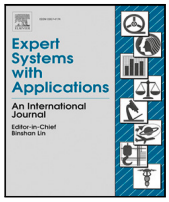
Expert Systems With Applications 249 (2024) 123521

Contents lists available at ScienceDirect

Expert Systems With Applications

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



Pattern recognition frequency-based feature selection with multi-objective discrete evolution strategy for high-dimensional medical datasets

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

[a] ITIS Software, Universidad de Málaga, Arquitecto Francisco Peñalosa 18, Málaga, 29071, Spain
[b] Departamento de Lenguajes y Ciencias de la Computación, Universidad de Málaga, Málaga, Spain
[c] Biomedical Research Institute of Málaga (IBIMA), Universidad de Málaga, Málaga, Spain

ARTICLE INFO

Keywords:
Feature selection
High-dimensional datasets
Filter
Wrapper
Multi-objective optimization

ABSTRACT

Feature selection has a prominent role in high-dimensional datasets to increase classification accuracy, decrease the learning algorithm computational time, and present the most informative features to decision-makers. This paper proposes a two-stage hybrid feature selection for high-dimensional medical datasets: Maximum Pattern Recognition - Multi-objective Discrete Evolution Strategy (MPR-MDES). MPR is a rapid filter ranker that significantly outperforms existing frequency-based rankers in recognizing non-linear patterns, effectively eliminating a majority of non-informative features. Then, the wrapper Multi-objective Discrete Evolution Strategy (MDES) uses the remaining features and obtains sets of solutions which are automatically presented to decision-makers. The experiments conducted on large medical datasets demonstrate that MPR-MDES achieves considerable improvements compared to state-of-the-art methods, in terms of both classification accuracy and dimensionality reduction. In this sense, the proposal successfully performs when presenting informative feature sets to decision-makers. The implementation is available on https://github.com/KhaosResearch/MPR-MDES.

1. Introduction

Feature selection covers a wide area in data mining (Forouzandeh, Aghdam, Forouzandeh, & Xu, 2020; Forouzandeh, Berahmand, Sheikhpour, & Li, 2023) and contains different techniques depending on areas of usage, including classification (Chaudhuri & Sahu, 2022; Sheikhpour, Berahmand, & Forouzandeh, 2023), regression (Amini & Hu, 2021), and time series (Niu, Wang, Lu, Yang, & Du, 2020) data. Many methods exist for feature selection of high-dimensional medical classification data (Abasabadi, Nematzadeh, Motameni, & Akbari, 2021, 2022; Nematzadeh, Enayatifar, Mahmud, & Akbari, 2019; Nematzadeh, García-Nieto, Navas-Delgado, & Aldana-Montes, 2022). High-dimensional data refer to those data in which the number of columns is excessively greater than the number of observations, such as medical gene expression or microarray datasets (Mohapatra, Chakravarty, & Dash, 2016; Sánchez-Maroño, Fontenla-Romero, & Pérez-Sánchez, 2019). Having this huge number of features relative to few observations decreases the classification accuracy of supervised learning algorithms while increasing the learning time. In such cases, feature selection has three advantages: First, it increases classification accuracy. Second, it decreases the classification time the learning algorithm spends. Third, it introduces informative features to the decision-makers. Generally, feature selection techniques are categorized into five groups: filter, wrapper, embedded, ensemble, and hybrid (Osama, Shaban, & Ali, 2023). Filter methods (Kushal et al., 2021) are feature selection methods that evaluate features based on their intrinsic properties or their relevance to the target variable without considering the performance of a specific predictive model. Filter methods may use statistical measures or simple learning algorithms (such as regression to calculate coefficients) to rank or select features, but they do not use the same learning algorithm as the one used for prediction; thus, they are fast but more likely not that accurate. In contrast, wrapper methods (Sahebi, Movahedi, Ebrahimi, Pahikkala, Plosila, & Tenhunen, 2020) evaluate features based on a specific predictive model's performance. Thus, they are slower but more accurate than filters. Embedded methods (Jiménez-Cordero, Morales, & Pineda, 2021) usually refer to methods in which the feature selection process is part of the learning algorithm and is carried out during the training process, such as Lasso regression and decision trees. Ensemble methods (Abasabadi et al., 2021; Hashemi, Mohammad, & Nezamabadi-pour, 2021) refers to a collection of individual feature selection methods that are combined to produce a single

* Corresponding author at: ITIS Software, Universidad de Málaga, Arquitecto Francisco Peñalosa 18, Málaga, 29071, Spain.
E-mail addresses: hnematzadeh@uma.es, hn_61@yahoo.com (H. Nematzadeh), jnieto@uma.es (J. García-Nieto), jfaldana@uma.es (J.F. Aldana-Montes), ismael@uma.es (I. Navas-Delgado).

https://doi.org/10.1016/j.eswa.2024.123521
Received 20 August 2023; Received in revised form 16 January 2024; Accepted 16 February 2024
Available online 17 February 2024

feature selection method. Finally, hybrid methods (Ganjei & Boostani, 2022; Nematzadeh et al., 2019, 2022; Wei, Zhao, Feng, He, & Yu, 2020) usually refer to the process of using multiple feature selection methods in a sequential manner. Hybrids are expected to achieve better results by taking advantage of the positive aspects of other methods.

Data exist in different shapes (i.e. concave or convex, linear or non-linear, etc), and there is not one best feature selection method for all cases. As such, the literature contains diverse research, so each attempt approaches the problem differently to achieve better results. This paper represents an improvement upon frequency-based rankers (Abasabadi et al., 2021, 2022; Nematzadeh et al., 2019), and goes beyond the Extended Mutual Congestion-Discrete Weighted Evolution Strategy (EMC-DWES) (Nematzadeh et al., 2022). In the following Section, the existing gaps and paper contributions are described. A thorough investigation is done while studying the related works. Mutual Congestion (MC) (Nematzadeh et al., 2019), Sorted Label Interference (SLI) (Abasabadi et al., 2021), and Sorted Label Interference-γ (SLI-γ) (Abasabadi et al., 2022) are among the fast frequency-based rankers, but do not count non-linearity, and they are only applicable on classification datasets with a two-label response variable. In contrast, Extended Mutual Congestion (EMC) is a frequency-based ranker that deals better with non-linearity and can be applied to datasets with a multi-label response variable. However, the label constraint adds to the time complexity generally. This inspires the development of an efficient frequency-based ranker that can be applied to multi-label datasets with better ranking capabilities. In the terminology of this paper, the term label is used interchangeably with class. When we say that an observation has label A, it means that it belongs to class A. Therefore, by multi-label, we specifically refer to datasets where each instance is assigned to one and only one label, and in this context, label is equivalent to class.

Additionally, the Discrete Weighted Evolution strategy (DWES) is a wrapper feature selection method used within a hybrid method in Nematzadeh et al. (2022) along with EMC. DWES has many hyperparameters to be tuned by the user, and it selects the final set of features using a single-objective optimization. This, in turn, inspires the development of a new wrapper feature selection method with fewer hyperparameters that could simultaneously present multiple feature subsets to the expert for decision-making instead of presenting just one feature subset. In summary, the contributions of this research with the aim of feature selection in microarray medical datasets are:

1. To propose Maximum Pattern Recognition (MPR) as an efficient filter frequency-based ranker that offers a more effective approach for handling non-linearity and applies to multi-label classification datasets.
2. To propose a wrapper Multi-objective Discrete Evolution Strategy (MDES) that automatically selects optimal subsets of features, while maintaining a suitable level of accuracy and improves upon the work done in Nematzadeh et al. (2022).
    - MDES clusters features using Kmeans, unlike DWES, which uses hierarchical clustering. The use of hierarchical clustering in DWES imposes many hyperparameter tuning, including the linkage type eliminated in MDES. In addition, MDES does not need to investigate the best number of clusters which is another hyperparameter in DWES. MDES always looks for the best features within 10 clusters.
    - MDES utilizes Roulette Wheel Selection without replacement (RWS) to guarantee the selection of minimum redundant maximum relevant features.
    - MDES is a multi-objective approach to evolution strategy, while DWES follows a single-objective approach.
3. To combine MPR with MDES to construct a hybrid feature selection method.

The rest of this paper is organized as follows. Section 2 describes the related works on frequency-based filter rankers and recent feature selection methods. Section 3 explains how existing frequency-based rankers evaluate the separability of a specific feature with a pilot example. Additionally, the algorithm of (1+1) Evolution Strategy ((1+1) ES) is shown as well, and it is described how it is going to be improved in MDES in this paper. Section 4 presents the methodology of MPR-MDES. Section 5 shows how the results of MPR-MDES improve classification accuracy with further related discussions. Finally, Section 6 concludes the paper with final remarks.

2. Related works

This section initially investigates the history of frequency-based feature selection rankers from the first attempt onwards in Section 2.1. Then, a series of recent feature selection methods have been introduced in Section 2.2. The gaps in the existing methods are identified to be resolved in this paper at the end of Section 2.2.

2.1. History of frequency-based rankers

Frequency-based rankers were first introduced by Nematzadeh et al. (2019) by introducing the concept of Mutual Congestion (MC) devised for two-label classification datasets. Later, Abasabadi et al. (2021) proposed Sorted Label Interference (SLI) as a frequency-based ranker inspired by MC. Likewise, Abasabadi et al. (2022) also proposed another variation of SLI called SLI-γ. Recently, Nematzadeh et al. (2022) proposed EMC applicable to multi-label datasets and improved MC's ranking. The details about how MC, SLI, SLI-γ, and EMC differently rank features are described with an example later in Section 3.1. All frequency-based rankers (MC, SLI, SLI-γ, and EMC) belong to the filter category. Table 1 shows the differences between frequency-based rankers regarding applicability on the response variable, time complexity, sensitivity to the response variable, linear assumption of data distribution, and year of publication. Time complexity varies based on the type of sorting algorithm used in frequency-based rankers. As such, time complexities in Table 1 were calculated based on the merge sort algorithm, which has the complexity of $O(n \log n)$. The parameters n, m, and l denote the number of samples, columns, and labels, respectively. Table 1 confirms that EMC theoretically spends more time on added-value services (applicability for multi-label datasets and better rankings for non-linear patterns). Sensitivity to the response variable means the result of a ranker depends on a specific label. This holds for SLI-γ so that it is sensitive to positive labels in two-label datasets. In other words, the calculation in SLI-γ mostly depends and has a bias to one of the two labels, highlighting a clear limitation of SLI-γ. Moreover, MC, SLI, and SLI-γ assume linear data distribution in their formulation. In contrast, EMC demonstrates better recognition of non-linear patterns (although it indeed needs improvement), but the time complexity is a theoretical overhead particularly if l is significant.

2.2. Feature selection in classification problems

This section introduces some recent methods in feature selection of classification problems so that some of these methods use frequency-based rankers presented in Section 2.1. Nematzadeh et al. (2019) proposed Whale Optimization Algorithm-Mutual Congestion (WOA-MC) that utilized MC within a hybrid method. WOA-MC first discarded half of the less informative features using Whale Optimization Algorithm (WOA), sorted the remaining features by MC, and manually selected the 10 best features. Finally, the 10 selected features were used for majority voting in a heuristic forward feature selection approach for accuracy improvement. The improvement was considerable in medical high-dimensional datasets. Later, Abasabadi et al. (2021) proposed Automatic Thresholding Feature Selection (ATFS), utilizing SLI frequency-based ranker inspired by MC. ATFS initially ranked

Table 1
Comparison of frequency-based rankers for classification problems.

