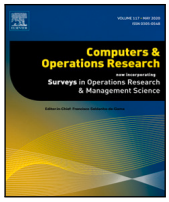
Computers & Operations Research 152 (2023) 106131

Contents lists available at ScienceDirect

Computers and Operations Research

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



Comparing two SVM models through different metrics based on the confusion matrix

Daniel Valero-Carreras, Javier Alcaraz *, Mercedes Landete

Departamento de Estadística, Matemáticas e Informática. Instituto Centro de Investigación Operativa, Universidad Miguel Hernández de Elche, Spain

ARTICLE INFO

Keywords:
Support vector machine
Feature selection
Multi-objective optimization
Metaheuristics

ABSTRACT

Support Vector Machines (SVM) are an efficient alternative for supervised classification. In the soft margin SVM model, two different objectives are optimized and the set of alternative solutions represent a Pareto-front of points, each one of them representing a different classifier. The performance of these classifiers can be evaluated and compared through some performance metrics that follow from the confusion matrix. Moreover, when the SVM includes feature selection, the model becomes hard to solve. In this paper, we present an alternative SVM model with feature selection and the performance of the new classifiers is compared to those of the classical soft margin model through some performance metrics based on the confusion matrix: the area under the ROC curve, Cohen's Kappa coefficient and the F-Score. Both the classical soft margin SVM model with feature selection and our proposal have been implemented by metaheuristics, given the complexity of the models to solve.

1. Introduction

The Support Vector Machine has proven to be a very effective tool for supervised classification. It was first introduced by Cortes and Vapnik (1995) as a way of keeping the value of the empirical risk fixed, minimizing the confidence interval. If only two classes of elements are considered, it looks for the optimal separating hyperplane, which is the intermediate hyperplane of two parallel hyperplanes, one lying above the elements of the first class and the other lying below the elements of the second class. Good separating hyperplanes are those such that the margin between the two associated parallel hyperplanes is large. If data is linearly separable, this margin is said to be hard. If a linear boundary is not feasible or misclassifications are allowed in the hope of achieving better generality, this margin is said to be soft. For the soft margin SVM problem, good separating hyperplanes are those having a large margin between the two parallel hyperplanes and a small distance between the misclassified vectors and the corresponding hyperplane. These two objectives are respectively known as the structural risk and the empirical risk. The paper by Burges (1998) constitutes a practical tutorial on SVM and the book by Vapnik (2013) is a comprehensive introduction to the relevance of learning theory for designing supervised classification.

One drawback that can be present when applying the SVM technique is the over-adjustment of the data. Feature selection can be applied to avoid it, thus conferring robustness to the solutions. Moreover, the selection of the most representative features allows reducing the size of the problem and therefore it adds operationality. The incorporation of feature selection to the SVM has been previously studied by several authors. To cite only some of them, Maldonado et al. (2014) introduce two SVM mixed integer models with feature selection which are later enhanced by Labbé et al. (2019); Aytug (2015) solves the SVM with feature selection with the help of Benders decomposition; Gaudioso et al. (2017) propose a Lagrangian relaxation approach for the SVM problem with feature selection; Benítez-Peña et al. (2019) propose an integer linear problem plus a quadratic convex problem.

Given that the soft margin SVM problem has two different objectives (structural risk and empirical risk), the output of the problem is not an optimal solution but the set of non-dominated solutions, that form the Pareto-front. Each one of these solutions represents a different classifier and therefore, having the complete Pareto-front provides the decision maker with a wide variety of alternative solutions from which the best classifier can be chosen.

Most of the methods developed to solve the soft margin SVM problem combine the two objectives of the problem into a single objective function, assigning a different weight to each of the objectives. However, when the number of selected features is fixed to p, the objective function for the soft margin SVM model is non-convex and the Pareto front cannot be obtained by just varying the compromise weight. In this way, different weight values could lead to the same solution. Moreover, the introduction of new parameters in the model leads to

* Corresponding author.
E-mail addresses: dvalero@umh.es (D. Valero-Carreras), jalcaraz@umh.es (J. Alcaraz), landete@umh.es (M. Landete).

https://doi.org/10.1016/j.cor.2022.106131
Received 11 April 2022; Received in revised form 15 December 2022; Accepted 22 December 2022
Available online 26 December 2022
0305-0548/© 2022 The Author(s). Published by Elsevier Ltd. This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/).

the fact that the accuracy of the method is highly dependent on them. Several metaheuristics have been developed in order to specify the model parameters and select the feature subset. In this way, once the model parameters and the selection of features have been determined by the metaheuristics, the SVM model, combining both objectives, is solved and one classifier is obtained. In recent decades, metaheuristics have become an efficient tool to solve very different optimization problems. The first set of classical metaheuristics were introduced between the early 70's and 90's, namely Genetic Algorithms (GA), Simulated Annealing (SA), Tabu Search (TS), Ant Colony Optimization (ACO) or Particle Swarm Optimization (PSO). Later, several improvements of these and new ideas were proposed to solve hard optimization problems. A good review and classification of metaheuristics can be found in Ezugwu et al. (2021). Throughout the work, we avoid references to "novel" metaphor-based metaheuristics which do not contribute to the field of metaheuristics, in the sense proposed in the works by Sörensen (2015) and Aranha et al. (2022).

