Experimental Comparison of Metaheuristics for Feature Selection in Machine Learning in the Medical Context

Thibault Anani, Francois Delbot, Jean-François Pradat-Peyre

▶ **To cite this version:**

Thibault Anani, Francois Delbot, Jean-François Pradat-Peyre. Experimental Comparison of Metaheuristics for Feature Selection in Machine Learning in the Medical Context. 18th IFIP International Conference on Artificial Intelligence Applications and Innovations (AIAI), Jun 2022, Hersonissos, Greece. pp.194-205, 10.1007/978-3-031-08337-2_17 . hal-04668638

HAL Id: hal-04668638

https://inria.hal.science/hal-04668638v1

Submitted on 7 Aug 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Distributed under a Creative Commons Attribution 4.0 International License

This document is the original author manuscript of a paper submitted to an IFIP conference proceedings or other IFIP publication by Springer Nature. As such, there may be some differences in the official published version of the paper. Such differences, if any, are usually due to reformatting during preparation for publication or minor corrections made by the author(s) during final proofreading of the publication manuscript.

Experimental comparison of metaheuristics for feature selection in machine learning in the medical context

Thibault Anani[2], François Delbot[1.2], Jean-François Pradat-Peyre[1.2]

[1] Université Paris Nanterre, Nanterre, France
[2] LIP6, Sorbonne Université, Paris, France
{thibault.anani-agondja,francois.delbot,jean-francois.pradat-peyre}@lip6.fr

Abstract. We explore in this paper the use of metaheuristics to select features from a dataset in order to improve the prediction performance of models build with different machine learning methods. To this end, we compare the performances of 5 learning methods: Logistic Regression (LR), K-Nearest Neighbors (KNN), Gaussian Naive Bayes (GNB), Support Vector Machine (SVM) and Random Forest (RF) on 4 heterogeneous datasets in the number of data and features, for different feature selection methods (metaheuristics or statistical filters).
The results obtained show that feature selection by improving a metaheuristic derived from the genetic algorithm leads to much better performances no matter the learning method used compared to without feature selection on the same dataset.

Keywords: machine learning. features selection. optimization

1 Introduction

The implementation of a recommendation algorithm based on a learning method is confronted with various concerns, including the dimension of the data (the number of features) versus the number of available and usable data. It is frequent in real contexts that the data set is relatively small in size but faces a large dimension. This is mainly the case in many problems coming from the medical world. In this case, the risks of overfitting are frequent and the solutions of recommendations found generalize poorly to a real population. One of the solutions adopted consists in reducing the dimension of the data.

Dimension reduction is characterized by the projection of data described in N dimensions to a reduced space of dimension $K < N$. The main objective is to preserve the initial profile of the data by proposing a more relevant and compact representation. Moreover, reducing the dimensionality makes visible the underlying structure generally not very readable in high dimension; one thus avoids the problems related to the concept of 'the curse of dimensionality' introduced by Bellman in 1961.

Several approaches exist [1] to reduce the dimension: feature selection which consists in keeping only a subset of the initial features or feature extraction which relies on a global transformation of the data thanks to an application that induces a change of coordinates [2], as in the case of the Fourier transform in signal processing.

Confronted with this high dimensionality problem in previous work on medical data analysis to improve the management of patients suffering from Amyotrophic Lateral Sclerosis (ALS) [3], we have developed a robust method based on 1) manual selection (with the help of ALS experts) of patient characteristics 2) followed by a dimension reduction phase using the Uniform Manifold Approximation and Projection (UMAP) method which is based on the assumption that the data belong to a Riemannian variety, a particular form of regular variety [4].

We are interested here in dimension reduction by feature selection and more precisely in dimension reduction with the envelope approach [5]. This approach is associated with learning and compares the different subsets of possible features with the performance of the learning model used.

More precisely, we show empirically that the use of metaheuristics and in particular a variant of the population-based metaheuristic called 'differential evolution' gives excellent results no matter the learning method used by selecting a relevant subset of features from the data of the problem studied.

These results are obtained by considering multiple datasets from the medical field one from Pro-Act on ALS and several benchmarks regularly used in comparisons of learning methods.