Frequency-based rankers	Applicability on response variable	Time complexity	Sensitivity to response variable	Linear assumption of data distribution	Year
Mutual Congestion (Nematzadeh et al., 2019)	Two-label	$O(mn\log n)$	No	Yes	2019
Sorted Label Interference (Abasabadi et al., 2021)	Two-label	$O(mn\log n)$	No	Yes	2021
Sorted Label Interference-γ (Abasabadi et al., 2022)	Two-label	$O(mn\log n)$	Yes	Yes	2022
Extended Mutual Congestion (Nematzadeh et al., 2022)	Multi-label	$O(mn\log n + mln)$	No	No	2022

the features of the high-dimensional datasets based on SLI, MC (Nematzadeh et al., 2019) and ReliefF. Finally, the results of the three rankers were ensembled based on the concept of non-dominated sorting. ATFS was solely applicable to two-label datasets and improved the accuracy of classifiers on medical high and non-high-dimensional datasets. Abasabadi et al. (2022) also proposed another hybrid feature selection method called $GA_{rank\&rand}$. $GA_{rank\&rand}$ combined SLI-γ (another variations of SLI) with Genetic Algorithm (GA). It was clearly shown that $GA_{rank\&rand}$ achieved good accuracy when the initial population of GA is generated using the most relevant features recognized by SLI-γ. $GA_{rank\&rand}$ recorded acceptable results on both two-label high and non-high-dimensional datasets as well. Recently, Nematzadeh et al. (2022) proposed Extended Mutual Congestion-Discrete Weighted Evolution Strategy (EMC-DWES). EMC-DWES was a two-stage method in which EMC frequency-based ranker was used initially to discard 95% of the less informative features. Next, DWES used the remaining features and automatically selected the best features. DWES had two hyperparameters that should be tuned by experts namely, type of linkages in hierarchical clustering (single, average, or complete) as well as number of clusters. Experiments showed that DWES was very fast even though it was a wrapper method and its combination with EMC could increase the accuracy of prediction considerably on high-dimensional medical datasets. In addition to WOA-MC (Nematzadeh et al., 2019), ATFS (Abasabadi et al., 2021), $GA_{rank\&rand}$ (Abasabadi et al., 2022), and EMC-DWES (Nematzadeh et al., 2022) that utilized frequency-based rankers, numerous other attempts have been made for feature selection in classification problems. These attempts have been compiled in Table 2 and are explained accordingly.

Similarly, the literature indicates a strong inclination towards using evolutionary algorithms, either single-objective or multi-objective, for feature selection in datasets with categorical response variables (Gutowski, Schang, Camp, & Abraham, 2022; Nematzadeh et al., 2019). Modified Gray Wolf Optimization (MGWO) (Pan, Chen, & Xiong, 2023) intelligently mixed filters and wrappers for feature selection. MGWO initially ranked the features using filters, including ReliefF algorithm and Copula entropy. The aim was to reduce the search space for large-scale feature selection problems based on correlation measures and avoid having poor quality in the initial population. Additionally, the differential evolution algorithm was also used to expand the search space of the standard GWO. MGWO achieved good results with 10 gene expression microarray datasets. Gutowski et al. (2022) also proposed the Genetic Algorithm with a multi-objective Compass (GAwC) with three objectives, namely: number of features, accuracy, and Area Under the ROC Curve (AUC). GAwC aggregated all three objectives within a single-objective function using the compass concept, where the angle theta corresponds to the trade-off between objectives. As such, GAwC can be recognized in the category of single-objective evolutionary algorithms aggregating three objectives within a single-objective fitness function. GAwC outperformed all other competitive genetic algorithm-based approaches, but it was only appropriate with binary classification.

Hashemi et al. (2021) proposed a Pareto-based Ensemble of Feature Selection (PEFS) as an ensemble method of four filters, namely: Fisher score, Local Learning-based Clustering, Correlation-based Feature Selection, and Maximum Information Coefficient. The ensemble strategy employed by PEFS is analogous to ATFS (Abasabadi et al., 2021), both utilizing non-dominated sorting as the underlying mechanism. The results showed that PEFS was superior in terms of accuracy and Fscore in comparison with other ensemble feature selection methods and basic algorithms. Sadeghian, Akbari, and Nematzadeh (2021) proposed an Ensemble Information Theory-based binary Butterfly Optimization Algorithm (EIT-bBOA), a hybrid method comprising a filter, a wrapper, and an ensemble method. EIT-bBOA utilized Minimal Redundancy-Maximal New Classification Information (MR-MNCI) as a filter method to initially discard 80% of non-relevant features. Then, the Information Gain binary Butterfly Optimization Algorithm (IG-bBOA) was used for best feature subset selection. Finally, EIT-bBOA manually selected the best 30 features using an ensemble of ReliefF and Fisher score. The findings confirmed the efficiency of EIT-bBOA on 6 standard and high-dimensional datasets from the UCI repository. Thaher, Chantar, Too, Mafarja, Turabieh, and Houssein (2022) proposed a Boolean variant of Particle Swarm Optimization boosted with Evolutionary Population Dynamics (BPSO-EPD). BPSO-EPD avoided getting stuck in local optima with its boosted exploration ability and repositioned the worst half of the solutions around the optimal solutions selected from the best half. The experimental results on 22 UCI repository datasets confirmed accuracy improvement and revealed the excellent behavior of EPD strategies in evolving the ability of BPSO. Saadatmand and Akbarzadeh-T (2023) proposed a Set-based Integer-coded Fuzzy granular Evolutionary algorithm (SIFE) that not only utilized fuzzy granulation as a surrogate technique but also extended the idea of fuzzy granulation to encourage diversity in the evolutionary process. SIFE implicitly proposed three-parent UI crossover and complement mutation operations for better exploitation and exploration. The experimental results on 22 real-world classification benchmark datasets showed SIFE robustly selected features with good accuracy. However, SIFE had a high computational cost as the number of samples increased. Asghari, Nematzadeh, Akbari, and Motameni (2023) recently proposed Best Clustering Normalized Mutual Information Quantile with Incremental Association Markov Blanket (BC-NMIQ-IAMB). Initially, the method ranked the best features using the square root threshold with (BC-NMIQ). Then, IAMB was used for automatically optimal feature selection. BC-NMIQ-IAMB increased the accuracy of high-dimensional microarray datasets with the optimal set of selected features.

To summarize, the following are the key takeaways from the related works:

1. Table 1 indicates that no efficient frequency-based ranker can effectively handle the non-linearity of data distribution and multi-label datasets.
2. Table 2 indicates that evolutionary algorithms are among the prominent methods for feature selection.
3. Table 2 also indicates that filter feature selection methods are typically used within a hybrid or ensemble approach, as they often do not significantly improve accuracy when used in isolation.

This paper proposes a hybrid feature selection method that combines filter Maximum Pattern Recognition (MPR) with wrapper Multi-objective Discrete Evolution Strategy (MDES). This hybrid method addresses the gaps uncovered in the related works. Specifically, the proposed MPR-MDES significantly outperforms EMC-DWES (Nematzadeh et al., 2022) in several key aspects. Firstly, MPR exhibits superior computational efficiency compared to EMC and, in certain scenarios, demonstrates enhanced capability in identifying non-linear data distributions. Secondly, MDES employs a multi-objective evolution strategy that enables us to achieve superior feature selection accuracy

Table 2
Recent feature selection methods for classification problems.

Method	Type	Selection procedure	Response variable	Evolutionary-based concept	Objective function	Year
WOA-MC (Nematzadeh et al., 2019)	Hybrid (filter+filter)	Manual	Two-label	Whale optimization algorithm	Single	2019
ATFS (Abasabadi et al., 2021)	Ensemble (3 filters)	Automatic	Two-label	Non-dominated sorting	Multi	2021
EIT-bBOA (Sadeghian et al., 2021)	Hybrid (filter+wrapper + ensemble (2 filters))	Manual	Multi-label	Butterfly optimization algorithm	Single	2021
PEFS (Hashemi et al., 2021)	Ensemble (4 filters)	Manual	Multi-label	Non-dominated sorting	Multi	2021
Min-Max (Jiménez-Cordero et al., 2021)	Embedded	Manual	Two-label	NA	Single	2021
$GA_{rank\&rand}$ (Abasabadi et al., 2022)	Hybrid (filter+wrapper)	Automatic	Two-label	Genetic algorithm	Single	2022
EMC-DWES (Nematzadeh et al., 2022)	Hybrid (filter+wrapper)	Automatic	Multi-label	Evolution strategy	Single	2022
BPSO-TEPD (Thaher et al., 2022)	Wrapper	Automatic	Multi-label	Particle swarm optimization & Evolutionary population dynamics	Single	2022
GAwC (Gutowski et al., 2022)	Wrapper	Automatic	Two-label	Genetic algorithm	Single	2022
SIFE (Saadatmand & Akbarzadeh-T, 2023)	Wrapper	Automatic	Multi-label	Genetic algorithm	Single	2023
MGWO (Pan et al., 2023)	Hybrid (filter+wrapper)	Automatic	Multi-label	Gray wolf optimization algorithm	Single	2023
BC-NMIQ-IAMB (Asghari et al., 2023)	Hybrid (filter+filter)	Automatic	Multi-label	NA	NA	2023

while minimizing the number of features selected compared to DWES. Thirdly, MDES effectively eliminates the need for hyperparameter tuning, a crucial step in the single-objective DWES algorithm. MDES exhibits improved efficiency and accuracy by dispensing with this tuning requirement. Finally, MDES's multi-objective approach allows us to present a set of solutions to decision-makers rather than a single solution provided by the single-objective DWES. This will enable decision-makers to consider and choose from various options, ultimately making a more informed and effective decision.

3. Preliminaries

This section first explains existing frequency-based rankers and compares their results to an example. Then, (1+1) ES is briefly described.

3.1. Different formulations of frequency-based rankers

The first step is common to all existing frequency-based rankers. This is illustrated in Fig. 1 for binary classification so that the values of each feature are sorted ascendingly, and the order of labels (including blue and red labels) in the response variable also changes accordingly in the vector of labels. The feature F1 in Fig. 1 is a non-linear separable feature. Classifiers dealing with non-linearity, such as Multi-Layer Perceptron (MLP) (Li, Wang, Hao, Wang, & Zhang, 2022) can perfectly classify red and blue labels using F1. However, the second step of frequency-based rankers differs from method to method. Generally, Mutual Congestion (MC) (Nematzadeh et al., 2019), Sorted Label Interference (SLI) (Abasabadi et al., 2021), and SLI-γ (Abasabadi et al., 2022) divide the vector of labels into three sections (S1, S2, and S3) for calculation. However, Extended Mutual Congestion (EMC) (Nematzadeh et al., 2022) follows a distinct procedure. MC counts the number of labels within the interference region relative to the entire labels in the vector of labels. An interference region is a region in vector of labels which two labels are not separable. Generally, MC holds that S1 and S3 are the accumulation of separable distinguished labels, but S2 is the interference region in which two labels are not simply separable. As a result, since MC does not count non-linearity, it assumes all non-linear separable five blue labels in Fig. 1 are non-separable. Assuming the ordering in the vector of labels starts with blue in Fig. 1, MC believes that the interference region (S2) starts from the first red label until the last blue label and MC measure is accordingly calculated as shown in Eq. (1), so that m_1^B and m_1^R are the number of separable observations for blue and red labels in S1 and S3, respectively. Likewise, m_2^B and m_2^R are the number of non-separable observations for blue and red labels in the interference region (S2), respectively. This strategy only works well if the distribution of data is linear. MC with the least values are the best and the linear MC measure for Fig. 1 is $\frac{5+7}{3+5+7+0} = 0.8$, which assumes the respective feature does not have separability characteristics although in fact it has.