We will now cite some of the metaheuristics used for determining the model parameters and selecting features. Huang and Wang (2006) and later Zhao et al. (2011) propose genetic algorithms which are also compared with others; García-Pedrajas et al. (2014) propose a memetic algorithm and Carrizosa et al. (2014) a nested heuristic; Raman et al. (2017) present an adaptive and a robust intrusion detection technique; Aladeemy et al. (2017) propose a variation of Cohort Intelligence algorithm; Bouraoui et al. (2018) propose a multi-objective approach to simultaneously optimize SVM parameters and feature subset using different kernel functions; Faris et al. (2018) present a multi-verse optimizer approach based on a robust system architecture; Candelieri et al. (2019) propose parallel global optimization for tuning the parameters of a water demand forecasting system; Dudzik et al. (2021) propose an evolutionary technique that efficiently classifies difficult datasets, including very large and extremely imbalanced cases Recently, the authors in Xue et al. (2021) have proposed a novel multiobjective metaheuristic for a 3-objective classification problem. They obtain the Pareto-front when the objectives are to maximize the margin between the two associated parallel hyperplanes, minimize the number of selected features and minimize the amount of missing data.

Most of the previous methods select optimal features and optimize the parameters of SVM simultaneously with the aim of reducing the number of features while maintaining the prediction accuracy. However, none of the methods presented above give the optimal Pareto-front of the soft margin SVM problem with feature selection or, at least, an approximation of it. Alcaraz et al. (2022) design an efficient metaheuristic tool for obtaining the Pareto-front of the soft margin SVM model with feature selection and they compare the frontiers given by the metaheuristic and by an exact method. They also prove that exact methods fail when obtaining the front in medium or large-sized problems. As far as we know, this method is the only one developed to build the Pareto-front of the soft margin SVM model with feature selection.

Once we have all the alternative solutions to the soft margin SVM model with feature selection, formed by the front, if the aim is to select the best classifier, comparison can be performed through metrics based on the confusion matrix, such as the area under the Receiver Operating Characteristic (ROC) curve, Cohen's Kappa coefficient or the F-Score. A ROC curve is a technique for organizing, visualizing and selecting classifiers based on their performance (Fawcett, 2006). Nowadays, ROC analysis is an extended technique and is employed in a wide variety of fields, such as engineering (Ierimonti et al., 2021), economics and finance (Yang et al., 2022), mathematics (Vijayan et al., 2021), business and decision science (Florian et al., 2021), computer science (Ponmalar and Dhanakoti, 2022), medicine (Halder et al., 2022), agriculture (Loh et al., 2022) or physics (Kwiatkowski and Sotor, 2022) to cite just a few. The Cohen's Kappa coefficient and the F-Score are another two widely used metrics based on the confusion matrix.

In this paper, we propose an alternative model to the soft margin SVM problem with feature selection. This model considers two different objectives instead of the ones proposed in the classical model. Then, we propose a metaheuristic in order to obtain the Pareto-front of the new model. ROC analysis, as well as other performance metrics, have been employed to compare the different classifiers given by the metaheuristic and select the best. On comparing the classifiers given by the classical model and the new model proposed, results show that the latter outperforms the former.

This paper is organized as follows; Section 2 presents the classical soft margin SVM model with feature selection along with the new proposal. In Section 3, different metrics that can be considered to compare classifiers based on the confusion matrix are presented. Section 4 describes the metaheuristic designed to approximate the solutions of the new model and Section 5 presents the results of this comparison. Finally, we present the conclusions and future lines of research.

2. SVM models with feature selection

In a training set Ω, the elements (usually called vectors) are partitioned into two classes and represented by a pair $(x_i, y_i) \in \mathbb{R}^n \times \{-1, 1\}$, where n is the number of features observed for each vector, x_i contains the feature values for vector i and y_i indicates to which of the two classes of Ω vector i belongs. The soft margin support vector machine problem consists in determining the hyperplane $f(x) = w^T \cdot x + b$ that separates the vectors in the training set in such a way that the distance between two parallel hyperplanes supporting some vector of the two classes is maximized and the sum of classification errors is minimized. The soft margin SVM model analyzed in Bradley and Mangasarian (1998) minimizes a compromise between the two aforementioned objectives, known respectively as the structural risk and the empirical risk.

$$\min \quad \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{m}\xi_i$$

$$\text{s.t.} \quad y_i(w^T x_i + b) \geq 1 - \xi_i \qquad i = 1, \dots, m, \tag{1}$$

$$\xi_i \geq 0 \qquad i = 1, \dots, m, \tag{2}$$

$$w_j \in \mathbb{R} \qquad j = 1, \dots, n, \tag{3}$$

$$b \in \mathbb{R} \tag{4}$$

Variables w_j and b represent the coefficients of the two parallel hyperplanes $w^T x + b = 1$ and $w^T x + b = -1$. The first term in the objective function $1/2\|w\|^2$ is the structural risk, since $\|w\|$ is twice the inverse of the distance between these two hyperplanes. The second term in the objective function $\sum_{i=1}^{m}\xi_i$ is the empirical, i.e., the sum of the deviation of misclassified objects. Decision makers make use of parameter C, that regulates the trade-off between the two objectives. Constraints in the model ensure that either vectors i in the class represented by $y_i = 1$ satisfy $(w^T x_i + b) \geq 1$ and vectors in class $y_i = -1$ satisfy $(w^T x_i + b) \leq -1$ or variable ξ_i takes a positive value.

When feature selection is incorporated, the model becomes harder and non-convex. For all $j \in \{1, \dots, n\}$, let t_j be a binary variable taking the value of 1 if and only if feature j is selected. In order to be restricted to the feasible solutions of the previous model that use p features, it is necessary to add the following constraints:

$$\sum_j t_j = p \tag{5}$$

$$|w_j| \leq M t_j \qquad j = 1, \dots, n \tag{6}$$

$$t_j \in \{0, 1\} \qquad j = 1, \dots, n, \tag{7}$$

where M is a large constant. If $t_j = 1$, constraint $|w_j| < M$ is redundant because M is large. If $t_j = 0$, i.e., feature j is not selected and constraint (6) states that feature j does not belong to the model. Therefore, obtaining the Pareto-front by combining the two objectives into a

Table 1
Feasible solutions to the soft margin SVM model with feature selection for the example in Fig. 1.

