Likewise, False Positive (FP) and False Negative (FN) refer to incorrectly prediction of positive and negative classes.

- *Accuracy*: Accuracy is calculated for both binary and multi-label datasets based on Eqs. (24) and (25) respectively so that *TS* stands for the Test Set, C_i and L_i are the classifier's prediction and the real label for *i*th element of the *TS* respectively, and $|TS|$ is the number of all observations in the Test Set. The numerator of Eq. (25) can be considered as a dummy variable that has the value of 0 or 1 so that if the condition $(C_i = L_i)$ meets, $1(C_i = L_i)$ in the numerator will be 1 and otherwise 0.

Table 4
Specification of datasets.

Datasets	Sample size	Feature size	Number of classes	Year of publication
Colon [13]	62	2000	2	1999
CNS [11]	60	7129	2	2002
GLI [11]	85	22,283	2	2004
SMK [11]	187	19,993	2	2007
Leukemia [28]	72	7129	2	1999
Leukemia-II [29]	72	7129	3	1999
Covid-19 [30]	234	15 979	3	2020
MLL [31]	72	12,582	3	2002
SRBCT [29]	83	2308	4	2001

$$Accuracy_{binary} = \frac{TP + TN}{TP + TN + FP + FN} \tag{24}$$

$$Accuracy_{multi-label} = \frac{\sum x_i \in TS \ \mathbf{1}(C_i = L_i)}{|TS|} \tag{25}$$

- *Precision*: Precision in Eq. (26) is solely calculated for binary datasets and denotes the fraction of relevant instances among the retrieved instances.

$$Precision = \frac{TP}{TP + FP} \tag{26}$$

- *Recall*: Recall in Eq. (27) is solely calculated for binary datasets and denotes the fraction of relevant instances that are retrieved.

$$Recall = \frac{TP}{TP + FN} \tag{27}$$

- Fscore: F-score in Eq. (28) is solely calculated for binary datasets and denotes the harmonic mean of precision and recall.

$$Fscore = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{28}$$

- *Subset length*: subset length is the length of automatically selected features by the proposed method.

It is noteworthy mentioning that all the above criteria are calculated based on the average of 10 times running the program.

5.3. Experimental setup

The evaluation of EMC-DWES and calculation of the measurement criteria in Section 5.2 can be investigated via any classifier. Nonetheless, Support Vector Machine (SVM) and Decision Tree (DT) are selected for investigation of the results in binary and multi-label datasets respectively to avoid having multiple figures and tables with semantically same results [27].

- SVM looks for a hyperplane with the most possible distance from classes using support vectors. The hyperplane is located exactly at the middle of support vectors. The perpendicular distance of support vectors to the hyperplane is called margin. SVM uses a constraint (C) to avoid having too many points in the margin. The linearity or non-linearity of SVM is specified by the kernel as well. The SVM classifier in this research uses a linear kernel with a cross validated C for each dataset. Experiments also revealed that C does not have a considerable impact in classification accuracy.
- DT selects the most informative features with respective cut points. Cut points are the points that divide the data space of a certain feature into binary regions and are defined for each feature. The region can be a terminal node (leaf of a tree). There exists a split quality measure that examines the cut points and both recognizes the best cut point of each feature and the best features periodically within a recursive binary splitting algorithm. Finally, the best features and their respective cut points constitute

Table 5
Average accuracy of EMC-DWES on benchmark datasets.

Dataset	Linkage type	Number of clusters in DWES	Average subset length	Average accuracy of EMC-DWES	Average initial accuracy
Colon	S-link	10	6	0.93	0.81
	A-link	10	6	0.93	
	C-link	10	6	0.93	
CNS	S-link	40	34	0.88	0.64
	A-link	50	26	0.87	
	C-link	40	28	0.90	
GLI	S-link	20	14	0.91	0.91
	A-link	50	33	0.96	
	C-link	50	29	0.98	
SMK	S-link	20	13	0.78	0.73
	A-link	20	13	0.78	
	C-link	20	14	0.78	
Leukemia	S-link	20	12	1	0.97
	A-link	30	16	1	
	C-link	30	16	1	
Leukemia-II	S-link	40	23	0.86	0.84
	A-link	40	20	0.95	
	C-link	30	20	0.97	
Covid-19	S-link	40	24	0.73	0.63
	A-link	40	25	0.74	
	C-link	40	25	0.75	
MLL	S-link	40	22	0.91	0.85
	A-link	10	6	0.94	
	C-link	10	6	0.96	
SRBCT	S-link	20	13	0.92	0.79
	A-link	20	10	0.93	
	C-link	20	12	0.94	

the DT. DT algorithm stops when the tree achieves complete purity either when it cannot proceed anymore (from each branch it reached to the intrinsic terminal nodes) or forces to stop by the analyst. When DT algorithm stops, the recognized region (terminal nodes) should be labeled. The labeling in each region is done based on majority voting of existing labels in that region. The decision tree in this research uses *Gini* index impurity measure. The nodes in the tree are expanded until all leaves are pure or contain less than 2 samples. The parameters *random_state* and *class_weight* are None and *ccp_alpha* is the default 0.