SLI follows the same instruction as MC does with a slight difference in the calculation as shown in Eq. (2) so that m_1^B and m_2^R are the number of blue and red labels in S1 and S2, respectively. Likewise, m_2^B and m_1^R are the number of blue and red labels in S2 and S3, respectively. SLI for Fig. 1 equals $\frac{(3\times0)-(5\times7)}{\sqrt{(3+5)(3+7)(5+0)(0+7)}} = -0.66$. SLI is a value in [-1,+1], whereby -1 and $+1$ denote the feature as completely non-separable and separable, respectively. As such, SLI assumes the feature F1 in Fig. 1 does not have separability characteristics, although it does. Unlike MC and SLI, SLI-γ does not have a deterministic approach in the calculation. It calculates the fraction of the number of observations in the interference region for a specific label relative to that label's entire number of observations (whether in interference or non-interference region). Therefore, the results of SLI-γ depend on what label will be selected for the basis of calculations. Hence, according to the piecewise function in Eq. (3) and the information in Fig. 1, SLI-γ equals $\frac{7}{0+7} = 1$ for red labels and $\frac{5}{3+5} = 0.63$ for blue ones. Obviously, the less SLI-γ is for a specific feature, the better that feature is for classification. It is important to notice that MC, SLI, and SLI-γ assume S3 = $\emptyset$, since they do not count non-linearity.

In contrast, Extended Mutual Congestion (EMC) has a better approach to dealing with non-linearity. It does not divide the vector of labels into three sections like MC, SLI, and SLI-γ do. However, it investigates the separability of each label within the vector of labels individually using Eq. (4) whereby k is the maximum number of labels (in our example in Fig. 1 k equals 2), m_{r_i} is the number of non-separable labels for label r_i, and θ_{r_i} is the summation of the number of both non-separable and separable labels for label r_i. Thus, EMC for Fig. 1 equals $\frac{7+0}{7+8+0+7} = 0.32$. EMC assigns less values to better features for classification and, in comparison with MC, SLI, and SLI-γ has a more realistic recognition for feature F1 in Fig. 1. Therefore, the feature F1 in Fig. 1 is $1-0.32 = 0.68$ good for classification based on EMC. However, a good ranker should recognize F1 as a perfect feature with separability power.

$$MC = \frac{m_2^B + m_2^R}{m_1^B + m_2^B + m_2^R + m_1^R} \tag{1}$$

$$SLI = \frac{(m_1^B \times m_1^R) - (m_2^B \times m_2^R)}{\sqrt{(m_1^B + m_2^B)(m_1^B + m_2^R)(m_2^B + m_1^R)(m_1^R + m_2^R)}} \tag{2}$$

$$SLI-\gamma = \begin{cases} \frac{m_2^R}{m_1^R + m_2^R}, & for\ reds \\ \frac{m_2^B}{m_1^B + m_2^B}, & for\ blues \end{cases} \tag{3}$$

$$EMC = \frac{\sum_{i=1}^{k} m_{r_i}}{\sum_{i=1}^{k} \theta_{r_i}} \tag{4}$$

In conclusion, EMC deals better with non-linearity and achieves more realistic results than MC, SLI, and SLI-γ. Moreover, EMC is applicable for multi-label response variables but still has high computational time based on Table 1. One of the main contributions of

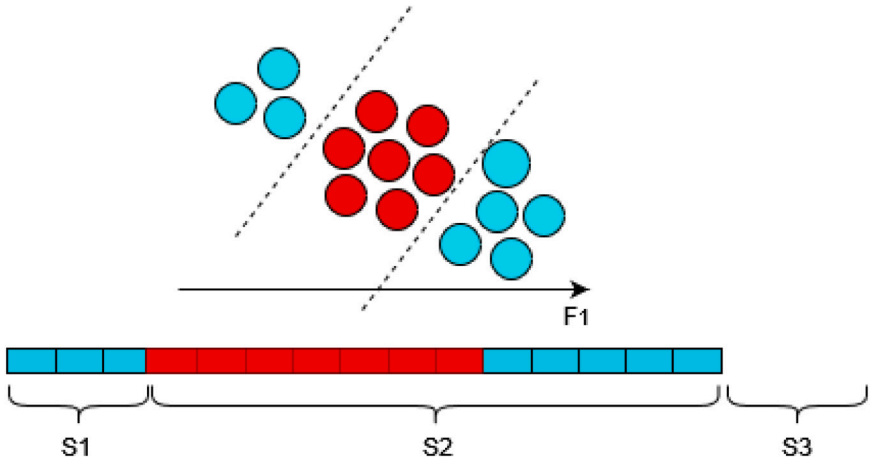


Fig. 1. An example that shows how existing frequency-based rankers calculate the separability of feature F1. The values of F1 are sorted ascendingly in advance, and the vector of labels (containing blue and red labels) is already reordered according to the ascendingly sorted feature F1.

this research is the proposal of a new frequency-based ranker, Maximum Pattern Recognition (MPR), that is aligned with the existing research (Abasabadi et al., 2021, 2022; Nematzadeh et al., 2019, 2022) in Table 1. MPR not only handles non-linearity better but also has a faster computational time for multi-label classification datasets.

3.2. (1+1) EA

Natural selection acts on a collection of genes, not on a single gene in isolation. Thus, evolution strategies mutate all of the genes in the chromosome simultaneously. Evolution strategies can solve many constrained and unconstrained optimization problems and outperform many highly complex existing optimization techniques. Experiments also reveal that the most straightforward evolution strategy, namely, single parent - single offspring or (1+1) EA works best (Negnevitsky, 2005). Unlike Genetic Algorithm (GA), (1+1) EA is fast because it only has the mutation operator. Likewise, (1+1) EA does not need to represent the problem in the coded form as GA does. However, the selection of a meta-heuristic algorithm is always application-dependent. It is important to notice that (1+1) EA is continuous in essence, like most other meta-heuristic algorithms, as shown in line 5 of Algorithm 1 in the offspring generation. Algorithm 1 shows the implementation steps of (1+1) continuous EA.

It was experimentally shown that formulation of the feature selection problem using a wrapper single-objective discrete (1+1) EA is fast and accurate. This paper aims to improve the method introduced in Nematzadeh et al. (2022), specifically the Discrete Weighted Evolution Strategy (DWES), by proposing a new method called the Multi-Objective Discrete Evolution Strategy (MDES), which utilizes the Roulette Wheel Selection (RWS) of initial solutions. Compared to DWES, MDES offers several advantages. First, it has fewer hyperparameters for determination by decision-makers. Second, it presents a set of solutions for decision-makers instead of only one solution that single-objective DWES provides. Decision-makers then choose one of these solutions to use in the problem domain. Third, it increases the accuracy of classification.

4. Proposed method

Maximum Pattern Recognition-Multi-objective Discrete Evolution Strategy (MPR-MDES) is a hybrid feature selection method proposed in this paper. To start with, Maximum Pattern Recognition (MPR) is proposed as a filter-supervised frequency-based ranker. It is fast and also does not assumne linear distribution of data in its formulation. MPR discards the majority of non-informative features and the remaining features are then passed to the wrapper Multi-objective Discrete Evolution Strategy (MDES), which automatically presents informative sets of features to the expert. MDES clusters the features initially and intelligently assigns more weights to better clusters. Thus, it tends to select features from informative clusters. Fig. 2 illustrates the overview of the methodology used in this paper.

Algorithm 1 (1+1) EA

Input: problem: an optimization problem
Output: parent
1: Represent the problem by N parameters and their ranges: $(X_{1_{min}}, X_{1_{max}}), (X_{2_{min}}, X_{2_{max}}), ..., (X_{N_{min}}, X_{N_{max}})$
2: Create the parent by selecting an initial value for each parameter: $x_1, x_2, ..., x_N$
3: Calculate the fitness of the parent: $X = f(x_1, x_2, ..., x_N)$
4: **while** stopping criterion has not been reached **do**
5: Create the offspring by adding a normally distributed random value (a) with mean zero and deviation (δ): $x'_i = x'_i + a(0, \delta), \quad i = 1, 2, ..., N$
6: Calculate the fitness of the offspring: $X' = f(x'_1, x'_2, ..., x'_N)$
7: **if** $X' > X$ **then**
8: $X \leftarrow X'$
9: *parent* $\leftarrow$ *offspring*
10: **end if**
11: **end while**
12: Return *parent*

4.1. Maximum pattern recognition

Maximum Pattern Recognition (MPR) is a supervised filter that ranks the features of a dataset based on the level of separability each one presents. Let X be a high-dimensional dataset in Eq. (5) with n rows and m features. Therefore, any observation in X can be shown through Eq. (6). Accordingly, observation notations in Eq. (7) can also show each dataset feature.

$$X = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1m} \\ x_{21} & x_{22} & \cdots & x_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n1} & x_{n2} & \cdots & x_{nm} \end{bmatrix} \tag{5}$$

$$x_i = (x_{i1},\ x_{i2},\ \cdots,\ x_{im}), \quad i = 1,\ 2,\ \cdots,\ n \tag{6}$$

$$f_j = (x_{1j},\ x_{2j},\ \cdots,\ x_{nj})^T, \quad j = 1,\ 2,\ \cdots,\ m \tag{7}$$

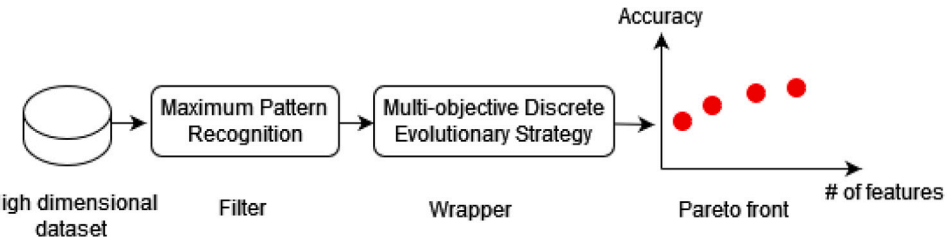


Fig. 2. MPR-MDES workflow overview.