Fig.	w_1	w_2	b	ξ_1	ξ_2	ξ_3	ξ_4	ξ_5	ξ_6
1(a)	0.0606	0.4848	−1.5455	0	1.0303	0	0	0	0
1(b)	0	2	−7	0	0	0	0	0	0
1(c)	0.3333	−0.6667	0.3333	1	0	0	0	0	3.3333

single one and assigning different values to the C parameter, is for some instances impossible. The model for considering both objectives separately reads as follows:

$$\min \quad O_{11} = \|w\|^2$$

$$\min \quad O_{12} = \sum_{i=1}^{m} \xi_i$$

s.t. (1)–(7)

The non-dominated solutions of the previous model form the Pareto-optimal front and several performance metrics can be used to select the best classifier.

We propose an alternative bi-objective model to solve the soft margin SVM problem with feature selection that, as we will demonstrate later, provides better classifiers when certain performance metrics are compared. The objectives are to minimize both the False Negative predictions (FN) and the False Positive prediction (FP). The model can be formulated in the following way. Variable $z_i \in \{0, 1\}$ takes the value of one if i is a false negative, i.e,. the vector i is *positive* ($y_i = 1$) and $(w^T x_i + b) \leq 0$ or if i is a false positive, i.e., the vector i is negative ($y_i = -1$) and $(w^T x_i + b) \geq 0$.

$$\min \quad O_{21} = \sum_{i:\, y_i = 1} z_i$$

$$\min \quad O_{22} = \sum_{i; y_i = -1} z_i$$

s.t. (1)–(7)

$$\xi_i - 1 \leq M z_i \qquad i = 1, \ldots, m, \tag{8}$$

$$z_i \in \{0, 1\} \qquad i = 1, \ldots, m. \tag{9}$$

The first objective function is the number of false negatives and the second is the number of false positives, i.e., $O_{21} = FN$ and $O_{22} = FP$. Constraints (8) state that Variable z_i takes the value of one when variable $\xi_i \geq 1$. If $\xi_i \leq 1$, vector i is among the two parallel hyperplanes $(w^T x_i + b) = 1$ and vectors in class $y_i = -1$ satisfy $(w^T x_i + b) = -1$ but it is well classified.

Let us illustrate a comparison of the two models through the example used in Alcaraz et al. (2022).

Example 1. Let Ω be the set of vectors in Fig. 1. Let us assume that vectors represented in red are the vectors with $y_i = -1$ and vectors represented in green are the vectors with $y_i = 1$. Table 1 gives three feasible solutions to the soft margin SVM model with feature selection and Table 2 gives the associated objective values. Points 1 and 2 are misclassified or not with respect to hyperplane π_1 while points 3, 4, 5, and 6 are misclassified or not with respect to the hyperplane π_2. The third vector is efficient for the two objectives O_{21} and O_{22} while it is not for the objectives O_{11} and O_{12}. The green line corresponds to the first row in Table 1, the red line corresponds to the second row in Table 1 and the blue line to the third row. The margin of each hyperplane is represented by dashed lines.

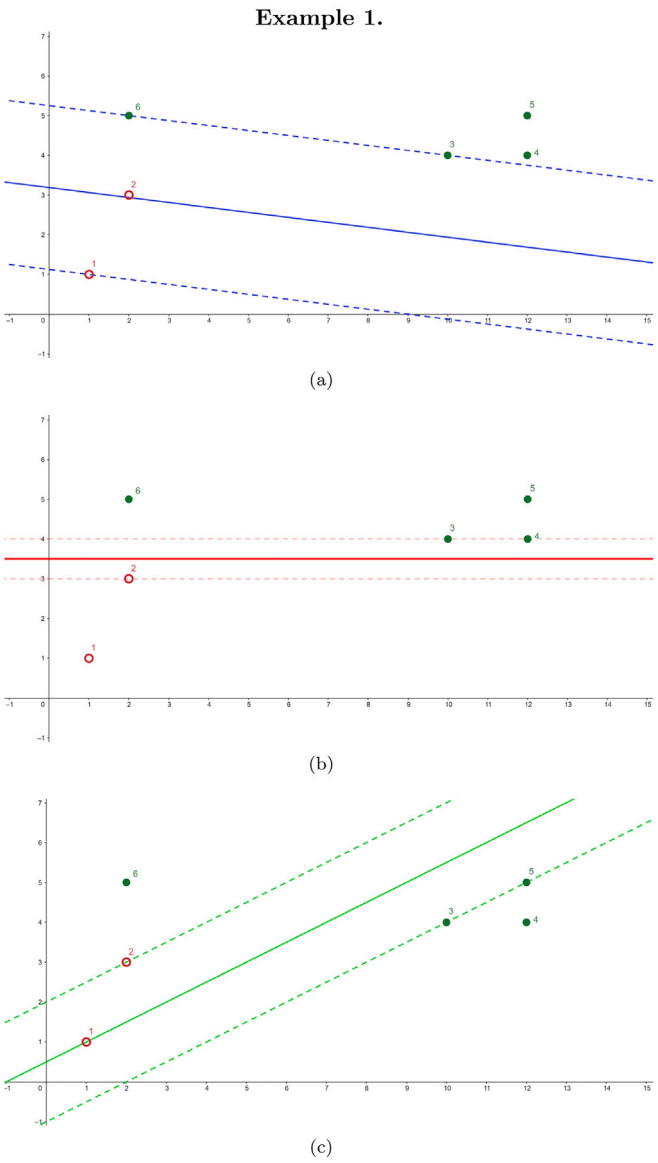


Fig. 1. Example with $m = 6$ and $n = 2$. Red color indicates class "−1", green color indicates class "1".