This research is implemented using Python 3.8 platform on a computer with Core i5 processor (1.60 GHz–2.30 GHz), 12 GB RAM, 720 GB HDD, and 64-bit Windows 10 operating system.

5.4. Performance analysis

Table 5 shows the average accuracy of EMC-DWES (recalling that SVM and DT are used for binary and multi-label datasets respectively to calculate measurement criteria). Number of clusters in DWES and Linkage type are two hyper-parameters need to be cross validated by the analyst. Table 5 indicates that the average accuracy of EMC-DWES exceeds initial accuracy in all datasets (Initial accuracy is the accuracy without feature selection while the results are the average of 10 times running the program). Moreover, accuracy of EMC-DWES often increases from single linkage to complete linkage except in CNS in which single linkage has competitively better accuracy than average linkage and similarly, the average accuracies do not differ in Colon, SMK, and Leukemia using any linkage type. Likewise, the average subset length is acceptable with respect to the number of clusters (maximum possible subset length) in DWES. Fig. 5 shows precision, recall, and Fscore of EMC-DWES calculated for binary datasets and its comparison with initial state in which no features have been selected. The parameters of EMC-DWES (number of clusters in DWES and retaining rate of EMC) are set as in Table 5. The proposed method increased the precision of all datasets. However, it could not successfully increase the recall of GLI and SMK. Nonetheless, EMC-DWES could successfully increase the Fscore of all binary datasets. Fig. 6 shows the average accuracies achieved with respect to the number of clusters in DWES. The accuracy mostly becomes stable or decreasing while reaching 50 number of clusters. Moreover, the overall target of EMC-DWES is to increase the accuracy while finding the optimal subset length. Thus, the investigation of accuracy in terms of number of clusters in Table 5 is restricted in $[10, 50]$ with step size of 10. Fig. 6 also reveals that average linkage and complete linkage are more reliable than single linkage in achieving higher accuracies. This unreliability of single linkage is completely evident in Leukemia-II and slightly obvious in GLI, Covid-19 as well as MLL. Nonetheless, in some minority of cases (like Leukemia) single linkage outperforms other linkages to achieve higher accuracies sooner with less subset length in Fig. 7. This means that cross validation could be used to select the best linkage for each dataset individually. Fig. 7 confirms that by increasing the number of clusters the average subset length is not increased strongly which is a positive aspect of EMC-DWES.

5.4.1. EMC vs. MC

EMC outperforms MC in two ways. First, EMC can deal with multi-label datasets unlike MC. Second, EMC assigns more accurate weights to features than MC. Assuming Fig. 8 is an ordered response variable based on an arbitrary ascending sorted feature in a binary dataset (blue and red labels) with 20 observations. MC finds this feature 40% bad $\left(\frac{1+7}{2+1+7+10} = 0.4\right)$. However, EMC calculates the bad ratio of $(\frac{1+7}{1+7+9+10} = 0.3)$ for the same feature. The observation specified with an arrow in Fig. 8 is the reason for two different calculations. This feature is indeed a good feature in terms of separability of blue and red labels. Since EMC calculates more accurate weights, it could more possibly retain the feature and pass to DWES for final feature selection. In contrast, MC is more likely to discard this informative feature. To achieve such a superiority, EMC executes more time in comparison with MC. Time complexity of EMC for a dataset of size $n \times m$ is calculated as follows. Ascending sorting of each feature can be done using

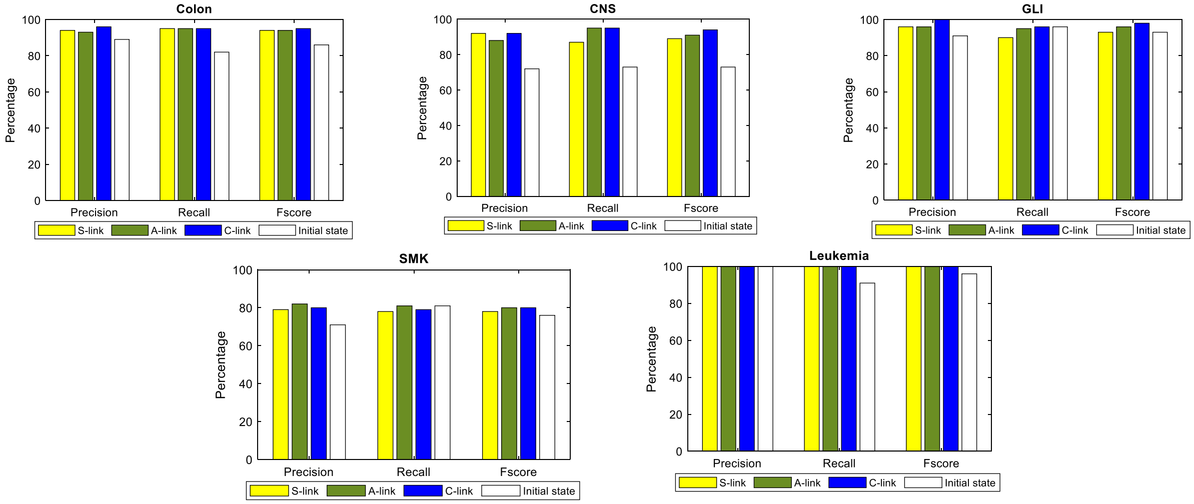

Fig. 5. Precision, recall, and Fscore of binary datasets using EMC-DWES.