The set of observations with similar labels can be defined as y^{l_p} in Eq. (8) in which l is the maximum number of labels.

$$y^{l_p} = (x_1^{l_p},\ x_2^{l_p},\ \cdots,\ x_{n_p}^{l_p}) = \{x_j^{l_p}\}_{j=1}^{n_p},\quad p = 1,\ 2,\ \cdots,\ l \tag{8}$$

where:

$$\bigcup_{p=1}^{l} y^{l_p} = X,\quad \sum_{p=1}^{l} n_p = n$$

MPR sorts the individual features initially and reorders the response variable (vector of labels) accordingly, as in Fig. 1, like existing frequency-based rankers. However, MPR follows a distinct procedure for ranking features and considers non-linearity. As such, let m_p be the maximum number of consecutive appearances (pattern) for label l_p and n_p is the entire number of observations corresponding to label l_p. MPR calculates the ρ_j value in Eq. (9) for each feature using the information in the vector of labels. Therefore, ρ_j for the example in Fig. 1 equals $\sqrt{\frac{5}{8}\times\frac{7}{7}} = \sqrt{\frac{35}{56}} = 0.79$ which completely outperforms existing frequency-based rankers by assigning more realistic value to the feature. ρ_j is a value in $[0, 1]$ and shows how separable an individual feature is. Therefore, the greater ρ_j is, the more separable that feature is.

$$\rho_j = \left\{\sqrt{\prod_{p=1}^{l}\frac{m_{p_j}}{n_p}}\right\}_{j=1}^{m} \tag{9}$$

Algorithm 2 Maximum Pattern Recognition

Input: X: a dataset including features and the response variable
Output: ρ

```
1: m = len(X.columns) − 1
2: ρ = zeros(m)
3: yCol = len(X.columns)
4: labels = len(unique(X[:,yCol]))
5: a = {};  c = 1;
6: for i=1 to labels do
7:     key = c
8:     value = sum(X[:,yCol] == labels[i])
9:     a[key] = value
10:    c = c + 1
11: end for
12: for j=1 to m do
13:    df = X.sort_values(j)
14:    z = Calculate number of consecutive patterns for df[:, yCol]
15:    max_pattern = {}
16:    max_pattern = Calculate the maximum pattern for each label in
       z
17:    d = 1
18:    for i=1 to labels do
19:        d=d × max_pattern[i] / a[i]
20:    end for
21:    ρ[j] = √d
22: end for
23: Return ρ
```

Algorithm 2 shows the steps of MPR, where m denotes the number of features in X and ρ is an empty array initialized to record the results (multiple instructions in a line are separated by a semicolon, as illustrated in line 5). Additionally, *yCol* is the column index of the response variable within X and *labels* denotes the number of unique categories (labels) in X. Lines 5–11 calculates the n_p in Eq. (8). Lines 12–22 are the main body of MPR and show how it is calculated so that each feature is ascendingly sorted in line 13 and the vector of labels (response variable) is reordered accordingly. Line 14 calculates distinguished numbers of consecutive appearances of each label in the vector of labels (patterns). Lines 15 and 16 store maximum pattern values for each label in a dictionary variable called max_pattern. Lines 17–22 calculate the MPR measure, ρ_j, for a specific feature with $max\_pattern[i]$ and $a[i]$ corresponding to m_{p_j} and n_p in Eq. (9), respectively. Finally, ρ in line 23 is an array of size m that contains the calculated MPR measure for each feature. The proposed methodology only keeps and passes the best 20 features recognized by MPR to the Multi-objective Discrete Evolution Strategy (MDES) and discards the remaining features. As such, the dataset X with size of $n \times m$ in Eq. (5) will be converted to dataset X' with size of $n \times m'$ ($m' = 20$). It is evident that $m' \ll m$, especially for high-dimensional datasets.

4.2. Multi-objective discrete evolution strategy

Algorithm 3 clearly shows the Multi-objective Discrete Evolution Strategy (MDES) steps. In this algorithm, the individual solution is a struct containing six fields, namely: Data (corresponding dataset), Fit (vector of objective functions), Rank (front number), CD (crowding distance value), s_p (domination set), and n_p (dominated count).

MDES divides the selected features from MPR ($F' = \{f'_1, f'_2, \ldots, f'_{m'}\}$) into q numbers of clusters in Eq. (10). This implicitly means that the best sets of features selected by MDES within the Pareto front could finally have between 1 and q features because MDES selects at most one feature for each cluster. Algorithm 3, generally formulates MDES using the variable q, while in our experiment, we set q to 10. The variable q also reflects the population size, nPop.

$$C^q = Kmeans(F', q) = c_1, c_2, \ldots, c_q \tag{10}$$

where: $c_i = \{f_1'^{(i)}, f_2'^{(i)}, \ldots, f_{s_i}'^{(i)}\}, s_i = |c_i|, \bigcup_{i=1}^{q} c_i = F' = X' \cdot T$

MDES assigns initial weights to each cluster of C^q using Eq. (11) and line 2 of Algorithm 3. These weights are then used in Algorithm 4 (Roulette Wheel Selection without replacement (RWS), which can alternatively be implemented using weighted random sampling). At the end of each iteration, the weights are updated based on Eq. (12) provided that the maximum accuracy in the Pareto front exceeds the existing best accuracy (In Algorithm 3, β is passed as an input variable. Nevertheless, in our experiment, we specifically assigned the value of β to 0.1). This updating process is reflected in lines 27–30 of Algorithm 3. In fact, clusters with greater weights are more informative and will have greater chances of selection by RWS. The initial population in MDES has q solutions. MDES guarantees these solutions have distinct lengths ensuring that the ith solution has the length of i with the help of RWS. Algorithm 4 shows RWS without replacement where the cluster and its corresponding weight are deleted in lines 8 and 9 upon

Algorithm 3 Multi-objective Discrete Evolution Strategy

Input: X', q, β
Output: BestParetoFront
1: C^q = Kmeans(X'.T, q)
2: W = q × [1/q]
3: **for** i=1 to q **do**
4: mask=RWS(C^q, i, W)
5: subset = Generate corresponding dataset from mask using X'
6: pop[i].Data=subset
7: pop[i].Fit = Fitness(subset)
8: **end for**
9: [pop, Fronts] = NonDominatedSorting(pop)
10: pop = CrowdingDistance(pop, Fronts)
11: [pop,ParetoFront,maxAcc,selectedClusters] = SortPopulation(pop)
12: bestAcc = maxAcc
13: **for** co=1 to MaxIt **do**
14: **for** i=1 to q **do**
15: mask=RWS(C^q, i, W)
16: subset = Generate corresponding dataset from mask using X'
17: pop2[i].Data=subset
18: pop2[i].Fit = Fitness(subset)
19: **end for**
20: pop = pop+pop2
21: [pop, Fronts] = NonDominatedSorting(pop)
22: pop = CrowdingDistance(pop, Fronts)
23: [pop,ParetoFront,maxAcc,selectedClusters]= SortPopulation(pop)
24: pop = pop[1:nPop]
25: **if** maxAcc > bestAcc **then**
26: bestAcc = maxAcc; BestParetoFront=ParetoFront;
27: **for** i in selectedClusters **do**
28: $W[i] = W[i] + \beta \times (1 - W[i])$
29: **end for**
30: $W = W/\text{sum}(W)$
31: **end if**
32: **end for**
33: Return BestParetoFront

selection. Lines 10-12 update the weights of the remaining clusters in Algorithm 4.

$$W_{c_i} = \frac{1}{q} \tag{11}$$

$$W_{c_i} = W_{c_i} + \beta \times (1 - W_{c_i}) \tag{12}$$

Algorithm 4 Roulette Wheel Selection without replacement

Input: cln, popSize, p
Output: list
1: list = []
2: **for** s=1 to popSize **do**
3: i = len(p)
4: r = U(0,1)
5: $cs_i = \sum_{j=1}^{i} p_j$
6: i = min{$j | r \le cs_j$}
7: list.append(cln[i])
8: del cln[i]
9: del p[i]
10: **if** p ≠ ∅ **then**
11: p = p/sum(p)
12: **end if**
13: **end for**
14: Return list

Lines 10–12 update the weights of the remaining clusters. Algorithm 3 records the selected list of clusters by RWS in *mask* and accordingly *subset* is a dataset created via random sample (feature) selection within each cluster of *mask*. The fitness of the *subset* is calculated immediately after each solution is created and finally recorded to the corresponding solution in *pop*. The fitness function is a vector constructed by the length of the subset (h) and accuracy as stated in Eq. (13).

$$Fitness(subset) = [\frac{q - h + 1}{q}, accuracy(subset)] \tag{13}$$

Algorithm 3 continues by calculating the ranks of each solution by *NonDominatedSorting* function in line 9 (which uses Eq. (14) for dominance calculation in Algorithm 5), ordering the members of each front based on *CrowdingDistance* in line 10 (Algorithm 6), and sorting the population members based on their corresponding ranks and crowding distance via *SortPopulation* in Line 11 (Algorithm 7). In multi-objective algorithms, the quality of solutions investigated by dominant solutions has greater priority than the diversity of solutions investigated by crowding distance calculations.

$$x \succeq y \quad \Leftrightarrow \quad \forall_i : x_i \ge y_i, \quad \exists_{i_0} : x_{i_0} > y_{i_0} \tag{14}$$

Algorithm 5 initializes the s_p, and n_p of the solutions in the initial population, *pop*, in line 1. Generally, Algorithm 5 specifies the Pareto front, F_1, in lines 3–11 so that the members in the Pareto front have a rank of 1. The remaining non-Pareto frontiers and their corresponding ranks (if any) are specified in lines 12–29. Algorithm 6 shows *CrowdingDistance* based on Eqs. (15)–(16) to ensure that the greater crowding distances are better. Generally, crowding distance for a solution i is calculated relative to the first and last solutions within a front and the previous and next neighbors of the solution i as stated in Eq. (15) in which d_i^j is the crowding distance of solution i relative to objective j. Eq. (15) is generalized in Eq. (16) with k objective functions (in this paper k=2 based on Eq. (13)). The parameters h, CD_i, and $nObj$ in Algorithm 6 are the size of the front in which the crowding distances should be calculated, crowding distance of solution i, and the number of objective functions respectively. Lines 7–13 in Algorithm 6 calculate the crowding distances for the elements in a certain front, and the crowding distance of corresponding populations are updated in line 14 accordingly. Finally, Algorithm 7 sorts the population based on the crowding distances in line 3 and then the population is reordered based on the ranks in line 5.

$$d_i^j = \frac{|f_j^{next} - f_j^{previous}|}{f_j^{max} - f_j^{min}} \tag{15}$$

$$d_i = d_i^1 + d_i^2 + \cdots + d_i^k = \sum_{j=1}^{k} d_i^j \tag{16}$$

Algorithm 7 also records the list of solutions within the Pareto front ($ParetoFront$), the maximum accuracy in the Pareto front ($maxAcc$), and the clusters from which the best solution in Pareto front is constructed ($selectedClusters$). Subsequently, Algorithm 3 utilizes $maxAcc$ and $selectedClusters$ to update the cluster weights. It repeats lines 13–32, which involve generating new populations in lines 14–19, merging populations in line 20, truncating the sorted population in line 24, and checking the possibility of updating cluster weights in lines 27–30, until reaching the maximum iteration limit (MaxIt=200 in this paper).

5. Experimental results

This section introduces the datasets under study, experimental setups for implementation to address reusability, and the measurement criteria.

Algorithm 5 NonDominatedSorting

Input: pop
Output: pop, Fronts
initialize s_p = [] and n_p = 0 for p in pop[p]
Fronts{1}=[]
for p=1 to nPop **do**
  for q=p+1 to nPop **do**
    Update s_p, n_q Or s_q, n_p based on the dominance of [pop[p].Fit, pop[q].Fit] calculated by Eq. (14)
  end for
  if pop[p].n_p == 0 **then**
    Fronts{1}.append(p)
    pop[p].rank = 1
  end if
end for
k=1
while True **do**
  draft = []
  for i in F{k} **do**
    for j in pop[i].s_i **do**
      pop[j].n_j = pop[j].n_j - 1
      if pop[j].n_j==0 **then**
        draft.append(j)
        pop[j].rank = k + 1
      end if
    end for
  end for
  if draft == ∅ **then**
    break
  end if
  Fronts{k + 1} = draft
  k = k + 1
end while
Return pop, Fronts

Algorithm 6 CrowdingDistance

Input: pop, Fronts
Output: pop
for s=1 to (numel(Fronts)) **do**
  h=|Fronts{s}|
  for i=1 to h **do**
    CD_i=0
  end for
  nObj=len(Fronts{s}.Fit)
  for j=1 to nObj **do**
    sort (Fronts{s},f_j)
    CD_1 = CD_h = inf
    for i=2 to h-1 **do**
      CD_i= CD_i + d_i^j based on Eqs. (15) and (16)
    end for
  end for
  Update the CD value of corresponding solution in pop with related CDs
end for
Return pop

5.1. Datasets

The datasets under study are nine benchmark high dimensional microarray datasets listed in Table 3. The datasets are diverse in number of features from 2000 in Colon to 22283 in GLI while containing few samples. The data distribution is almost balanced in SMK and MLL, but the rest of the datasets have different degrees of imbalance. Numerous papers extensively use these datasets in feature selection (Abasabadi et al., 2021, 2022; Agarwalla & Mukhopadhyay, 2022; Nematzadeh et al., 2022; Yan, Ma, Luo, & Patel, 2019). Moreover, they were selected due to different traits that render them suitable for testing the performances of MPR-MDES.