2 Methodology used

The data used in the medical context (for classification or prediction) present a particular profile: we frequently observe quite few complete data but the features associated with these data are often numerous (or even very numerous) due to the fear of underfitting by neglecting important parameters. This means that, without precaution, the models obtained by learning (supervised or not) generalize rather poorly. Reducing the number of features used during the learning phases has several benefits: on the one hand, it avoids overfitting and reduces the noise produced by the data, which improves the performance of the model and its ability to generalize. On the other hand, it induces a simplification of the hypotheses necessary for the use of the model, thus facilitating the treatments and improving the calculation time. Finally, it is easier to produce complete datasets because the amount of information to be collected is less.

In order to select the 'best' subset of features. we need to define how one subset is better than another and how to obtain this subset in an efficient and relevant way and how to validate this choice.

2.1 Evaluation criteria

The evaluation criteria we use are the most frequently used criteria in this context; let $D = \{x\}$ be a data set. $V(x)$ a classification function that defines

whether x is 1 or 0 and $f(x)$ a prediction function (that associates a boolean to x); We first define TP as the true positive (i.e. $\{x|f(x) = V(x) = 1\}$), FP as the false positive (i.e. $\{x|f(x) = 1 \wedge V(x) = 0\}$), TN as the true negative (i.e. $\{x|f(x) = V(x) = 0\}$), and FN as the false negative (i.e. $\{x|f(x) = 0 \wedge V(x) = 1\}$).

- The **sensitivity** or **recall** measures the true positive rate (i.e. $\frac{|TP|}{|TP|+|FN|}$). In medicine: the proportion of people correctly tested positive for a disease among those who have this disease.
- The **specificity** measures the true negative rate (i.e. $\frac{|TN|}{|TN|+|FP|}$). In medicine: the proportion of people tested negative for a disease among those who do not have that disease.
- The **accuracy** measures the proportion of correct predictions (i.e. $\frac{|TP|+[TN|}{|D|}$). In medicine: the proportion of people correctly diagnosed for a disease among the whole population.

We use either the accuracy (for datasets that have a balance between positive and negative cases) or the average of the sensitivity and the specificity (for unbalanced datasets) to measure performance.

2.2 Obtaining and validating an 'optimal' subset

The objective of the experiments is to find the optimal subset of features i.e. the subset that will allow us to obtain the most performing model (w.r.t. accuracy, recall or specificity) with the data we have. A potential subset called solution is represented as a vector of booleans of the size of the number of features we have (see Fig.1). Depending on the value of a boolean, a feature is taken into account or not to perform a learning: 0 the feature is not taken into account and 1 it is taken into account for learning. To carry out a learning, at least one explanatory feature is needed, therefore a vector cannot be composed only of 0.

Fig. 1. Example of 3 solutions represented as a vector of booleans with a number of Features equal to 6

	v_0	v_1	v_2	v_3	v_4	v_5
i_1	0	1	0	1	0	1
i_2	0	1	1	0	1	0
i_3	1	1	0	1	1	1

In order to validate the relevance of a solution for a learning method, we use cross-validation which consists in decomposing the dataset into k subsets and then using $k-1$ subsets for learning and the k^{th} subset for validation. More precisely, we do k experiments by choosing at each experiment a different subset for the validation. We use $k = 5$ in our experiments.

The machine learning methods we use are the Logistic Regression (LR), the Support Vector Machine (SVM), the K-Nearest Neighbors (KNN), the Random Forest (RF) and the Gaussian Naive Bayes (GNB). See [6] for a complete description of these methods.

3 Feature selection with the use of Metaheuristics

Feature selection consists of selecting a portion of our features that are most relevant to the construction of the model to improve its performance. Ideally, we should test and evaluate all combinations of features to find the most efficient one. However, when the number of features is high, it is simply impossible to test and evaluate all possible combinations. which leads to a combinatorial explosion e.g. with 180 features, we have $2^{180} - 1$ subsets to explore, which is impossible to achieve in a reasonable time. We will therefore use methods from operational research, called metaheuristics, to find the best possible subset of features.