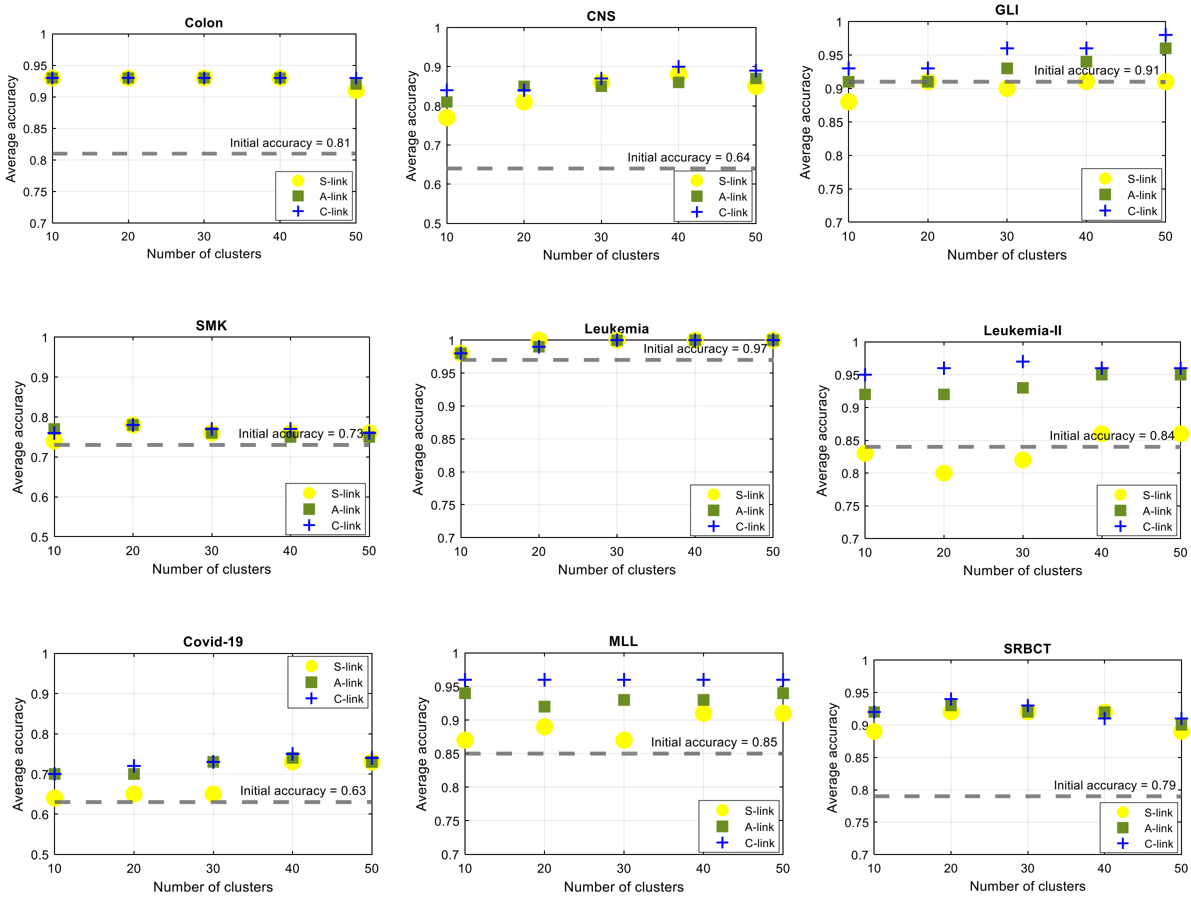

Fig. 6. Average accuracy of EMC-DWES with respect to number of clusters in DWES.

quicksort algorithm which has the time complexity $O(n^2)$ in the worst case. Thus, the overall time complexity of sorting is $O(mn^2)$, since sorting is done for all m features. Moreover, calculating both the number instances with non-separable and separable labels based on a certain label has the complexity of $O(n)$ which then is multiplied by number of labels (l) and number of features (m). Thus, the overall complexity of EMC is $\mathrm{O}\left(mn^2\right) + \mathrm{O}(mln)$ in the worst case. Comparing to time complexity of MC which is $O\left(mn^2\right)$, EMC increases the execution time by $O(mln)$. It is evident that execution time of EMC directly depends on the number of instances, features, and labels (with more concentration on instances). According to the acceptable time complexity of EMC, Colon and Covid-19 have the fastest and slowest execution time respectively.