Table 2
Objective values for the example in Fig. 1.

O_{11}	O_{12}	O_{21}	O_{22}
0.216	1.022	0	1
4	0	0	0
0.444	4.057	1	0

3. Performance metrics based on the confusion matrix for comparing different objective functions

In the literature, we can find several measures for evaluating the performance of a machine learning technique. The evaluation of the

		Actual values	
		+	-
Predicted values	+	True positive(TP)	False positive(FP)
	-	False negative(FN)	True negative (TN)

Fig. 2. Confusion matrix.

performance is used in very different fields like medicine (Ismael and Şengür, 2021) or engineering (Roshani et al., 2021), to cite just a few. In binary classification problems, as Support Vector Machines (SVM), it is common to use a confusion matrix (see Fig. 2) which represents the prediction of the algorithm versus the real values. Based on these values it is possible to establish different measures which allow the quality of a model to be evaluated (Gu et al., 2009). Some of these measures are the following.

- Sensitivity is the model's capability of predicting positive cases. It is calculated as the proportion between truly positive predictions and the set of all the positive predictions.

$$Sensitivity = \frac{TP}{TP + FN} \tag{10}$$

- Specificity is the model's capability of predicting negative cases. It is the proportion between truly negative predictions and the set of all the negative predictions.

$$Specificity = \frac{TN}{TN + FP} \tag{11}$$

- Precision is a measure used to evaluate the relevant instances among the retrieved instances. It is calculated as the proportion between truly positive predictions and the set of all the real positive values.

$$Precision = \frac{TP}{TP + FP} \tag{12}$$

- Accuracy is the proportion of correct predictions. It measures how well a classify predicts a condition.

$$ACC = \frac{TP + TN}{N} \tag{13}$$

- F-Score is a metric that takes into account Precision and Sensitivity in the same way. It is the harmonic mean between precision and sensitivity.

$$FSC = \frac{Precision * Sensitivity}{Precision + Sensitivity} \tag{14}$$

- Area Under the Curve (AUC) is a measure used to compare different classifiers. This statistic measures the area generated by the ROC curve in the ROC space. The ROC curve represents the sensitivity versus the specificity according to a discrimination threshold. In the ROC space, it is represented 1-specificity on the X axis and sensitivity on the Y axis. Models which are near to the point (0,1) are better classifiers since they present higher values of AUC (see Fig. 3).

$$AUC = \frac{Sensitivity * Specificity}{2} \tag{15}$$

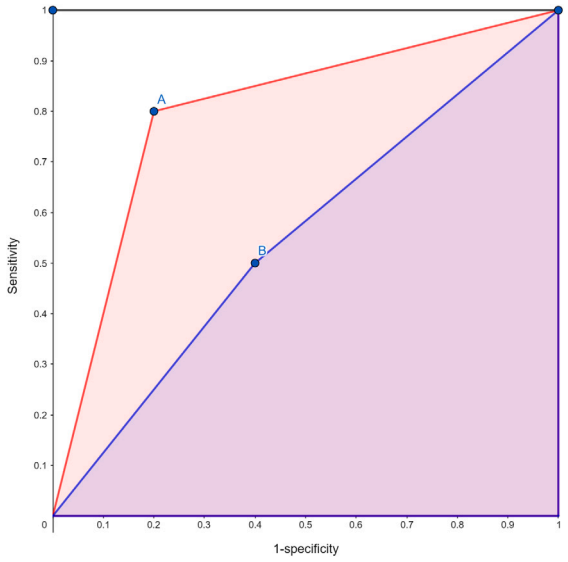


Fig. 3. AUC for two classifiers.

- Cohen's Kappa (KAP) (Cohen, 1960) is a statistic that measures the agreement or disagreement between two classifiers. To do this, the probability of agreements that are not given by chance (ACC-P_c) and the probability of disagreement by chance (1-P_c) are calculated. The probabilities taken into account for this measure are the observed agreement (ACC) and the probability that two classifiers choose the same one by chance (P_c). Higher values of these measures indicate better classifiers.

$$P_c = \frac{(TP + FN) * (TP + FP) + (TN + FN) * (TN + FP)}{N^2} \tag{16}$$

$$KAP = \frac{ACC - P_c}{1 - P_c} \tag{17}$$

The previous metrics have been calculated for the three solutions presented in Fig. 1 and have been presented in Table 3. These measures will be employed to compare the different classifiers provided by the metaheuristics used to solve the SVM with feature selection.

4. Metaheuristics for soft margin SVM with feature selection

Soft margin SVM with feature selection is a hard optimization problem and Alcaraz et al. (2022) demonstrated that exact techniques fail to approximate the optimal Pareto-front in many real instances. Therefore, in order to obtain that frontier, the authors proposed a multi-objective metaheuristic implementing this bi-objective model and demonstrating its ability to build a good approximation of the optimal Pareto-front in a wide variety of size instances. Although there are other works that propose metaheuristics for solving the soft margin SVM with feature selection, as far as we know, all of them implement the model with only one objective function in which both objectives, the structural risk and the empirical risk, are combined. In these cases, the metaheuristics are used to determine the parameters of this model and not to solve the model itself. Then, once these parameters have been established, the classical soft margin SVM model with only one objective is solved through single-objective exact techniques. The main advantage of multi-objective optimization, which results in an efficient frontier, is the potential of the efficient frontier. All the solutions of the approaches with a single objective, which is a linear combination of the multiple objectives, are contained in the efficient frontier of the

Table 3
Performance measures for the example in Fig. 1.