Metaheuristics are computational methods for solving complex optimization problems and finding the optimal solution or at least an approximate solution. These methods are divided into 2 branches. Solution based metaheuristics try to improve one solution at a time by searching in its neighborhood whereas Population-based metaheuristics improve several solutions at the same time and merge them together to obtain better solutions. In a previous study we used 7 metaheuristics: 4 are solution-based (Tabu search, Simulated annealing, Random search and Hill climbing) and 3 are population-based (Genetic algorithm, Differential evolution and Particle swarm optimization). Here we focus on the ones that gave the best results: 3 population-based (Population-Based Incremental Learning, Differential evolution and Particle swarm optimization) and 2 solution-based (Tabu search and Simulated annealing). For all metaheuristiscs the number of individuals/neighbors at each generation is set to 50 and the number of generations is set to 500.

3.1 Solution-based metaheuristics

Tabu Search (TS) performs a local search to solve complex or large optimization problems [7]. The concept is to use a memory system to deny for a given period of time to revisit a previously visited solution and to allow moves that do not necessarily improve it, thus allowing the search to continue even when a local optimum is found. The size of the tabu list is set to 1000.

Simulated Annealing (SA) is an algorithm based on the annealing process used in metallurgy to achieve thermal equilibrium at each temperature. The initial solution is used as a candidate feature subset, and the feature subset is updated according to its neighborhood and according to the temperature-dependent probabilities of selecting better or worse solutions [8]. The temperature is set to 500 and decrease by 1 at each generation.

3.2 Population-based metaheuristics

Population-base incremental learning (PBIL) is an improvement of the well-known genetic algorithm. It converges toward the optimal solution of a problem using a probability vector as large as the number of features [9]. This vector is used to generate the individuals of a population and is updated at each generation. The learning rate is set to 0.1, the mutation probability to 0.2 and the mutation shift to 0.05.

Particle swarm optimization (PSO) relies on population collaboration. Individuals called particles move in the search space, each representing a features subset. The particles will evolve by following the influence of the best performing ones and their own previous movements in the search space [10]. The inertia weight coefficient and both the acceleration factors are set to 0.5.

Differential evolution (DE) uses the diversity present between individuals of a population to explore the different areas of the search space using mutation operations [11]. The population is composed of N Individuals denoted $P_G = \{X_1^G, X_2^G, ..., X_N^G\}$ at the generation G (where $G \in [1, G_{max}]$ and $X_{i,j}^G$ denoting the j-th, $j \in [1, D]$, component of the vector X_i^G). By performing mutation, crossover and selection operations the population improves over the generations until the stopping criterion is reached. For the initial generation, these individuals are generated randomly.

At each generation the algorithm performs first a mutation step using a mutation strategy which can be expressed as 'DE/x/y' where DE stands for differential evolution, x refers to how a vector (individual) in the mutation operation is chosen and $y \in \mathbb{N}$ specifies the number of differential vectors in the mutation strategy. Then the algorithm performs a crossover operation.

The most common and widely used strategy for the mutation is 'DE/rand/1' as depicted below (V_i^G is the mutant obtained by the mutation):

$$V_i^G = X_{r1}^G + F \times (X_{r2}^G - X_{r3}^G) \tag{1}$$

with $i = \{1, 2, \ldots, N\}$, $r1, r2, r3$ are random numbers belonging to $\{1, 2, \ldots, N\}$ s.t. $r1 \neq r2 \neq r3 \neq i$ and where $F \in [0, 2]$ is a constant probability factor that controls the amplification of the differential variation. There are other mutation strategies like 'DE/best/1' which uses the best performing vector of the generation instead of a randomly chosen one for X_{r1}^G.

In this paper the strategy 'DE/best/1' is the one used for better convergence in a limited number of generations.

The crossover is performed in order to generate a new vector U_i^G which is the cross between the original vector X_i^G and the mutant V_i^G. There are two types of crossover: binomial and exponential. In this paper the binomial crossover is used. The new vector U_i^G is generated as follows:

$$U_{i,j}^G = \begin{cases} V_{i,j}^G, & \text{if rand}(0,\ 1) \leq CR \text{ or } j = j_{rand} \\ X_{i,j}^G, & \text{otherwise} \end{cases} \tag{2}$$

where $j_{rand} \in [1, D]$ is a number chosen at random to reduce the chances that the vector U_i^G is composed only of the elements of X_i^G and $CR \in [0, 1]$ is the crossover probability. CR has a great influence on the diversity of the population build by the algorithm, since depending on its value, the number of elements that will change will be different: the higher the value, the greater the variation.