Algorithm 7 SortPopulation

Input: pop
Output: pop, ParetoFront, maxAcc, selectedClusters
ParetoFront = []
CDSO = Sort order the population based on the crowding distance descendingly
pop = pop[CDSO]
RSO = Sort order the population based on the ranks
pop = pop[RSO]
ParetoFront = append all solutions with rank of 1
maxAcc = Save the maximum classification accuracy in Pareto front
selectedClusters = Save the clusters of the solution in ParetoFront with maxAcc
Return pop, ParetoFront, maxAcc, selectedClusters

5.2. Measurement criteria

The overall accuracy is a well-known metric for the evaluation of classification models. It shows the fraction of predictions the model got right. A good feature selection method is expected to increase the overall accuracy by selecting the most informative features. The overall accuracy is defined mathematically in Eq. (17), where TS stands for the test set with a size of $|TS|$. Additionally, $h(x_i)$ and Y_i are the classifier's prediction and the real label for ith element of the TS, respectively. The numerator of Eq. (17) is a dummy variable and equals 1 if ($Y_i = h(x_i)$) and otherwise 0. Precision in Eq. (18), Recall in Eq. (19), and Fscore in Eq. (20) are also important metrics besides the overall accuracy, particularly when the datasets are imbalanced as in medical datasets. Recall is the ability of a model to find all the relevant cases within a dataset. At the same time, precision quantifies the accuracy of positive predictions. Additionally, Fscore is the harmonic mean of precision and recall. Eqs. (18)–(20) denote the precision, recall, and Fscore for binary classification so that False Negatives(FN) are positive samples falsely classified as negative, False Positives(FP) are negative samples falsely classified as positive, and True Positives(TP) are the correct predictions of positive samples. For multi-label classification, the performance metrics (precision, recall, and Fscore) for each label (class) are computed independently using one-vs-rest strategy, and then the averages are taken.

$$\text{Overall accuracy} = \frac{\sum x_i \in TS \;\; \mathbf{1}(Y_i = h(x_i))}{|TS|} \tag{17}$$

$$\text{Precision} = \frac{TP}{TP + FP} \tag{18}$$

$$\text{Recall} = \frac{TP}{TP + FN} \tag{19}$$

$$\text{Fscore} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{20}$$

The balanced accuracy is a metric used for evaluating the performance of a classifier in binary or multi-label classification problems. It takes into account the imbalance in the class distribution and provides a more reliable measure of classification performance than traditional accuracy. For binary classification, the formula is in Eq. (21).

$$\text{Balanced accuracy-2L} = \frac{\text{Recall} + \text{Specificity}}{2} \tag{21}$$

Table 3
Descriptions of nine benchmark datasets.

Dataset	Sample size	Feature size	Number of classes	Sample distribution	Year of publication
Colon	62	2000	2	22–40	1999
CNS	60	7129	2	21–39	2002
GLI	85	22283	2	26–59	2004
SMK	187	19993	2	90–97	2007
Leukemia - Two-label	72	7129	2	47–25	1999
Leukemia - Multi-label	72	7129	3	25-38-9	1999
Covid-19	234	15979	3	100-41-93	2020
MLL	72	12582	3	24-20-28	2002
SRBCT	83	2308	4	29-11-18-25	2001

Table 4
Average accuracy (overall and balanced) before and after applying MPR-MDES, along with the average subset length of MPR-MDES, on benchmark datasets.

Dataset	Overll accuracy (before MPR-MDES)	Balanced accuracy (before MPR-MDES)	Overall accuracy (after MPR-MDES)	Balanced accuracy (after MPR-MDES)	Subset length (after MPR-MDES)
Colon	0.74	0.74	0.94	0.94	6
CNS	0.58	0.52	0.88	0.87	6
GLI	0.79	0.76	0.93	0.93	5
SMK	0.59	0.59	0.75	0.74	6
Leukemia - Two-label	0.84	0.84	0.99	0.99	3
Leukemia - Multi-label	0.84	0.79	0.95	0.94	4
Covid-19	0.63	0.59	0.72	0.71	7
MLL	0.85	0.84	0.95	0.95	5
SRBCT	0.79	0.77	0.94	0.94	7

where specificity is calculated using True Negatives(TN) and False Positives(FP) in Eq. (22).

$$\text{Specificity} = \frac{TN}{TN + FP} \tag{22}$$

However, balanced accuracy is defined as the average of recalls obtained on each class for multi-label datasets where C equals number of classes in Eq. (23).

$$\text{Balanced accuracy-ML} = \frac{1}{C}\sum_{i=1}^{C} \text{Recall}_i \tag{23}$$

The last criterion is the subset length, which is the length of the automatically selected features by MPR-MDES.

5.3. Experimental setup

This paper uses Decision Tree (DT) as a classifier to calculate the measurement criteria discussed in Section 5.2. 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 two samples. The parameters *random_state* and *class_weight* are None and *ccp_alpha* is the default 0. For calculating the precision, recall, and Fscore of multi-label datasets, the *average* parameter is set to 'micro'. The stratified train-test split is used with test size=20% and the accuracy of a selected subset is the average of 10 times stratified train-test split. This research is implemented using Python 3.9.13 platform on a computer with a Core i5 processor (1.60 GHz–2.30 GHz), 12 GB RAM, 720 GB HDD, and 64-bit Windows 10 Pro operating system.

5.4. Performance analysis

Table 4 shows the average subset length and average accuracy (overall and balanced) after applying MPR-MDES on datasets of Table 3 compared with average accuracy (overall and balanced) before applying MPR-MDES. The average accuracy after applying MPR-MDES refers to the average greatest accuracy achieved in the Pareto front for 10 runs of the proposed method. Meta-heuristic algorithms are stochastic, and thus MPR-MDES is stochastic accordingly. Therefore, the selected features (subset) may differ for each proposed method run. Hence, the results in Table 4 are rounded based on averaging 10 runs of MPR-MDES to make the results reproducible.

Moreover, it is evident in Table 4 that MPR-MDES increases the accuracy of the decision tree classifier considerably in comparison with the average accuracy before applying feature selection. The difference between overall accuracy and balanced accuracy is generally negligible. Thus, we will refer to overall accuracy throughout the rest of the paper when discussing accuracy. The highest increase in overall accuracy is for CNS, from 0.58 to 0.88, and the lowest increase is recorded for Covid-19 from 0.63 to 0.72. This considerable accuracy increase becomes more important because the lengths of the selected subset of features are always less than 10 in all datasets. The lowest average subset length is recorded for Leukemia - Two-label with 3 features. Likewise, the highest subset lengths are recorded for both Covid-19 and SRBCT with 7 features in Table 4. The average accuracies and average subset lengths of datasets, as reported in Table 4 after applying MPR-MDES, show a significant improvement in classification performance. However, this improvement alone is not sufficient for a complete evaluation.

Accuracy alone can be misleading, particularly in imbalanced datasets. Therefore, the overwhelming number of samples from the majority class (or classes) will overwhelm the number of samples in the minority class. In such cases, the calculation of precision, recall, and Fscore are important besides accuracy. Fig. 3 shows precision, recall, and Fscore of MPR-MDES (red bars) for the datasets in Table 3 and its comparison with before applying MPR-MDES (gray bars). Fig. 3 clearly illustrates the increase of all measurement criteria for all datasets after applying MPR-MDES. This increase is considerable and evident in Colon, CNS, GLI, and SMK, and moderate in the rest of the datasets. It can be concluded that MPR-MDES suggests that the selected features are particularly relevant for identifying the minority class, which led to an improvement in the model's performance.

Fig. 4 shows the Pareto front when MPR-MDES stops optimization. The red circles are the best solutions found in the Pareto front, and the blue dashed line is the overll accuracy of the decision tree before applying MPR-MDES using all features of the corresponding dataset. It should be noted that the Pareto front may contain some red circles overlapping, which makes sense when using meta-heuristic algorithms. Additionally, the Pareto fronts can contain at most q solutions based on Algorithm 3 (q=10 in this paper).

Accordingly, the Pareto fronts in Fig. 4 contain a sufficient number of solutions (relative to the maximum possible solutions), which

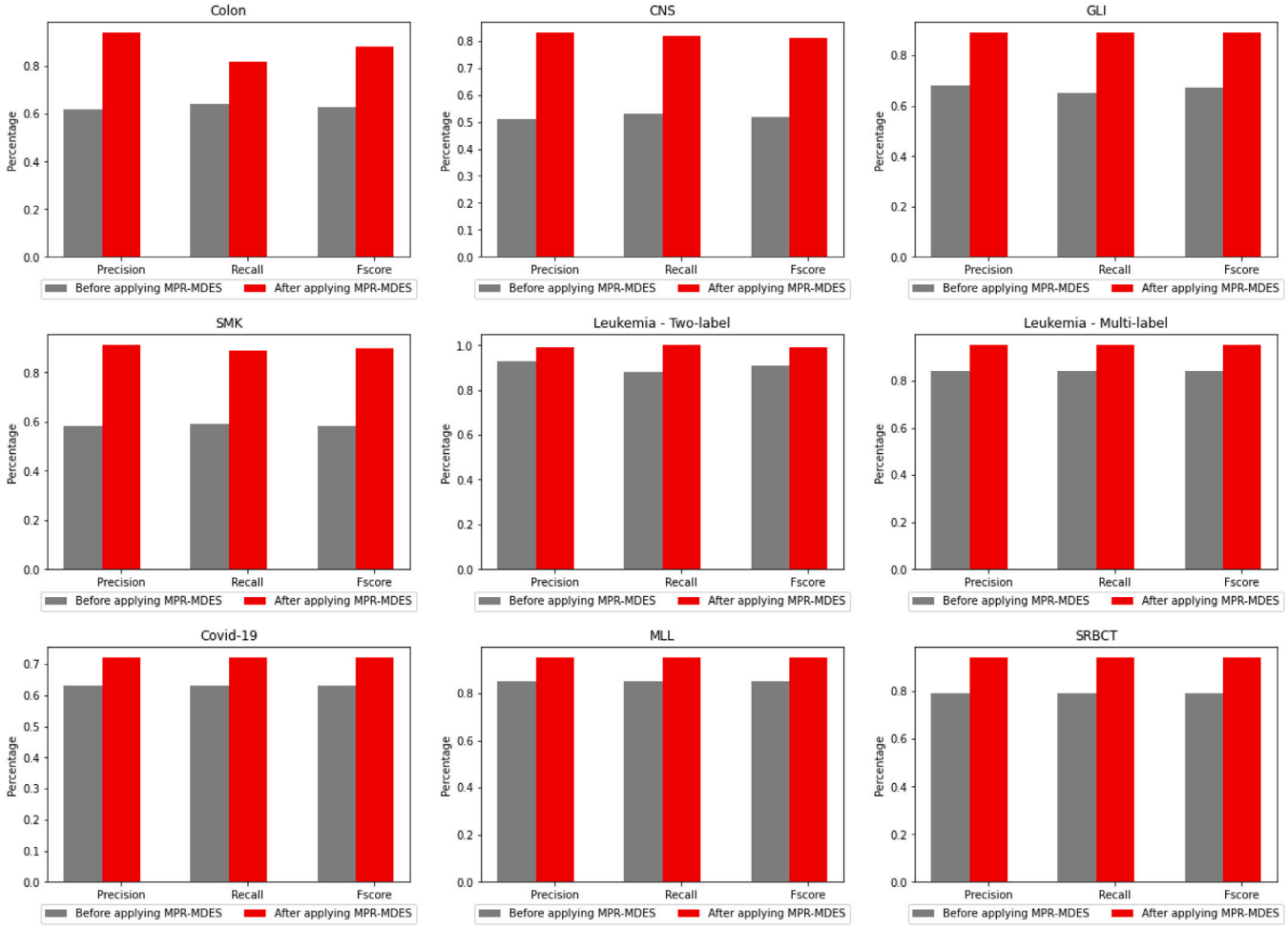


Fig. 3. Precision, recall, and Fscore achieved by MPR-MDES.