FN	FP	TP	TN	SEN	SPE	PRE	ACC	FSC	AUC	KAP
0	1	3	2	1	0.5	0.75	0.833	0.429	0.25	0.667
0	0	4	2	1	0.5	1	1	0.5	0.25	1
1	0	3	2	0.8	0.67	1	0.833	0.429	0.25	0.667

multi-objective problem, so any decision that is adopted optimizing one objective is an option when it comes to optimizing several. And what is more important, among all the efficient solutions of the multi-objective problem, the decision maker can find novel solutions for his problem that he would not have considered otherwise.

The computational cost of multi-objective optimization versus single-objective optimization is high. While the optimal value of a single-objective problem is unique, the number of efficient values in a multi-objective problem is large. Since the computational cost of finding the optimal solution of the problem with one objective is similar to the computational cost of finding an efficient solution of the problem with several objectives, the computational cost of the problem with several objectives can be assumed to be similar to the product of the computational cost of an efficient solution by the number of efficient solutions. This fact implies that multi-objective problems are often dealt with through a heuristic as opposed to an exact approach.

Multi-objective evolutionary algorithms (MOEAs) have become an alternative to exact approaches to solve difficult optimization problems when several objectives are considered and these techniques require an excessive computation time. One of the most used is the Non-dominated Sorting Genetic Algorithm, NSGA-II (Deb et al., 2002) that has been successfully applied to find a diverse set of solutions in very different types of optimization problems with several objectives. NSGA-II improves the performance of its predecessor NSGA (Srinivas and Deb, 1995), which was one of the first MOEAs. NSGA-II combines ideas of the GAs with domination concepts of multi-objective optimization. Some other MOEAs have also been successfully used to different multi-objective problems and we can cite, for example, the Strength Pareto Evolutionary Algorithm (SPEA-2) (Kim et al., 2004) or the Pareto Archive Evolutionary Strategy (PAES) (Knowles and Corne, 2000).

In order to compare the classifiers given by the classical model with those resulting from the new model we propose, we have used the metaheuristic proposed by Alcaraz et al. (2022) which has been employed to obtain the results of the classical soft margin SVM model with feature selection. The metaheuristic is based on the NSGA-II and is presented, in pseudocode, in Algorithm 1. Although the operation of this algorithm is explained in detail in the cited paper, it can be summarized as follows. First, a population of a fixed size (N) of individuals is created randomly. Each one of these individuals is evaluated, i.e., the two objectives are computed and then the solutions are grouped depending on domination criteria. The individuals which are not dominated by any other in the population form the first front (F_1), the individuals dominated by one or more solutions of F_1 form F_2, and so on. Then, the evolution process starts and is carried out until the stopping criterion, which is often related to the computation time or the number of solutions evaluated, is satisfied. First, crossover and mutation are employed to generate a new population size N. Two solutions, a mother and a father, are chosen using again domination criteria, to be recombined to generate a new solution which can be also affected by mutation. The individuals in the new population are also evaluated. Then, the original and the new populations are merged in a double sized population and the individuals are sorted again. The last step consists in reducing the size of the current population from $2N$ to its original size, N. To do that, we select the individuals, by order, from the consecutive fronts, F_1, F_2, ...When all the individuals of a given front cannot be placed in the new population, these are sorted with a distance criterion, and only the best are selected to form part of the new population.

Algorithm 1:

P_0 = **create_initial_population**(N);
$t = 0$;
P_0 = **evaluate_population**(P_0);
F = **fast_non_dominated_sort**(P_0);
while *not stopping_criterion* **do**
  for $j = 1$ *to* N **do**
    (mother, father)=**tournament_dominance_selection**(P_t);
    $Q_t[j]$=**crossover**(mother, father);
    if ***random()*** $\leq PMUT$ **then**
      $Q_t[j]$=**mutation**($Q_t[j]$);
  Q_t = **evaluate_population**(Q_t);
  $R_t = P_t \cup Q_t$;
  F = **fast_non_dominated_sort**(R_t);
  $P_{t+1} = \emptyset$;
  $i = 1$;
  while $|P_{t+1}| + |F_i| \leq N$ **do**
    $P_{t+1} = P_{t+1} \cup F_i$;
    $i = i + 1$;
  if $|P_{t+1}| < N$ **then**
    i_distance_assignment(F_i) ;
    sort(F_i, i_distance);
    $P_{t+1} = P_{t+1} \cup F_i[1 : (N - |P_{t+1}|)]$;
  $t = t + 1$;
return F_1

This algorithm has been used as a base for the new metaheuristic implemented to obtain the classifiers derived from the new model proposed. The operation of both algorithms is identical, except for the two objectives considered. This difference completely changes the results given by both algorithms. For the purposes of comparison, that proposed by Alcaraz et al. (2022) is referred to as Algorithm 1 and the new one as Algorithm 2. The objectives considered are:

- Algorithm 1: The first objective consists in the maximization of the distance between the two parallel hyperplanes supporting some vectors of both classes, O'_{11}. The second objective is to minimize the sum of the distances of the misclassified vectors to the corresponding hyperplane, O'_{12}. Values O'_{11} and O'_{12} are proportional to values O_{11} and O_{12} respectively. In Alcaraz et al. (2022), the authors analyze the close relationship between these functions.
- Algorithm 2: Both objectives consist of minimization. The first minimizes the number of misclassified vectors in one of the classes (O_{21}) and the other the sum of misclassified vectors in the other class (O_{22}).