The last step is to select the best performing individuals. To know if the vectors generated by the crossover step will be kept, their score is compared to the score of the current vectors.

$$X_i^{G+1} = \begin{cases} U_i^G, & f(U_i^G) \geq f(X_i^G) \\ X_i^G, & \text{otherwise} \end{cases} \tag{3}$$

where $f()$ represents the fitness function of an individual. If a vector U_i^G has a better score than the vector X_i^G then we keep this vector for the next generation otherwise we reject it and we keep the previous one. The parameter F is set to 1 and CR is set to 0.5 in our experiments.

3.3 A new population-based metaheuristics

We propose in this paper a new enhanced binary differential evolution algorithm: the binary progressive learning differential evolution (BPLDE). This one is based on an improved mutation strategy: we propose to use directly the binary strings of the different individuals for the mutation strategy. The only possible values for a bit are 1 or 0 to indicate that a feature is selected for learning or not respectively. From this observation it is possible to calculate the result of all combinations of the initial mutation strategy 'DE/best/1' as follows:

$$V_{i,j}^G = \begin{cases} X_{best,j}^G, & \text{if } X_{r1,j}^G = X_{r2,j}^G \\ X_{r1,j}^G, & \text{otherwise} \end{cases} \tag{4}$$

Performing this transformation removes the F factor from the equation. Besides, this approach allows to use the bits present in a vector directly without having to perform a conversion operation which can take time when the size and number of the vectors are relatively important.

The choice of the mutation strategy is crucial to achieve good convergence. 'DE/rand/1' strategy takes a single random individual of the population as reference which allows to have a good exploration in the search space and to keep a good diversity in the population. However, performing a learning can be time consuming depending on the structure of the data and the learning algorithm used, thus the number of possible iterations for the algorithm is also limited. 'DE/best/1' strategy takes the best individual as a reference which allows to favour the exploitation and a faster convergence at the risk of reducing quickly the diversity between the individuals of the population and to remain blocked on a local optimum.

Therefore we can conclude that determining the reference individual has an important place in the proper running of the strategy. We propose a mutation strategy that offers a compromise between the two by emphasizing the exploration by using $\frac{N}{2}$ random individuals of the population ($P'_G \subset P_G$) with a wide range of possibilities to select the reference individual at early stage while towards the end we only use the best individual for the whole population to favor the exploitation. Based on these assumptions we propose the same equation as equation 4 except that instead of using $X^G_{best,j}$ we use $X^G_{pbest,j}$ which stands for one of the $p \in \mathbb{N}$ best performing individuals. If $X^G_i \notin P'_G$ then the best solution is chosen like 'DE/best/1' mutation strategy. Furthermore, $X^G_{r2',j}$ is now a solution randomly chosen from the union of the current population and the archive $P \cup A$. Indeed, at each generation the individuals that have been rejected are kept for a certain amount of time in a separate population A called archive. Having an archive provides information about the progress direction and is also capable of improving the diversity of the population [12]. If the size of A exceeds that of P then randomly selected solutions are removed from A to keep its size at most N. As the algorithm progresses the value of p is gradually reduced until it reaches 1 by using this method:

$$p = Max(1, N \times (1 - (\sqrt{\frac{G}{G_{max}} \times \alpha})))$$

where α is a parameter that determines the speed of reduction of p. The smaller α is, the slower p will decrease and more the exploration will be privileged over the exploitation.

Some values of CR generate individuals that are more likely to survive and these values should be kept for the following generations. This is the reason why having CR that can adapt itself according to population evolution at each generation is important. The operation is to record successful crossover probabilities (SCR) and use them to guide the new generation of new crossover rate for each individual X^G_i according to a normal distribution ($\mathcal{N}$) of mean μ_{CR} and standard deviation 0.1 which are described in [12].

$$CR^G_i = \mathcal{N}(\mu_{CR}, 0.1) \tag{5}$$

CR^G_i is the crossover probability of the individual X^G_i. The value of μ_{CR} is set to 0.5 at the beginning and is updated at each generation as follow:

$$\mu_{CR} = \mu_{CR} \times (1 - LR) + LR \times (\frac{1}{n} \sum_{i=1}^{n} SCR^G_i) \tag{6}$$

where $LR \in [0.1]$ is the learning rate which is a constant value that will impact the speed at which the value of μCR increase or decrease.