5.4.2. DWES analysis

Evolution Strategy (ES) is a fast metaheuristic algorithm which only uses mutation within evolutionary process. The proposed

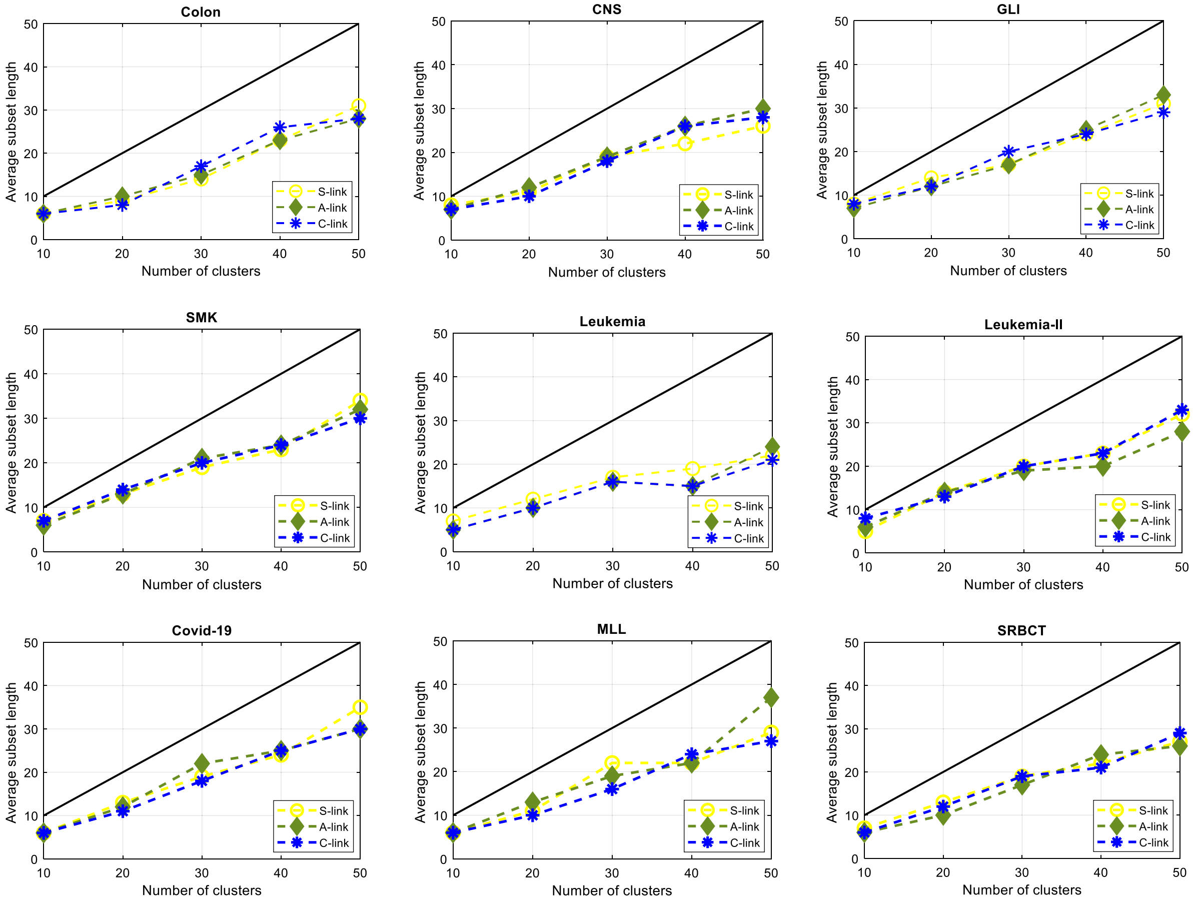

Fig. 7. Average subset length of EMC-DWES with respect to number of clusters in DWES.

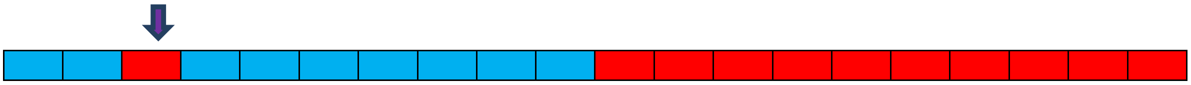

Fig. 8. Sorted response variable based on an arbitrary ascending sorted feature.