Following Alcaraz et al. (2022), solutions in the metaheuristic are encoded with a structure comprising three components: (i) mode: two possible values. Mode=A indicates that we have p vectors in Ω in class

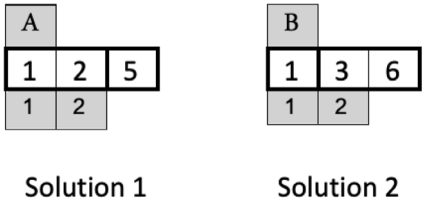


Fig. 4. Two different solutions encoded by metaheuristics for Example 1.

Table 4
Values of the objectives for the solutions in Fig. 4 in both metaheuristics.

	Algorithm 1		Algorithm 2	
	O'_{11}	O'_{12}	O_{21}	O_{22}
Solution 1	1.61	2.06	0	1
Solution 2	1.95	1.00	1	0

−1 and one in class 1; mode=B indicates that hyperplanes are built with one vector in class −1 and p vectors in class 1. (ii) vectors: an array with the indices of the $p+1$ vectors used to build the two hyperplanes. The first position/s in the array will be occupied with the vector/s in class -1 and the following position/s with the vector/s in class 1. (iii) features: an array with the indices of the p features selected in the solution. The selected features consists of an unordered list of a subset of the complete set of features.

Most metaheuristics dealing with feature selection employ a binary mask chromosome with n genes, one for each observed feature (Huang and Wang, 2006; García-Pedrajas et al., 2014; Aladeemy et al., 2017; Bouraoui et al., 2018; Faris et al., 2018) indicating whether or not the corresponding feature is selected. However, in the problem considered in this work, the number of selected features ($p \leq n$) is an input of the problem. In that sense, using the binary mask would require n genes whereas this encoding uses only p and therefore we think the encoding employed in this case is more adequate.

In Fig. 4, we present two different solutions encoded by the metaheuristics (Algorithm 1 and Algorithm 2) for the dataset presented in Fig. 1. In Solution 1, the first hyperplane supports vectors 1 and 2 (class -1), and the second supports vector 5 (class 1) and is parallel to the first one. In Solution 2 (that matches the presented in Fig. 1a), the first hyperplane to be built is that supported by vectors in class 1, vectors 3 and 6 in this case, and the second hyperplane supports vector 1 (class -1) and is parallel to the first one. For each solution, once the two hyperplanes are built, the separating hyperplane, the intermediate hyperplane of two parallel hyperplanes, represents the classifier and the objectives can be computed. In Table 4, we have calculated the values of the objectives of the corresponding metaheuristic.

5. Computational experiment

In order to compare the performance of the classifiers provided by both metaheuristics, Algorithm 1 and Algorithm 2 under three performance metrics based on the confusion matrix (FSC, AUC and KAP), we have carried out a computational experience solving different datasets. Both metaheuristic algorithms have been implemented in C++ and the experiments were carried out in the Scientific Computer Cluster of the Miguel Hernandez University (UMH). This computer equipment uses CentOS Linux release 7.5.1804 operating system which is composed of five nodes with the following characteristics. Node 0: Bull X410 model with two processors Xeon Planitum v5 8160–2.1 GHz (in total 96 cores), 384 GB of RAM and four Nvidia V100 GPUs. Node 1: similar to Node 0 with 768 GB of RAM. Node 2 and 3: Dell PowerEdge R440 model with two processors Xeon Gold 5120 @ 2.2 GHz (in total 56 cores) and 128 GB of RAM. Node 4: Supermicro SYS-1029GQ-TRT model with two processors Xeon Gold 6230N @ 2.3 GHz (in total 80 cores), 640 GB of RAM and three GPUs (one NVidia V100 and two NVidia A100). Simulations used SLURM Workload Manager assigning one core to each dataset.

Table 5
Characteristics of the datasets.

Instance	#*vectors* (m)	#*features* (n)
Housing	506	13
GC	1000	24
WBC	569	30
Iono	351	33
Arrhythmia	420	278
Madelon	2600	500
Isolet	600	617
Mfeat	2000	649
Gina_agnostic	3468	970
Bioresponse	3751	1776
Gisette	6000	4952
Duke	44	7129
Arcene	200	10000

For our computational experiments, we considered several datasets with different characteristics. Arcene, Arrhythmia, GC, Gisette, Housing, IONO, Isolet, Madelon, MFeat and WBC datasets can be found in the UCI repository (Dua and Graff, 2017). The Duke datased can be found in LIBSVM library (Chang and Lin, 2011), Gina_agnostic is availed in the work by Guyon et al. (2007) and Bioresponse was used by Abdunabi and Basir (2014). The main characteristics of the datasets can be seen in Table 5. The first column shows the name of the instance. The second column indicates the number of vectors in the sample. Finally, the last column shows the number of features. These datasets represent a diverse set of data, as they come from very different contexts and have very different characteristics, as regard the number of vectors and features.

We have used two different population sizes, $N = 100$ and $N = 50$ and three different levels of CPU time, measured in seconds, as stopping criterion, $t = 1200$ s, $t = 3600$ s and $t = 10800$ s. The combination of these two parameters gives us 6 different versions of each algorithm and each instance has been solved with all the versions of both algorithms. All other parameters of the metaheuristics have been set to the values used by Alcaraz et al. (2022), given that they carried out several experiments to tune up the algorithm. However, the authors explain that the performance of the metaheuristics is not greatly influenced by slight variations in the values of these parameters.