implicitly confirms using multi-objective optimization. Single-feature solutions could not significantly increase accuracy compared to the baseline (blue dashed line) in Covid-19, MLL, and SRBCT. In addition, the accuracy of the solutions on the Pareto front increases as the number of features increases, which is reasonable because the solutions on the Pareto front are non-dominated solutions. In contrast to single-objective optimization, which always ends up with one solution, there are multiple non-dominated solutions in multi-objective optimization. The idea is to let the decision-maker (domain expert) decide and choose between one of these solutions on the Pareto front. For example, the Pareto front of GLI has three solutions with sizes one, two, and three and overall accuracy of 0.91, 0.93, and 0.95 respectively. Thus, the domain expert has three alternatives (subsets of microarray genes) to select from. However, if the maximum accuracy is intended, the subset with a size of three is the best. All in all, the information in Fig. 4 shows that the combination of Maximum Pattern Recognition (MPR) and Multi-objective Discrete Evolution Strategy (MDES) considerably increases the accuracy of the classifier. It also provides multiple solutions to the decision maker to select and examine different sets.

MDES is fast because it only uses intelligent mutation by assigning more weights to clusters containing better features for purposeful selection of subsets by Roulette Wheel Selection (RWS) without replacement in Algorithm 4.

5.5. Comparison

To begin with, this section will compare the accuracy of frequency-based rankers for the top 10 features ranked by each ranker in Section 5.5.1. Then, we will compare the time complexity of these rankers in Section 5.5.2. Finally, the proposal is compared with related state-of-the-art works in Section 5.5.3.

5.5.1. Accuracy of rankers

Fig. 5 compares the existing frequency-based rankers namely, Mutual Congestion (MC) (Nematzadeh et al., 2019), Sorted Label Interference (SLI) (Abasabadi et al., 2021), Sorted Label Interference-γ (SLI-γ) (Abasabadi et al., 2022), Extended Mutual Congestion (EMC) (Nematzadeh et al., 2022), and Maximum Pattern Recognition (MPR). MC, SLI, and SLI-γ (which is shown by SLI_g in Fig. 5) only apply for two-label datasets; however, EMC and MPR are applicable for both two-label and multi-label datasets. The results in Fig. 5 show that MPR performs outstanding behavior in two-label datasets.

This superiority of MPR is more evident in Colon and competitive in others. However, there is no considerable difference between EMC and MPR in general in multi-label datasets. For example, MPR outperforms EMC with some feature sizes, and EMC outperforms MPR in some others in Leukemia - Two-label, Covid-19, and MLL. All in all, the information in Fig. 5 shows that MPR selects diverse and different features compared to other frequency-based rankers. This happens because MPR has an entirely different formulation to rank the features, as stated in Section 4.1. The selected features by MPR sometimes lead to better accuracy, such as Colon, and sometimes are very competitive.

5.5.2. Time complexity of rankers

Additionally, the comparison between time complexities of frequency-based methods besides the accuracy can better clarify the superiority of MPR. According to Algorithm 2 each feature should be

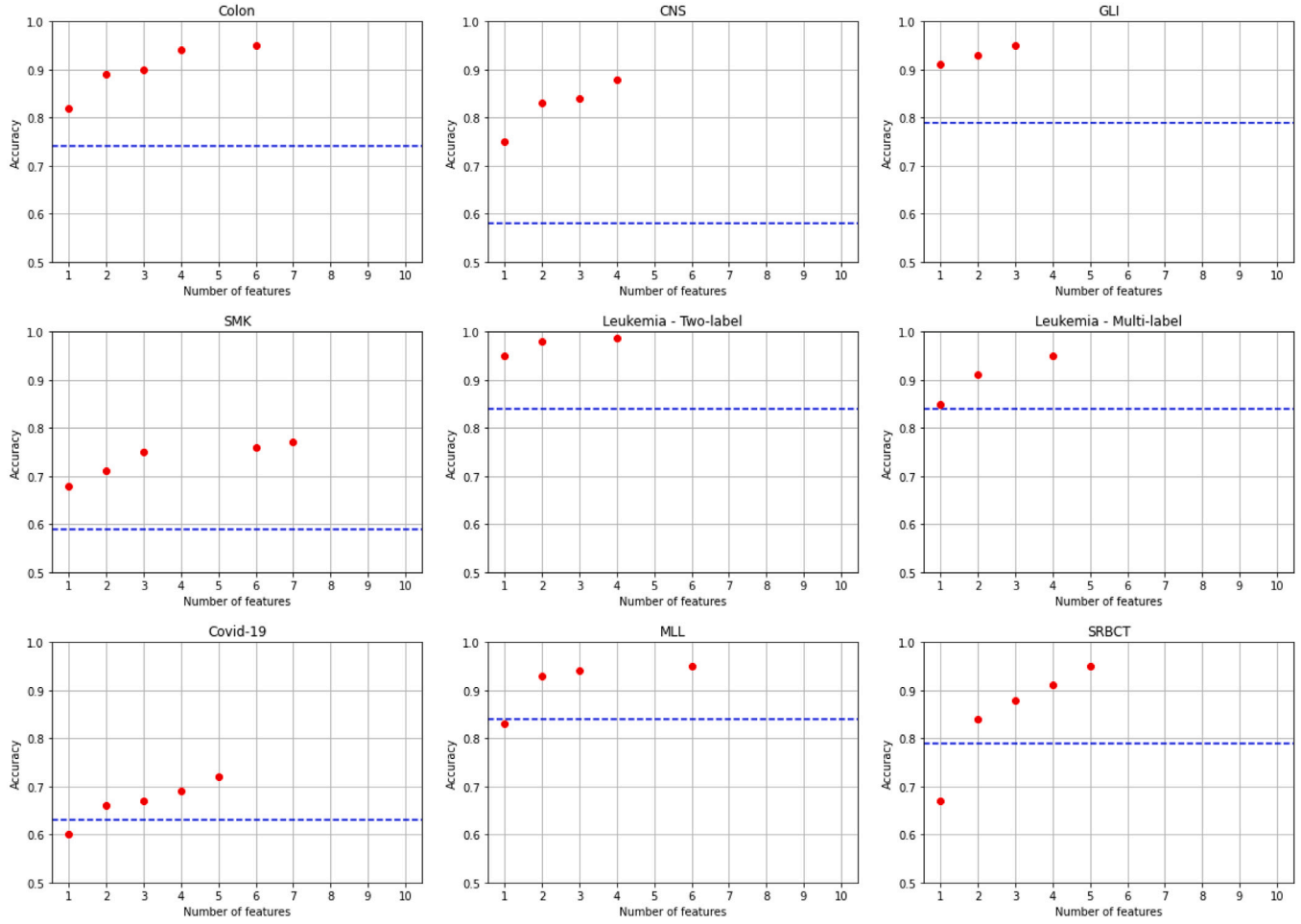


Fig. 4. Pareto front of MPR-MDES (red circles) compared with overall accuracy before applying MPR-MDES (blue dashed line).

sorted in line 13 and MPR is calculated based on the corresponding reordered response variable. As such, sorting has the time complexity of $O(n \log n)$ with a merge sort algorithm for each feature. The calculation of z and max_pattern in lines 14 and 16 takes $O(n)$ time each, and lines 18–20 spend $O(l)$ time as well for the MPR calculation of individual features. Therefore, the overall time complexity of MPR for all features of the dataset is $O(m(n \log n + n + n + l))$. We know that $l \leq n$ in the multiclass classification problem. Hence, even by changing l to n (the worst case) the overall complexity will be $O(m(n \log n + n + n + n)) = O(mn \log n)$. This outstanding achievement confirms the efficiency of MPR compared to time complexities in Table 1. Therefore, the superiority of MPR on existing frequency-based methods can be itemized as follows, concentrating on the time and accuracy achieved:

1. MPR generally achieves better or competitive accuracy (depending on the dataset).
2. MPR and EMC are not clearly distinct by accuracy, and both deal with multi-label datasets. However, the efficiency of MPR is considerably better than EMC.
3. MPR is applicable for both two-label and multi-label datasets, unlike MC, SLI, SLI-γ, which are only applicable for two-label datasets.

5.5.3. Comparison of MPR-MDES with state-of-the-art methods

This section compares MPR-MDES with some existing feature selection methods intended for microarray medical datasets namely, WOA-MC (Nematzadeh et al., 2019), ATFS (Abasabadi et al., 2021), PEFS (Hashemi et al., 2021), EIT-bBOA (Sadeghian et al., 2021), BC-NMIQ-IAMB (Asghari et al., 2023), and EMC-DWES (Nematzadeh et al., 2022) (Information about these methods are available in Table 2). These methods were selected for comparison as they share many similarities, which is necessary for a fair comparison. First, they avoid blindly selecting numerous features and instead aim to identify optimal or near-optimal ones. However, some methods, such as BPSO-TEPD (Thaher et al., 2022) in Table 2, tend to select excessive features, which may lead to overfitting and reduced performance on unseen data. For example, on the Colon dataset using a decision tree classifier, BPSO-TEPD selects an average of 766 features with an average overall accuracy of 0.94. Likewise, the GLI dataset achieves an average overall accuracy of 0.96 with an average of 10304 features using a decision tree classifier as well. Second, other methods such as GAwC (Gutowski et al., 2022) need specific parameter settings requiring high computation resources such as High-Performance Computing (HPC) cluster. Therefore, to ensure a fair comparison, we have excluded the methods that could not be good representatives for comparison and only selected those from Table 2 that tend to achieve an optimal or near-optimal number of features while maintaining a suitable level of accuracy for the decision tree classifier as done in the proposal. Additionally, all the methods in Table 5 are hybrid except ATFS and PEFS, recalling that ATFS is an ensemble of three rankers, so one of the rankers (SLI) is frequency-based oriented. The performance of MPR-MDES is compared with related works in Table 5 (based on average subset length (SL) and average overall accuracy achieved (Acc)) as follows.