In our study, μ_{CR} is set to 0.05 and the α parameter to 1.5.

Algorithm 1: Binary progressive learning differential evolution

Set $\mu CR := 0.5$; $A := \emptyset$;
for $G := 1$ *to* G_{max} **do**
  $SCR^G := \emptyset$;
  $p := Max(1, N \times (1 - (\sqrt{\frac{G}{G_{max}} \times \alpha})))$;
  $P'_G :=$ Randomly choose $\frac{N}{2}$ individuals from P;
  for $i := 1$ *to* N **do**
    $CR_i := \mathcal{N}(\mu_{CR}, 0.1)$;
    if $i \in P'_G$ **then**
      $X^G_{pbest} :=$ Randomly choose one of the p best individuals from P;
    else
      $X^G_{pbest} :=$ Best individual from P;
    end
    do
      Randomly Choose X^G_{r1} from P and $X^G_{r2'}$ from $P \cup A$;
    while $r1 \neq r2 \neq i$;
    for $j := 1$ *to* D **do**
      if $X^G_{r1,j} = X^G_{r2',j}$ **then**
        $V^G_{i,j} := X^G_{pbest,j}$;
      else
        $V^G_{i,j} := X^G_{r1,j}$;
      end
    end
    $j_{rand} := randint(1, D)$;
    for $j := 1$ *to* D **do**
      if $rand(0, 1) \leq CR_i$ *or* $j = j_{rand}$ **then**
        $U^G_{i,j} := V^G_{i,j}$;
      else
        $U^G_{i,j} := X^G_{i,j}$;
      end
    end
    if $f(U^G_i) > f(X^G_i)$ **then**
      $A \leftarrow X^G_i$; $X^G_i := U^G_i$; $SCR^G \leftarrow CR_i$;
    end
  end
  $\mu_{CR} := \mu_{CR} \times (1 - LR) + LR \times (\frac{1}{n} \sum_{i:=1}^{n} SCR^G_i)$;
  Randomly removes solutions from A so $size(A) \leq N$;
end

4 Datasets

4.1 ALS Database

ALS is a rare neurodegenerative disease that induces a progressive degeneration of the neurons that innervate the muscles of the body, the motor neurons. Although studied since 1824 [13]. It was not until 1864 that Charcot, on the basis of his anatomical work carried out at the Pitié Salpêtrière Hospital (Paris, French), proposed the current name of the pathology and synthesized the work of his European colleagues. He established the link between the damage to the corticospinal bundle on post-mortem examination and the symptoms of the disease. ALS leads to a gradual loss of motor skills and dysfunctions in the bulbar sphere. There are no treatments to date to cure the disease. Survival from the onset of the first symptoms is, on average, between 3 and 5 years. Death often

occurs as a result of respiratory failure. High clinical variability and heterogeneity of disease progression complicate reliable prognostication. The incidence of ALS is approximately 2.5 per 100,000 population per year and the prevalence is approximately 8 per 100,000 population (ARSLA 2020)[3].

The ALS Therapy Development Institute (ALS TDI) estimates that approximately 450,000 people worldwide have ALS (ALS TDI 2020)[4]. Only two treatments have been approved by the U.S. Food and Drug Administration (FDA) to slow disease progression: riluzole (Bensimon et al. 1994) and edaravone (Takei et al. 2017). However, their effect on survival is limited, providing only a relative slowing of progression (Dharmadasa et al. 2018; Fang et al. 2018).

In our work on the prognosis of 1 year survival of patients with ALS[3], we primarily used the PRO-ACT database, an acronym for Pooled Resources Open-Access ALS Clinical Trials [14]. It includes twenty-two clinical trials and one observational study, conducted between 1990 and 2010. Funded in 2012 by the ALS Treatment Alliance, it was made available through the "DREAM Phil Bower ALS prediction Prize4Life" research competition. The PRO-ACT data has a sample to feature ratio of 765 when considering the seventeen features from the database. The size of the overall set, while significant for the domain, is not sufficient for complex model development.