To solve one instance with one of the versions of one of the metaheuristics we have carried out a stratified hold-out with repetition procedure. We have generated 5 different splits with 2/3 of the vectors which correspond to the 5 train datasets, $Train_1, \ldots, Train_5$. For each training dataset $Train_i$, the rest of the vectors in the original dataset not included in the train set, form the corresponding test train set, $Test_i$. In all these splits, train and test sets, the percentage of vectors in each class is maintained as in the original dataset. For each one of the splits, we have run the algorithm 3 independent times, obtaining a set of non-dominated solutions per run. The number of non-dominated solutions can vary from one run to another. Each of these solutions represents a classifier. The classifiers obtained in one run of the metaheuristic, have been tested with the corresponding test set. The testing of a classifier consists of calculating each one of the measures considered to make a comparison from different perspectives. These measures are the Area Under the Curve (AUC), F-Score (FSC) and Cohen's Kappa measure (KAP). Therefore, for each run we have calculated three different measures. Given a measure, the best value for that measure is selected from each run. Then, we calculate the average and the maximum of these three values, as representative of the split. Therefore, we have 5 different maximum and 5 different averages for each one of the

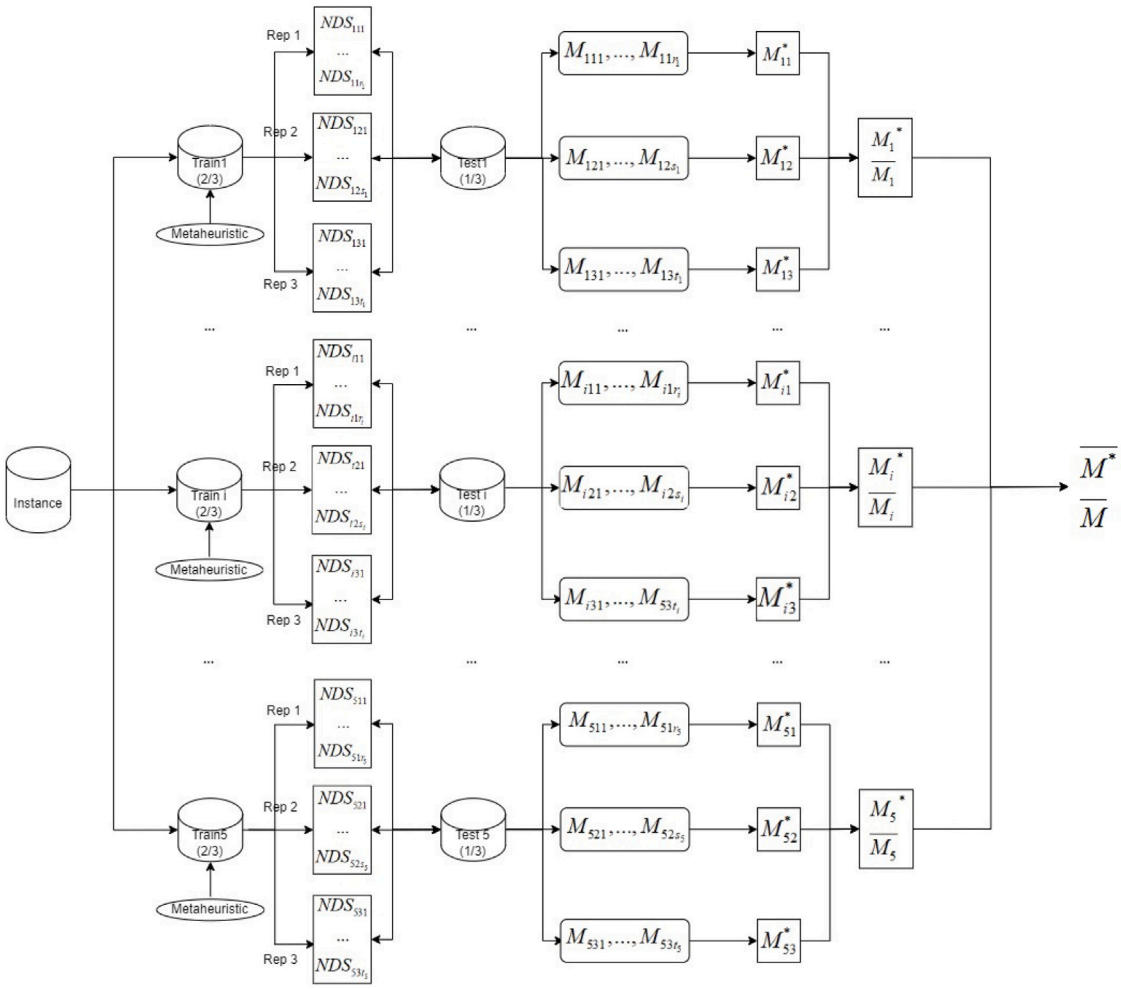


Fig. 5. Stratified hold-out with repetition.

measures. Finally, the average of each set of 5 values gives us the two scores that a given metaheuristic obtains when solving an instance. The procedure which calculates the scores of measure M for an instance and a given version of the metaheuristic has been represented in Fig. 5.

5.1. Results

In this section, we present the results of the computational experiment carried out to compare the classifiers derived from the resolution of the two SVM models presented before and which has been explained in the previous section. The cpu time needed to evaluate each one of the datasets by each of the two algorithms is 130 h and therefore the complete computational experiment took up 3380 h (140.8 days) of cpu time.