DWES in this research uses hierarchical clustering to cluster remaining features calculated by EMC. Hierarchical clustering has various types of clustering with different levels of noise sensitivity in comparison with Kmeans. Thus, DWES in this research is aimed to test different formats of clusterings that hierarchical clustering offers through its linkages rather than just one format of clustering by Kmeans which is also accessible by complete linkage basically (recalling that complete linkage is more robust to noise compared with Kmeans). In addition, although Kmeans is more efficient than hierarchical clustering generally, but 95% of the dimensionality of the input dataset is reduced by EMC. As such, DWES specifies the final feature subset with its embedded hierarchical clustering on the drastically shrinked dataset generated by EMC. Hence, hierarchical clustering does not have any time burden in comparison with Kmeans since quadratic time complexity of hierarchical clustering is not tangible for small datasets. Thus, hierarchical clustering is preferred for clustering instead of Kmeans in DWES this research. Second, the proposed DWES does not select redundant features via clustering the best features recognized by EMC using hierarchical clustering. This happens because those features in the same cluster have similar characteristics and DWES selects one feature within a cluster at a time. Moreover, the proposed DWES even intelligently select more informative clusters by using the concept of weight associated to each cluster so that some clusters would not be selected for feature selection at all. In fact, the concept of cluster weights helps automatic feature selection as well. In addition, not only DWES supports minimum redundancy, but it also applies maximum relevance using classifiers' (SVM and DT) accuracy. It is noteworthy mentioning that further investigation was done on associating dynamic weights to cluster members (features) so that features that increased the accuracy had greater weights accordingly which was not any better than current DWES. Fig. 9 compares the average accuracies of EMC-DWES using complete linkage with EMC-Random selection. Unlike EMC-DWES, EMC-Random selection does not cluster input features. It randomly selects and mutates features generated from EMC as shown in Algorithm 3. Even though EMC-Random selection selects features with maximum relevance (according to the respective accuracy) but does not include minimum redundancy nor supports automatic feature selection. Fig. 9 clearly illustrates that the accuracy achieved by EMC-Random selection with 50 features is considerably less than the accuracy of EMC-DWES with 10 number of clusters (except in Leukemia in which the initial accuracy is very high) which confirms the effectiveness of DWES against random selection. Fig. 9 also shows that EMC-Random selection could successfully increase the initial accuracy in majority of datasets (except in GLI, SMK, Colon with 10 and 20 dimensionality, and Leukemia with 10 dimensionality). This happens because EMC-Random selection randomly selects features from

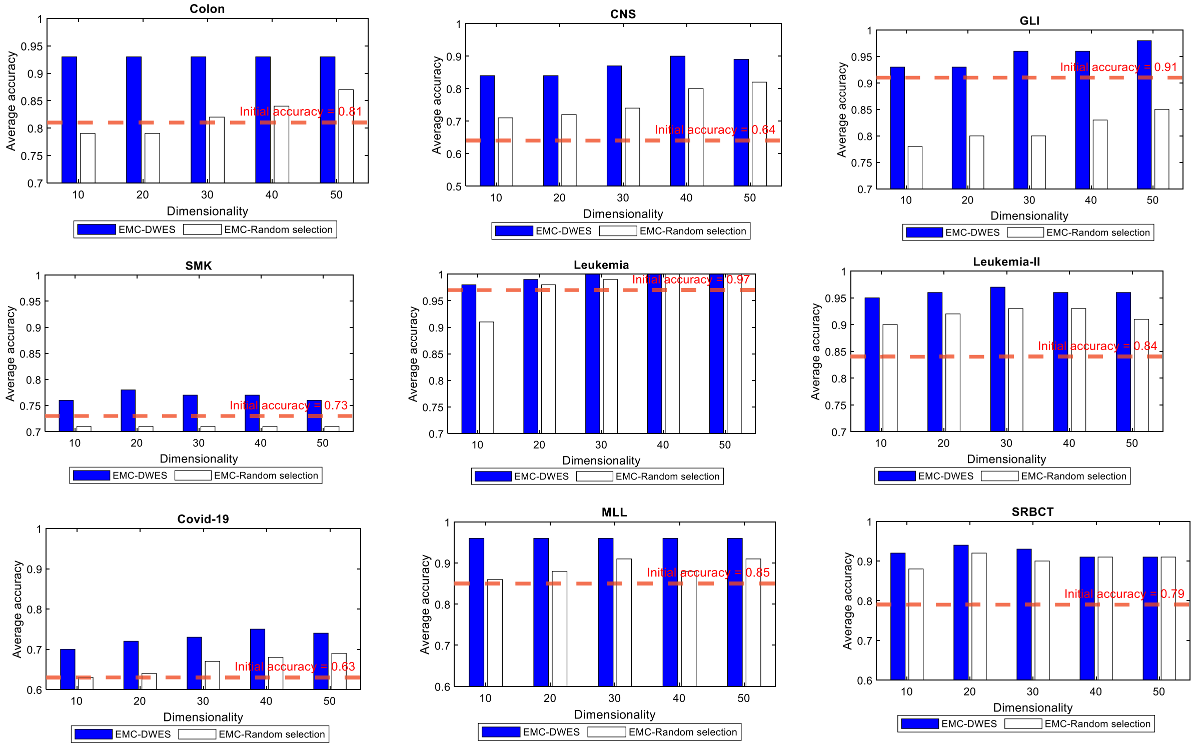

Fig. 9. Accuracy analysis of EMC-DWES and EMC-random selection.

those recognized by EMC. This also confirms the effectiveness of EMC in distinction of informative features.