Due to an imbalance in the number of patients in the different classes the score used for this specific dataset is $\frac{recall+specificity}{2}$. This metric takes this information into account, unlike accuracy, which is biased towards the largest class.

4.2 Benchmarks

In the study presented here we use also the following data sets :

- **Scene**: Scene recognition dataset from OpenML. It contains characteristics about images and their classes. The current dataset is a binary classification problem [15] (Instances: 2407; Features: 299)
- **Gravier**: Gravier et al. (2010) have considered small, invasive ductal carcinoma without axillary lymph node involvement (T1T2N0) to predict metastasis of small node-negative breast carcinoma. [16]. (Instances: 168; Features: 2 905)
- **Tian**: Tian et al. (2003) investigated the purified plasma cells from the bone marrow of control patients along with patients with newly diagnosed multiple myeloma. [17] (Instances: 173; Features: 12 625)

For these datasets we use accuracy for the scoring since the distribution between the classes is balanced.

[3] https://www.arsla.org/la-sla-en-chiffres
[4] https://www.als.net/als-resources/faq/

5 Results

We compare now the performance of different machine learning method presented above completed by 5 state-of-the-art filter-based methods Chi-squared (Chi2) test, Anova test, Mutual Information (MI) [18], ReliefF [19] and Maximum Relevance Minimum Redundancy algorithm (MRMR) [20] - applied to the different mentioned datasets without and with feature selection. The results obtained with these methods presented below are those with the best $k \in [1, D]$ number of features.

Table 1. The classification performance (%) between the algorithms

Dataset	Algorithm	LR	SVM	KNN	RF	GNB	Avg score	Max score	Rank
ALS	w/o FS	77.03	73.14	57.04	57.18	73.36	67.55	77.03	12
	ReliefF	77.78	76.12	65.44	78.94	77.30	75.12	78.94	10
	MRMR	78.60	78.63	63.31	79.98	78.34	75.77	79.98	7
	MI	78.73	77.02	65.43	79.67	76.86	75.54	79.67	9
	Chi2	77.71	76.42	64.01	70.24	76.54	72.98	77.71	11
	Anova	78.17	77.24	65.43	79.98	76.46	75.46	79.98	7
	TS	81.93	79.84	64.78	62.83	79.75	73.83	81.93	6
	SA	82.59	79.41	66.02	63.21	79.65	74.18	82.59	5
	PBIL	83.38	**81.04** (+7.9)	67.90	66.60	81.30	76.04	83.38	3
	PSO	83.30	80.33	67.07	66.36	80.87	75.59	83.30	4
	DE	84.26	80.53	69.88	86.60	82.03	80.66	86.60	2
	BPLDE	**84.42** (+7.39)	80.26	**71.06** (+14.02)	**86.67** (+29.49)	**82.20** (+8.84)	**80.92** (+13.37)	**86.67** (+9.64)	**1**
Scene	w/o FS	97.22	96.14	91.65	92.15	84.55	92.34	97.22	12
	ReliefF	97.47	97.55	92.52	93.69	85.75	93.40	97.55	11
	MRMR	97.47	97.71	95.68	94.97	87.08	94.58	97.71	10
	MI	97.80	98.13	96.51	94.68	85.67	94.56	98.13	9
	Chi2	98.92	98.92	98.92	**98.92** (+6.77)	86.95	96.53	98.92	7
	Anova	97.47	98.30	95.89	94.64	86.12	94.48	98.30	8
	TS	98.92	98.63	99.00	95.35	93.85	97.15	99.00	5
	SA	98.96	98.63	98.59	95.35	94.02	97.11	98.96	6
	PBIL	98.96	98.92	99.04	96.43	94.27	97.52	99.04	4
	PSO	99.09	98.84	99.04	96.55	93.73	97.45	99.09	3
	DE	99.09	98.88	99.13	96.51	**95.39** (+10.84)	97.80	99.13	2
	BPLDE	**99.17** (+1.95)	**99.84** (+8.19)	**99.21** (+7.06)	96.43	95.26	**97.98** (+5.64)	**99.84** (+2.62)	**1**
Gravier	w/o FS	72.62	73.21	67.86	68.45	70.24	70.48	73.21	12
	ReliefF	80.36	82.74	73.81	82.74	79.76	79.88	82.74	7
	MRMR	86.90	**86.90** (+13.69)	**84.52** (+16.66)	**84.52** (+16.07)	**83.33** (+13.09)	**85.23** (+14.75)	86.90	3
	MI	80.95	80.36	75.00	80.36	77.98	78.93	80.95	11
	Chi2	82.14	81.55	75.60	80.36	80.95	80.12	82.14	10
	Anova	82.14	83.33	77.98	82.14	80.36	81.19	83.33	6
	TS	79.76	82.74	71.43	80.95	75.60	78.10	82.74	7
	SA	77.98	82.74	71.43	79.76	75.60	77.50	82.74	7
	PBIL	86.31	85.12	75.60	77.98	78.57	80.72	86.31	4
	PSO	84.52	83.93	75.60	78.57	78.57	80.24	84.52	5
	DE	88.10	83.33	76.79	77.38	80.36	81.19	88.10	2
	BPLDE	**89.29** (+16.67)	84.52	77.38	77.98	79.76	81.79	**89.29** (+16.08)	**1**
Tian	w/o FS	73.41	77.46	78.61	79.19	80.35	77.80	80.35	12
	ReliefF	80.92	81.50	84.39	85.55	87.28	83.93	87.28	6
	MRMR	91.33	84.39	83.82	**86.71** (+7.52)	90.17	87.28	91.33	3
	MI	82.08	84.97	82.66	85.55	86.71	84.39	86.71	7
	Chi2	78.61	83.92	84.39	84.39	84.97	83.26	84.97	11
	Anova	80.92	84.97	83.24	86.71	86.13	84.39	86.71	7
	TS	79.77	84.97	83.24	83.24	86.13	83.47	86.13	10
	SA	79.19	86.13	82.66	82.66	86.71	83.47	86.71	7
	PBIL	91.33	**89.02** (+11.56)	86.71	81.50	90.17	87.75	91.33	3
	PSO	90.17	87.86	86.71	82.08	89.60	87.28	90.17	5
	DE	**94.22** (+20.81)	87.86	**87.28** (+18.67)	81.50	**90.75** (+10.4)	**88.32** (+10.52)	**94.22** (+13.87)	**1**
	BPLDE	93.64	88.44	86.71	82.08	**90.75** (+10.4)	**88.32** (+10.52)	93.64	2