WOA-MC Vs. MPR-MDES: According to Table 5, it is clear that the results of MPR-MDES considerably outperform WOA-MC on both criteria. MPR-MDES automatically detects the best features with a subset length smaller than WOA-MC. However, the subset length in WOA-MC was manually set using a predefined threshold of 10. The average

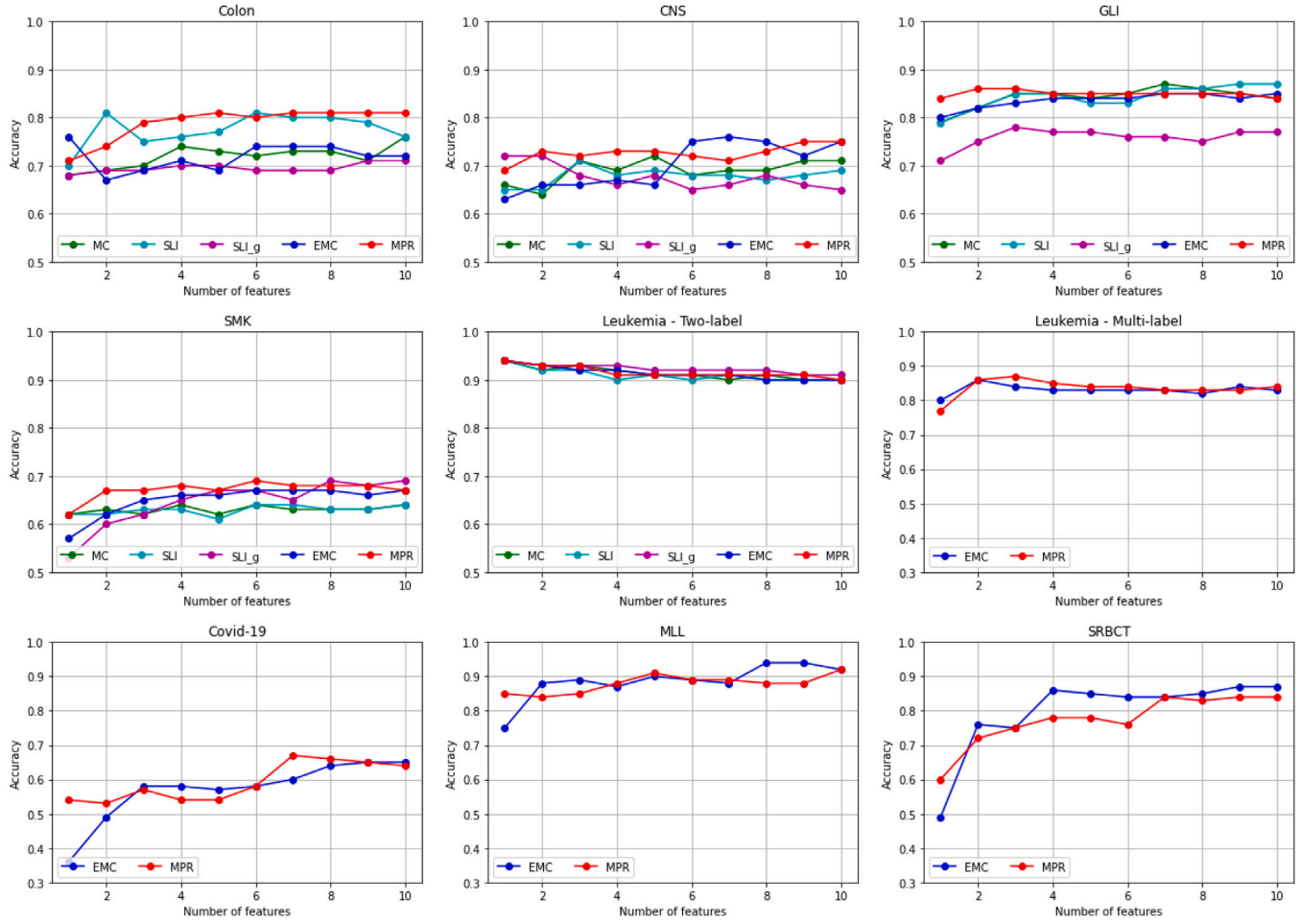


Fig. 5. Comparison of existing frequency-based rankers based on the overall accuracy achieved from the top 10 features.

accuracies reported in Table 5 for MPR-MDES are greater than WOA-MC even with fewer features. In fact, WOA-MC is a hybrid method of two filters, but MPR-MDES uses the wrapper MDES, which is one of the reasons for the outstanding performance of MPR-MDES over WOA-MC. Second, the MPR ranker addresses non-linearity more effectively compared to MC, which neglects non-linearity. Consequently, this could lead to a more accurate ranking of features.

ATFS Vs. MPR-MDES: ATFS is an ensemble feature selection method with automatic thresholding. It uses the concept of non-dominated sorting in multi-objective optimization for thresholding. MPR-MDES in Table 5 also achieves more accurate solutions with competitively less subset length. The wrapper MDES and the MPR ranker are the main reasons for this outstanding achievement, similar to what has been discussed for WOA-MC Vs. MPR-MDES.

PEFS Vs. MPR-MDES: PEFS formulates the feature selection problem as a Pareto-based optimization problem, similar to ATFS. It uses non-dominated sorting followed by crowding distance as a secondary measure. PEFS manually examines the best subset of features for the top features ranked from 10 to 100 with a step size of 10. The results for two-label datasets in Table 5 show that ATFS achieves better accuracies for Colon and CNS, while PEFS performs better for GLI, SMK, and Leukemia - Two-label. However, it is worth noting that ATFS consistently selects fewer features than PEFS in all cases, as shown in Table 5 for the two-label datasets. Nonetheless, the results of MPR-MDES outperform those of PEFS in Table 5. What is noteworthy in Table 5 is that the overall accuracy of PEFS for Covid-19 decreased after feature selection compared to the initial accuracy in Table 4.

EIT-bBOA Vs. MPR-MDES: Table 5 shows that EIT-bBOA achieves the highest overall accuracy for Covid-19 dataset. However, MPR-MDES generally achieves better overall accuracy. Furthermore, the selected feature subset lengths are always less than 10 using MPR-MDES, even though EIT-bBOA is fixed to select 30 features.

EMC-DWES Vs. MPR-MDES: EMC-DWES is the most analogous method to MPR-MDES in Table 5. This is because these two methods share many similarities; thus, comparing them provides readers with more insights. Both methods are a hybrid of a filter and a wrapper method. The filters belong to frequency-based rankers (EMC Vs. MPR) and the wrappers are created by the enhancement from the evolution strategy. DWES is a single-objective optimization which uses hierarchical clustering for discretization; however, MDES follows a multi-objective optimization approach. The reports in Table 5 show that MPR-MDES slightly outperforms EMC-DWES based on the average accuracy achieved. This is more evident in all two-label datasets. Moreover, the average length of selected subsets of features in MPR-MDES when achieving the average best overall accuracy in the Pareto front is always less than EMC-DWES. Additionally, the computational complexity of MPR ($O(mn \log n)$) also confirms that MPR-MDES is considerably faster than EMC-DWES (it should be taken into account that both DWES and MDES spend more or less the same execution time). Moreover, MDES in MPR-MDES does not need to tune any hyperparameter. This is despite the fact that the user should decide on determining two hyperparameters in DWES namely, the number of clusters in DWES and the type of linkage in hierarchical clustering. Hyperparameter tuning is truly time-consuming and adds time burden to DWES in comparison with MDES. Finally, MDES (owing to its multi-objective optimization) provides better perception by presenting many sets of selected features to decision-makers instead of just one set of features by single-objective DWES.

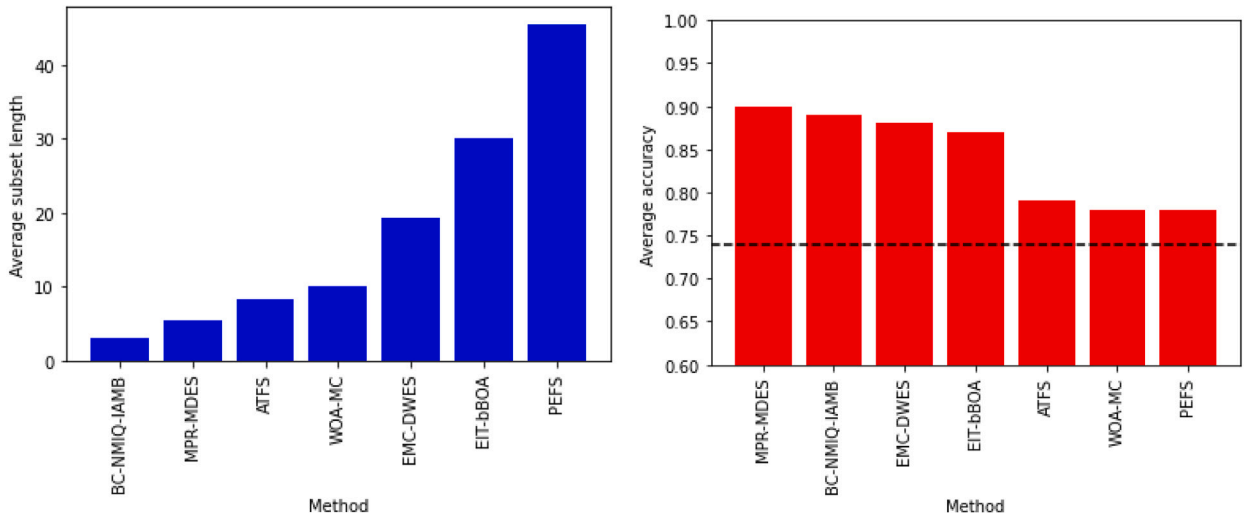


Fig. 6. Average overall accuracy and subset length across all datasets.

Table 5
Comparison of the proposed method MPR-MDES (in the last column at right) with state-of-the-art related methods. Dataset Leuk2L refers to Leukemia for two-label classification, while LeukML refers to multiclass classification. SL indicates the average subset length for each compared technique, and Acc is the mean accuracy.

Dataset	WOA-MC (Nematzadeh et al., 2019)		ATFS (Abasabadi et al., 2021)		PEFS (Hashemi et al., 2021)		EIT-bBOA(Sadeghian et al., 2021)		EMC-DWES (Nematzadeh et al., 2022)		BC-NMIQ-IAMB (Asghari et al., 2023)		MPR-MDES	
	SL	Acc	SL	Acc	SL	Acc	SL	Acc	SL	Acc	SL	Acc	SL	Acc
Colon	10	0.74	14	0.83	40	0.76	30	0.86	6	0.91	3	0.83	6	**0.94**
CNS	10	0.72	8	0.68	20	0.66	30	0.84	26	0.82	4	**0.93**	6	0.88
GLI	10	0.86	6	0.83	100	0.85	30	0.84	29	0.91	4	0.92	5	**0.93**
SMK	10	0.66	9	0.70	80	0.71	30	0.82	33	0.70	5	**0.83**	6	0.75
Leuk2L	10	0.92	4	0.92	10	0.94	30	0.89	17	0.97	3	0.98	3	**0.99**
LeukML	NA	NA	NA	NA	10	0.85	30	0.85	20	**0.97**	2	0.95	4	0.95
Covid-19	NA	NA	NA	NA	10	0.38	30	**0.94**	25	0.75	3	0.71	7	0.72
MLL	NA	NA	NA	NA	80	0.90	30	0.88	6	**0.96**	2	**0.96**	5	0.95
SRBCT	NA	NA	NA	NA	60	0.83	30	0.92	12	**0.94**	2	0.89	7	**0.94**

BC-NMIQ-IAMB Vs. MPR-MDES: The comparison of the results for MPR-MDES and BC-NMIQ-IAMB in Table 5 shows that these two methods achieve almost similar results in terms of subset length and accuracy. While BC-NMIQ-IAMB selects slightly fewer features, MPR-MDES achieves slightly better accuracy. In general, BC-NMIQ-IAMB is a good example to demonstrate that intelligently mixing filter feature selection methods may lead to good results.