Algorithm 3. EMC-Random selection

```
Input: $F' = \{F'_1, F'_2, \dots, F'_{m'}\}$
Output: Best_subset
q ← Number of features // q ∈ [10,50] with the step size of 10
Best_subset ← NULL
WHILE stopping condition is not met DO
    Random_subset ← Generate a random subset of size q
    IF Best_subset = NULL THEN
        Best_subset ← Random_subset
    END IF
    Fitness ← Calculate accuracy of Random_subset
    IF Fitness > fitness (Best_subset) THEN
        Best_subset ← Random_subset
    END IF
END WHILE
Return Best_subset
```

5.4.3. Comparison

In the following five recent works are selected from Table 1 for comparison with EMC-DWES in Table 6 so that three of them have frequency-based rankers (MC in WOA-MC, SLI in ATFS, and SLI-γ in $GA_{rank\&rand}$) exactly like EMC-DWES that exploits EMC frequency-based ranker. Likewise, all the state-of-the-art works in Table 6 utilize the concept of metaheuristic algorithms same as EMC-DWES. SL in Table 6 stands for average Subset Length and Acc denotes average accuracy while NA means Not Applicable.

EMC-DWES Vs WOA-MC: Whale Optimization Algorithm-Mutual Congestion (WOA-MC) is the base paper in which the current research is inspired from. WOA-MC is a hybrid feature selection for binary datasets consists of two filter methods in which the final subset was determined manually. According to Table 6 the accuracy of EMC-DWES is greater than WOA-MC in binary datasets. This superiority is expectable since EMC-DWES contains a wrapper section and the proposed EMC is the extended version of MC.

EMC-DWES Vs ATFS: Automatic Thresholding Feature Selection (ATFS) is an ensemble automatic feature selection method. The method is based on ensembling three rankers (Relief F, Mutual Congestion (MC), and Sorted Label Interference (SLI)) using the concept of fast non-dominated sorting. SLI is another extension of MC. The results in Table 6 shows that EMC-DWES outperforms ATFS as well in all binary datasets except in SMK in which both achieved same accuracies.

EMC-DWES Vs $GA_{rank\&rand}$: $GA_{rank\&rand}$ is a hybrid method which combined SLI-γ (also inspired from MC) as a filter method with genetic algorithm as wrapper method. SLI-γ is a filter ranker that only passes the best 1% of the features to genetic algorithm to generate initial population from. $GA_{rank\&rand}$ also uses some random features besides those best ranked by SLI-γ for generating initial population as well. The results calculated in Table 6 for $GA_{rank\&rand}$ are based on a K-Nearest Neighbors fitness function with k = 1. The best solution found from $GA_{rank\&rand}$ is then passed to SVM with linear kernel for accuracy calculation. The results in Table 6 reveal that EMC-DWES outperforms $GA_{rank\&rand}$ in all binary datasets both in subset length and accuracy so that this superiority is completely obvious in accuracy of Colon and subset length of GLI and SMK.

EMC-DWES Vs EIT-bBOA: Ensemble Information Theory-based binary Butterfly Optimization Algorithm (EIT-bBOA) is a hybrid method for both binary and multi-label datasets including the filter Minimal Redundancy-Maximal New Classification Information (MR-MNCI) and a wrapper Information Gain binary Butterfly Optimization Algorithm (IG-bBOA). The results in Table 6 shows that EIT-bBOA achieves greater accuracies only in SMK and Covid-19 (this probably happens because EMC was not as distinctive in SMK and Covid-19 as it was for rest of the datasets). However, EMC-DWES automatically detects smaller effective feature subsets in comparison with EIT-bBOA in which the length of the feature subset is manually limited to 30.

Table 6
Comparison with state-of-the-art works.

Dataset	WOA-MC [13]		ATFS [11]		$GA_{rank\&rand}$ [23]		EIT-bBOA [14]		MPSONC [5]		EMC-DWES	
	SL	Acc	SL	Acc	SL	Acc	SL	Acc	SL	Acc	SL	Acc
Colon	10	0.90	14	0.85	11	0.82	30	0.86	18	0.85	6	**0.93**
CNS		0.80	29	0.80	29	0.88		0.84	23	0.71	28	**0.90**
GLI		0.92	52	0.94	98	0.97		0.84	28	0.95	29	**0.98**
SMK		0.71	38	0.78	106	0.76		**0.82**	35	0.75	14	0.78
Leukemia		0.98	37	0.98	29	0.99		0.89	45	0.98	16	**1**
Leukemia-II	NA		NA		NA			0.96	53	0.89	20	**0.97**
Covid-19								**0.93**	42	0.86	25	0.75
MLL								0.92	45	0.90	6	**0.96**
SRBCT								**0.94**	16	0.83	12	**0.94**

EMC-DWES Vs MPSONC: Multi-objective PSO algorithm and Node Centrality (MPSONC) is a three-phase filter model in which the features are represented by an undirected weighted graph to calculate the feature popularity using node centrality. The node centrality is then used to generate the initial population of multi-objective PSO. The multi-objective PSO uses statistical properties of data instead of a learning model. Table 6 shows that EMC-DWES outperforms MPSONC (except in Covid-19). The superiority of EMC-DWES against MPSONC in accuracy achievement is considerable in Colon, CNS, Leukemia-II, MLL and SRBCT. Likewise, the accuracy of EMC-DWES is competitively better than MPSONC in GLI, Leukemia. It should be noted that EMC-DWES achieves the smaller subset length in all datasets except in CNS and GLI. In addition, EMC-DWES has less parameters need to be cross validated by the user and thus is more applicable in situations where the construction of the learning model has a high computational complexity.