The bold numbers in this table 1 indicate the best score obtained for each of the statistical learning or filtering methods used for a given data set (ALS, Scene, Gravel, Tian) with in parenthesis the performance delta between the feature

selection method and the performance when no feature selection is performed (w/o FS : first lines in the table).

These results show that using metaheuristics to select part of data feature leads to a better performance for all machine learning methods used.

6 Conclusion

We have shown in this paper that feature selection by metaheuristics improve significantly the performance of learning methods commonly used to build predictive models when the number of data is low and the number of characteristics is high (as in the medical field).

The extra cost related to feature selection comes mainly from the cost of cross-validation which leads to evaluate several times the performance of the model on parts of the initial dataset (up to $k = 5$ times longer than without feature selection). Nevertheless, the quality of the obtained model is much better in terms of performance (accuracy, recall or specificity) as our experiments show.

So, in view of this study, we recommend to systematically proceed to a feature selection by the DE or BPLDE metaheuristic whatever the method chosen to build a prediction model for a given dataset.

In our future works on this topic, we plan to use a statistical filter (e.g. MRMR) to initialize the first population of the DE or BPLDE metaheuristics in order to improve their efficiency or to study if smaller values of k for the cross-validation (which has an impact on the learning time) allow to keep as good results.

References

[1] Hiroshi Motoda and Huan Liu. "Feature Selection Extraction and Construction". In: 2002.

[2] Andrew John Chipperfield et al. "A. Carreira-perpiñán. a Review of Dimension Reduction Techniques. Technical". In.

[3] Vincent Grollemund et al. "Development and validation of a 1-year survival prognosis estimation model for Amyotrophic Lateral Sclerosis using manifold learning algorithm UMAP". In: *Scientific Reports* 10.1 (Aug. 2020), p. 13378. DOI: 10.1038/s41598-020-70125-8.