Fig. 6 provides a more comprehensive comparison by presenting the average subset length (blue bars) and average overall accuracy (red bars) across all datasets using the decision tree classifier. The dashed line at the bottom indicates the average overall accuracy across all datasets using the decision tree classifier without any feature selection method. The results demonstrate that PEFS performs the worst in both criteria. On the other hand, MPR-MDES, which is a proposed improvement on EMC-DWES, outperforms EMC-DWES in both subset length and overall accuracy. While ATFS, WOA-MC, and PEFS show only slight improvements in average accuracy compared to the base dashed line, MPR-MDES, BC-NIMQ-IAMB, EMC-DWES, and EIT-bBOA have more significant improvements in average overall accuracy, with MPR-MDES showing the best results. Overall, Fig. 6 not only confirms the superiority of MPR-MDES over EMC-DWES, which is the main focus of the paper, but also shows that MPR-MDES outperforms many other existing feature selection methods, emerging as the top performer in terms of both subset length and accuracy.

Finally, these results are also reflected when checking the statistical confidence (in this study p-value = 0.05) regarding the average overall accuracy and the average number of features in the selected subsets. To this end, we have assessed the entire distribution of these two values with non-parametric statistical tests (Sheskin, 2007). In particular, Friedman's ranking and Holm's post-hoc tests have been applied to distinguish those algorithms statistically worse than the control one (the best-ranked according to Friedman). This way, as shown in Table 6 and focusing on overall accuracy (top), MPR-MDES is the best-ranked variant according to Friedman test and it is followed by BC-NMIQ-IAMB, EMC-DWES, and the remaining ones. Therefore, MPR-MDES is established as the control algorithm in the post-hoc Holm tests, which is compared with the rest of algorithms. The adjusted p-values (indicated as $Holm's_{Adj-p}$ in Table 6) resulting from these comparisons are, for algorithms BC-NMIQ-IAMB, EMC-DWES and EIT-bBOA, higher than the confidence level (0.05), so this means that no statistical difference can be observed with regards to MPR-MDES. Conversely, for the remaining variants PEFS, ATFS and WOA-MC, the adjusted p-values are lower than the confidence level, meaning that MPR-MDES performs statistically better than these algorithms in the context of average overall accuracy.

Regarding the average number of features, Table 6 (bottom) shows that BC-NMIQ-IAMB is established as the control algorithm according to Friedman's ranking, although with no statistical differences compared to MPR-MDES. On the contrary, the remaining techniques show statistically lower performance for this objective.

6. Conclusions

The paper proposes a hybrid (two-stage) feature selection method, MPR-MDES, explicitly designed for high-dimensional microarray and medical datasets. The method comprises two stages. Firstly, a new frequency-based ranker called Maximum Pattern Recognition (MPR) is introduced as a filter ranker, which enhances existing rankers by more effectively recognizing non-linear patterns, supporting multi-label

Table 6
Average Friedman's rankings with Holm's Adjusted p-values (0.05) of the compared algorithms. Symbol * indicates the control algorithm and the column at the right contains the overall ranking of positions with regard to average overall accuracy (top) and average number of selected features (bottom).

Average Overall Accuracy		
Algorithm	$Friedman's_{Rank}$	$Holm's_{Adj-p}$
MPR-MDES*	**2.22**	–
BC-NMIQ-IAMB	2.72	1.091E+0
EMC-DWES	2.83	1.091E+0
EIT-bBOA	3.83	3.408E−1
PEFS	5.16	1.534E−2
ATFS	5.61	5.251E−3
WOA-MC	5.61	5.251E−3
Average Number of Features		
Algorithm	$Friedman's_{Rank}$	$Holm's_{Adj-p}$
BC-NMIQ-IAMB*	**1.05**	–
MPR-MDES	2.00	3.537E−1
ATFS	3.66	2.060E−2
WOA-MC	4.04	9.659E−3
EMC-DWES	5.05	3.189E−4
EIT-bBOA	6.05	1.354E−5
PEFS	5.83	5.467E−6

datasets, and reducing time complexity. The filter MPR selects the top 20 features, which are then passed to the wrapper Multi-objective Discrete Evolution Strategy (MDES) to obtain the final feature subset. MDES provides several solutions in the Pareto front to choose from, allowing for the automatic selection of the optimal subset.

The experimental results show that MPR-MDES significantly outperforms state-of-the-art methods in terms of both accuracy and subset length. The proposed method exceeds the performance of existing works and achieves competitive results, making it a promising approach for feature selection in high-dimensional datasets.

As an avenue for future research, exploring the development of a frequency-based ranker that can effectively recognize non-linear patterns remains open, despite the superiority of the Maximum Pattern Recognition (MPR) method over linear frequency-based rankers in this aspect. Likewise, another direction of future work would be exploring the use of MPR-MDES in other domains, such as agriculture and smart cities.

CRediT authorship contribution statement

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

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

At the end of abstract, there is a link to the data and implementation.

Acknowledgments

This work has been partially funded by grants (funded by MCIN/AEI/10.13039/501100011033/) PID2020-112540RB-C41, AETHER-UMA (A smart data holistic approach for context-aware data analytics: semantics and context exploitation), and QUAL21 010UMA (Junta de Andalucía). It is also funded for open access charge: Universidad de Málaga/CBUA.

References

Abasabadi, S., Nematzadeh, H., Motameni, H., & Akbari, E. (2021). Automatic ensemble feature selection using fast non-dominated sorting. *Information Systems*, *100*.

Abasabadi, S., Nematzadeh, H., Motameni, H., & Akbari, E. (2022). Hybrid feature selection based on SLI and genetic algorithm for microarray datasets. *The Journal of Supercomputing*, *78*, 19725–19753.

Agarwalla, P., & Mukhopadhyay, S. (2022). GENEmops: Supervised feature selection from high dimensional biomedical dataset. *Applied Soft Computing*, *123*.

Amini, F., & Hu, G. (2021). A two-layer feature selection method using genetic algorithm and elastic net. *Expert Systems with Applications*, *166*.

Asghari, S., Nematzadeh, H., Akbari, E., & Motameni, H. (2023). Mutual information-based filter hybrid feature selection method for medical datasets using feature clustering. *Multimedia Tools and Applications*.

Chaudhuri, A., & Sahu, T. P. (2022). Multi-objective feature selection based on quasi-oppositional based Jaya algorithm for microarray data. *Knowledge-Based Systems*, *236*.

Forouzandeh, S., Aghdam, A. R., Forouzandeh, S., & Xu, S. (2020). Addressing the cold-start problem using data mining techniques and improving recommender systems by cuckoo algorithm: A case study of facebook. *Computing in Science & Engineering*, *22*(4), 62–73.

Forouzandeh, S., Berahmand, K., Sheikhpour, R., & Li, Y. (2023). A new method for recommendation based on embedding spectral clustering in heterogeneous networks (RESCHet). *Expert Systems with Applications*, *231*, 1–11.

Ganjei, M. A., & Boostani, R. (2022). A hybrid feature selection scheme for high-dimensional data. *Engineering Applications of Artificial Intelligence*, *113*.

Gutowski, N., Schang, D., Camp, O., & Abraham, P. (2022). A novel multi-objective medical feature selection compass method for binary classification. *Artificial Intelligence in Medicine*, *127*.

Hashemi, A., Mohammad, B. D., & Nezamabadi-pour, H. (2021). A pareto-based ensemble of feature selection algorithms. *Expert Systems with Applications*, *180*.

Jiménez-Cordero, A., Morales, J. M., & Pineda, S. (2021). A novel embedded min-max approach for feature selection in nonlinear Support Vector Machine classification. *European Journal of Operational Research*, *293*(1), 24–35.

Kushal, K. G., Begum, S., Sardar, A., Adhikary, S., Ghosh, M., Kumar, M., et al. (2021). Theoretical and empirical analysis of filter ranking methods: Experimental study on benchmark DNA microarray data. *Expert Systems with Applications*, *169*.

Li, X.-D., Wang, J.-S., Hao, W.-K., Wang, M., & Zhang, M. (2022). Multi-layer perceptron classification method of medical data based on biogeography-based optimization algorithm with probability distributions. *Applied Soft Computing*, *121*.

Mohapatra, P., Chakravarty, S., & Dash, P. (2016). Microarray medical data classification using kernel ridge regression and modified cat swarm optimization based gene selection system. *Swarm and Evolutionary Computation*, *28*, 144–160.

Negnevitsky, M. (2005). *Artificial intelligence : a guide to intelligent systems / michael negnevitsky* (2nd ed.). New York: Addison-Wesley, xiv, 415.

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

Nematzadeh, H., García-Nieto, J., Navas-Delgado, I., & Aldana-Montes, J. F. (2022). Automatic frequency-based feature selection using discrete weighted evolution strategy. *Applied Soft Computing*, *130*.

Niu, T., Wang, J., Lu, H., Yang, W., & Du, P. (2020). Developing a deep learning framework with two-stage feature selection for multivariate financial time series forecasting. *Expert Systems with Applications*, *148*.

Osama, S., Shaban, H., & Ali, A. A. (2023). Gene reduction and machine learning algorithms for cancer classification based on microarray gene expression data: A comprehensive review. *Expert Systems with Applications*, *213*.

Pan, H., Chen, S., & Xiong, H. (2023). A high-dimensional feature selection method based on modified Gray Wolf Optimization. *Applied Soft Computing*, *135*.

Saadatmand, H., & Akbarzadeh-T, M.-R. (2023). Set-based integer-coded fuzzy granular evolutionary algorithms for high-dimensional feature selection. *Applied Soft Computing*.

Sadeghian, Z., Akbari, E., & Nematzadeh, H. (2021). A hybrid feature selection method based on information theory and binary butterfly optimization algorithm. *Engineering Applications of Artificial Intelligence*, *97*.

Sahebi, G., Movahedi, P., Ebrahimi, M., Pahikkala, T., Plosila, J., & Tenhunen, H. (2020). GeFeS: A generalized wrapper feature selection approach for optimizing classification performance. *Computers in Biology and Medicine*, *125*.

Sánchez-Maroño, N., Fontenla-Romero, O., & Pérez-Sánchez, B. (2019). *Classification of microarray data* (pp. 185–205).

Sheikhpour, R., Berahmand, K., & Forouzandeh, S. (2023). Hessian-based semi-supervised feature selection using generalized uncorrelated constraint. *Knowledge-Based Systems*, *269*, 1–14.

Sheskin, D. J. (2007). *Handbook of parametric and nonparametric statistical procedures*. Chapman & Hall/CRC.

Thaher, T., Chantar, H., Too, J., Mafarja, M., Turabieh, H., & Houssein, E. H. (2022). Boolean particle swarm optimization with various evolutionary population dynamics approaches for feature selection problems. *Expert Systems with Applications*, *195*, 1–30.

Wei, G., Zhao, J., Feng, Y., He, A., & Yu, J. (2020). A novel hybrid feature selection method based on dynamic feature importance. *Applied Soft Computing*, *93*, 1–13.

Yan, C., Ma, J., Luo, H., & Patel, A. (2019). Hybrid binary coral reefs optimization algorithm with simulated annealing for feature selection in high-dimensional biomedical datasets. *Chemometrics and Intelligent Laboratory Systems*, *184*, 102–111.