It can be concluded that the combination of EMC and DWES improves the accuracy while retains the small feature length in majority of datasets. The EMC ranker is a reliable filter in recognizing informative features which can be confirmed by its great discarding rate. DWES is then fed with the features selected by EMC to find final feature subset through minimum redundancy (hierarchical clustering) and maximum relevance (learning models).

6. Conclusion

This paper proposes an automatic hybrid feature selection method containing a filter EMC as a frequency-based method and DWES as an efficient wrapper method. EMC sorts the features and then discards most of the features to feed DWES with the most informative features. Then, DWES automatically selects the feature subset by applying minimum redundancy and maximum relevance. In brief, EMC-DWES has three main advantages. First, it has fewer hyper-parameters to be tunned compared with many of the same research. EMC-DWES has two hyper-parameters that should be determined by the user which can be investigated using cross validation namely, linkage type and number of clusters in DWES. Second, EMS-DWES is computationally efficient. Based on the time complexity of EMC, the proposed method (EMC-DWES) has an acceptable performance when the number of features excessively exceeds the number of observations as in high dimensional datasets. Increasing the number of labels (in multi-label datasets) and observations slow the EMC calculation though. Likewise, DWES is fast since it only uses mutation operator. Third, EMC-DWES generally records higher accuracies (accuracy, precision, recall, and Fscore) on benchmark datasets with satisfactorily small subset length in comparison with state-of-the-art. Furthermore, two investigations are conducted on the relation between the accuracy achieved by EMC-DWES and number of clusters as well as the subset length of EMC-DWES and number of clusters. In addition, the superiority of DWES is also argued by comparing the accuracy of EMC-DWES against EMC-Random selection. The results reveal the superiority of proposed method against state-of-the-art works. All in all, this research generally concludes that satisfying feature selection results can be achieved by EMC-DWES on high dimensional medical datasets. However, the main limitation of EMC-DWES is the high computational time of EMC on multi-label datasets with too many labels or observations. Thus, future research will look into how to propose new frequency-based rankers to recognize best features more efficiently and achieve more accuracies.

CRediT authorship contribution statement

Hossein Nematzadeh: Writing – original draft, Writing – review & editing, Conceptualization, Methodology, Software, Validation, Investigation, Resources. **José García-Nieto:** Supervision, Project administration, Validation. **Ismael Navas-Delgado:** Supervision, Project administration, Validation. **José F. Aldana-Montes:** Supervision, Project administration, Validation, Funding acquisition.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Code and data are available on online-supplement to be used as a supplementary in paper as well as code and datasets in https://github.com/KhaosResearch/EMC-DWES (mentioned in abstract)

Acknowledgments

This work has been partially funded by the Spanish Ministry of Science and Innovation via Grant PID2020-112540RB-C41 (AEI/FEDER, UE) and Andalusian PAIDI program with grant P18-RT-2799. It is also granted by the LifeWatch-ERIC initiative ENVIRONMENTAL AND BIODIVERSITY CLIMATE CHANGE LAB (EnBiC2Lab). Funding for open access charge: Universidad de Málaga / CBUA.

Appendix A. Supplementary data

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.asoc.2022.109699. The implemented python code and the datasets related to this article are available as supplementary data.

References

[1] M. Rostami, K. Berahmand, S. Forouzandeh, A novel method of constrained feature selection by the measurement of pairwise constraints uncertainty, J. Big Data 7 (1) (2020) 83.

[2] K.-C. Lin, J.C. Hung, J.-t. Wei, Feature selection with modified lion's algorithms and support vector machine for high-dimensional data, Appl. Soft Comput. 68 (2018) 669–676.

[3] M. Rostami, S. Forouzandeh, K. Berahmand, M. Soltani, M. Shahsavari, M. Oussalah, Gene selection for microarray data classification via multi-objective graph theoretic-based method, Artif. Intell. Med. 123 (2022) 102228.

[4] M. Afshar, H. Usefi, High-dimensional feature selection for genomic datasets, Knowl.-Based Syst. 206 (2020) 106370.

[5] M. Rostami, S. Forouzandeh, K. Berahmand, M. Soltani, Integration of multi-objective PSO based feature selection and node centrality for medical datasets, Genomics 112 (6) (2020) 4370–4384.

[6] H. Lyu, M. Wan, J. Han, R. Liu, C. Wang, A filter feature selection method based on the maximal information coefficient and Gram–Schmidt orthogonalization for biomedical data mining, Comput. Biol. Med. 89 (2017) 264–274.

[7] O. Tarkhaneh, T.T. Nguyen, S. Mazaheri, A novel wrapper-based feature subset selection method using modified binary differential evolution algorithm, Inform. Sci. 565 (2021) 278–305.