[4] Vincent Grollemund et al. "Manifold Learning for Innovation Funding: Identification of Potential Funding Recipients". In: *AIAI - 16th IFIP WG 12.5 International Conference, AIAI 2020, June 5-7, 2020, Proceedings, Part I*. Ed. by I. Maglogiannis, L. Iliadis, and E. Pimenidis. Vol. 583. IFIP Advances in Information and Communication Technology. Springer, 2020, pp. 119–127. DOI: 10.1007/978-3-030-49161-1_11.

[5] Ron Kohavi and George H. John. "Wrappers for feature subset selection". In: *Artificial Intelligence* 97.1 (1997). Relevance, pp. 273–324. ISSN: 0004-3702.

[6] Vincent Grollemund et al. “Machine Learning in Amyotrophic Lateral Sclerosis: Achievements, Pitfalls, and Future Directions”. In: *Frontiers in Neuroscience* 13 (2019), p. 135. DOI: 10.3389/fnins.2019.00135.

[7] Hongbin Zhang and Guangyu Sun. “Feature selection using tabu search method”. In: *Pattern Recognit.* 35 (2002), pp. 701–711.

[8] Majdi M. Mafarja and Seyed Mohsen Mirjalili. “Hybrid Whale Optimization Algorithm with simulated annealing for feature selection”. In: *Neurocomputing* 260 (2017), pp. 302–312.

[9] Shumeet Baluja. *Population-Based Incremental Learning: A Method for Integrating Genetic Search Based Function Optimization and Competitive Learning*. Tech. rep. CMU-CS-94-163. Pittsburgh, PA: Carnegie Mellon University, June 1994.

[10] Alireza Marandi et al. “Boolean Particle Swarm Optimization and Its Application to the Design of a Dual-Band Dual-Polarized Planar Antenna”. In: Jan. 2006, pp. 3212–3218. DOI: 10.1109/CEC.2006.1688716.

[11] Kingshuk Chakravarty et al. “Feature selection by Differential Evolution algorithm - A case study in personnel identification”. In: *2013 IEEE Congress on Evolutionary Computation*. 2013, pp. 892–899.

[12] Jingqiao Zhang and Arthur C. Sanderson. “JADE: Adaptive Differential Evolution With Optional External Archive”. In: *IEEE Transactions on Evolutionary Computation* 13.5 (2009), pp. 945–958.

[13] Lewis P. Rowland. “How Amyotrophic Lateral Sclerosis Got Its Name: The Clinical-Pathologic Genius of Jean-Martin Charcot”. In: *Archives of Neurology* 58.3 (Mar. 2001), pp. 512–515. ISSN: 0003-9942.

[14] Nazem Atassi et al. “The PRO-ACT database”. In: *Neurology* 83 (2014), pp. 1719–1725.

[15] Matthew R. Boutell et al. *Learning multi-label scene classification*. 2004.

[16] Gravier, Eleonore et al. “A prognostic DNA signature for T1T2 node-negative breast cancer patients.” In: *Genes, Chromosomes and Cancer* 49.12 (Sept. 2010), pp. 1125–1125.

[17] Erming Tian et al. “The Role of the Wnt-Signaling Antagonist DKK1 in the Development of Osteolytic Lesions in Multiple Myeloma”. In: *New England Journal of Medicine* 349.26 (Dec. 2003), pp. 2483–2494.

[18] Andrea Bommert et al. “Benchmark for filter methods for feature selection in high-dimensional classification data”. en. In: *Computational Statistics & Data Analysis* 143 (Mar. 2020), p. 106839. ISSN: 01679473. DOI: 10.1016/j.csda.2019.106839. (Visited on 09/26/2021).

[19] Marko Robnik-Šikonja and Igor Kononenko. “Theoretical and Empirical Analysis of ReliefF and RReliefF”. In: *Machine Learning* 53.1 (Oct. 2003), pp. 23–69. ISSN: 1573-0565. DOI: 10.1023/A:1025667309714.

[20] Hanchuan Peng, Fuhui Long, and C. Ding. “Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 27.8 (2005), pp. 1226–1238. DOI: 10.1109/TPAMI.2005.159.