[8] M. Mafarja, S. Mirjalili, Whale optimization approaches for wrapper feature selection, Appl. Soft Comput. 62 (2018) 441–453.

[9] A. Jiménez-Cordero, J.M. Morales, S. Pineda, A novel embedded min–max approach for feature selection in nonlinear support vector machine classification, European J. Oper. Res. 293 (1) (2021) 24–35.

[10] M. Lu, Embedded feature selection accounting for unknown data heterogeneity, Expert Syst. Appl. 119 (2019) 350–361.

[11] S. Abasabadi, H. Nematzadeh, H. Motameni, E. Akbari, Automatic ensemble feature selection using fast non-dominated sorting, Inf. Syst. 100 (2021) 101760.

[12] C.-F. Tsai, Y.-T. Sung, Ensemble feature selection in high dimension, low sample size datasets: parallel and serial combination approaches, Knowl.-Based Syst. 203 (2020) 106097.

[13] H. Nematzadeh, R. Enayatifar, M. Mahmud, E. Akbari, Frequency based feature selection method using whale algorithm, Genomics 111 (6) (2019) 1946–1955.

[14] Z. Sadeghian, E. Akbari, H. Nematzadeh, A hybrid feature selection method based on information theory and binary butterfly optimization algorithm, Eng. Appl. Artif. Intell. 97 (2021) 104079.

[15] N. Singh, P. Singh, A hybrid ensemble-filter wrapper feature selection approach for medical data classification, Chemometr. Intell. Lab. Syst. 217 (2021) 104396.

[16] Z. Zhang, Y. Tian, L. Bai, J. Xiahou, E. Hancock, High-order covariate interacted lasso for feature selection, Pattern Recognit. Lett. 87 (2017) 139–146.

[17] H. Zhou, J. Zhang, Y. Zhou, X. Gou, Y. Ma, A feature selection algorithm of decision tree based on feature weight, Expert Syst. Appl. 164 (2021) 113842.

[18] M. Joodaki, M.B. Dowlatshahi, N.Z. Joodaki, An ensemble feature selection algorithm based on PageRank centrality and fuzzy logic, Knowl.-Based Syst. 233 (2021) 107538.

[19] M. Alirezanejad, R. Enayatifar, H. Motameni, H. Nematzadeh, Heuristic filter feature selection methods for medical datasets, Genomics 112 (2) (2020) 1173–1181.

[20] F. Vafaee Sharbaf, S. Mosafer, M.H. Moattar, A hybrid gene selection approach for microarray data classification using cellular learning automata and ant colony optimization, Genomics 107 (6) (2016) 231–238.

[21] S. Azadifar, A. Ahmadi, A graph-based gene selection method for medical diagnosis problems using a many-objective PSO algorithm, BMC Med. Inform. Decis. Mak. 21 (1) (2021) 333.

[22] F. Amini, G. Hu, A two-layer feature selection method using genetic algorithm and elastic net, Expert Syst. Appl. 166 (2021) 114072.

[23] S. Abasabadi, H. Nematzadeh, H. Motameni, E. Akbari, Hybrid feature selection based on SLI and genetic algorithm for microarray datasets, J. Supercomput. (2022).

[24] A.K. Varshney, P.K. Muhuri, Q.M. Danish Lohani, PIFHC: The probabilistic intuitionistic fuzzy hierarchical clustering algorithm, Appl. Soft Comput. 120 (2022) 108584.

[25] M. Negnevitsky, Artificial Intelligence: A Guide to Intelligent Systems, second ed., Addison-Wesley, 2005.

[26] O.S. Ajani, R. Mallipeddi, Adaptive evolution strategy with ensemble of mutations for reinforcement learning, Knowl.-Based Syst. 245 (2022) 108624.

[27] G. James, D. Witten, T. Hastie, R. Tibshirani, An Introduction to Statistical Learning : With Applications in R, Springer, New York, 2013, [2013] © 2013.

[28] P. Qiu, Z. Niu, TCIC_FS: Total correlation information coefficient-based feature selection method for high-dimensional data, Knowl.-Based Syst. 231 (2021) 107418.

[29] G. Manikandan, S. Abirami, An efficient feature selection framework based on information theory for high dimensional data, Appl. Soft Comput. 111 (2021) 107729.

[30] E. Mick, J. Kamm, A.O. Pisco, K. Ratnasiri, J.M. Babik, G. Castaneda, J.L. Derisi, A.M. Detweiler, S.L. Hao, K.N. Kangelaris, G. Renuka Kumar, L.M. Li, S.A. Mann, N. Neff, P.A. Prasad, P. Hayakawa Serpa, S.J. Shah, N. Spottiswoode, M. Tann, C.S. Calfee, S.A. Christenson, A. Kistler, C. Langelier, Upper airway gene expression reveals suppressed immune responses to SARS-CoV-2 compared with other respiratory viruses, Nature Commun. 11 (1) (2020) 5854.

[31] J. Lee, I.Y. Choi, C.-H. Jun, An efficient multivariate feature ranking method for gene selection in high-dimensional microarray data, Expert Syst. Appl. 166 (2021) 113971.