Tables 6, 7, 8 and 9 show the detailed computational results for the three measures of interest (AUC, FSC and KAP) for the different datasets. The first two tables correspond to the datasets having 500 features or less and the last two tables to datasets having a larger number of features. The results for Algorithm 1 are presented in Tables 6 and 8 and the results for Algorithm 2 in Tables 7 and 9. The same three measures are recorded when executing both algorithms. The same dataset is solved with six different combinations of time limit (column TIME) and genetic algorithm population size (column SIZE). Since for each combination of algorithm, time and size we have ten different values for the three measures (five averages and five maxima for each training dataset), we have recorded both the average of the five averages ($\bar{M}$) and the average of the five maxima ($\bar{M}^*$).

A bold number in Table 6 (Table 7) indicates that this value is better for Algorithm 1 (2) than for Algorithm 2 (1). If a value is in bold in Table 6, it is not in bold in Table 7 and vice-versa. For example, the values in the first row of Tables 6 and 7 are the results for the dataset Arrhythmia when the time limit is set to 1200 s and the population size of the genetic algorithm is set to 50. In this case: the average AUC measure for the five training datasets is 0.708 for Algorithm 1 (Table 6) and 0.750 for Algorithm 2 (Table 7); the average FSC measure is 0.792 for Algorithm 1 and 0.811 for Algorithm 2; the average KAP measure is 0.430 for Algorithm 1 (Table 6) and 0.508 for Algorithm 2 (Table 7); the best (average of the five maxima of the training datasets) AUC measure is 0.721 for Algorithm 1 and 0.753 for Algorithm 2; the average of the best (maximum) FSC measure 0.799 for Algorithm 1 and 0.814 for Algorithm 2; the average of the best (maximum) KAP measure is 0.452 for Algorithm 1 and 0.514 for Algorithm 2. The remaining rows give the same values for the rest of datasets, time limit and size combinations. Thus, the tables provide a large number of values that are used to compare the goodness of the two algorithms. That an algorithm is better in average values does not mean that it is better in maximum values, nor that an algorithm is better for a combination of

Table 6
Computational results for Algorithm 1 in datasets with $n \leq 500$.

	TIME	SIZE	$\overline{M}$			$\overline{M}^*$		
			AUC	FSC	KAP	AUC	FSC	KAP
Arrhythmia	1200	50	0.708	0.792	0.430	0.721	0.799	0.452
	1200	100	0.748	0.814	0.511	**0.765**	0.825	**0.546**
	3600	50	0.731	0.801	0.475	0.744	0.809	0.500
	3600	100	0.739	0.812	0.493	0.748	0.817	0.512
	10800	50	0.720	0.800	0.450	0.733	0.813	0.484
	10800	100	0.741	0.809	0.493	0.751	0.815	0.513
GC	1200	50	0.696	0.575	**0.389**	0.705	0.587	**0.407**
	1200	100	0.701	0.581	**0.410**	0.710	0.594	**0.424**
	3600	50	0.681	0.552	**0.374**	0.693	0.573	**0.392**
	3600	100	0.706	0.587	0.413	0.712	0.597	**0.427**
	10800	50	0.687	0.559	**0.382**	0.703	0.581	**0.405**
	10800	100	0.704	0.584	0.411	0.707	0.589	0.417
Housing	1200	50	0.844	0.846	0.687	0.852	0.854	0.703
	1200	100	0.851	0.850	0.702	0.860	0.859	0.720
	3600	50	0.836	0.839	0.672	0.844	0.845	0.687
	3600	100	0.856	0.855	0.712	0.861	0.859	0.720
	10800	50	0.841	0.845	0.682	0.853	0.857	0.706
	10800	100	0.854	0.859	0.708	0.866	0.869	0.732
IONO	1200	50	0.856	0.913	0.736	**0.863**	**0.918**	**0.749**
	1200	100	0.866	**0.923**	**0.763**	**0.876**	**0.927**	**0.779**
	3600	50	0.836	0.905	0.705	0.851	0.912	0.732
	3600	100	0.850	0.915	0.734	0.862	0.920	0.752
	10800	50	0.858	0.916	0.740	**0.876**	**0.923**	**0.766**
	10800	100	0.855	0.917	0.740	0.862	**0.921**	**0.757**
Madelon	1200	50	0.531	0.646	0.061	0.535	0.660	0.069
	1200	100	0.530	0.664	0.060	0.537	0.667	0.073
	3600	50	0.530	0.663	0.060	0.537	0.668	0.074
	3600	100	0.531	0.648	0.063	0.535	0.662	0.071
	10800	50	0.529	0.656	0.059	0.538	0.667	0.076
	10800	100	0.548	0.639	0.096	0.568	0.643	0.136
WBC	1200	50	**0.974**	**0.966**	**0.946**	**0.980**	**0.974**	**0.959**
	1200	100	**0.975**	**0.967**	**0.947**	**0.976**	**0.968**	**0.948**
	3600	50	0.973	0.964	0.942	0.974	0.965	0.944
	3600	100	0.970	0.961	0.937	**0.974**	**0.964**	**0.941**
	10800	50	0.968	0.960	0.936	0.970	0.962	0.939
	10800	100	**0.976**	**0.967**	**0.947**	**0.976**	**0.968**	**0.948**
Average			0.772	0.793	0.552	0.781	0.801	0.568

time and size does it imply that it is better for another, even if we keep the instance. The last row shows the average for the different metrics calculated for all the datasets presented in the table. If we compare the average for the different metrics shown in Tables 6 and 7, we realize that, for the datasets with 500 features or less Algorithm 2 gives, in general terms, better classifiers than Algorithm 1, regardless of the metric considered. For example, the average AUC for Algorithm 1 is of 0.77 in contrast to the 0.79 that Algorithm 2 provides. The results given by Algorithm 2 are, for all the metrics considered, better than those presented by Algorithm 1 in all the datasets in the first group (those with 500 features or less).

The results shown in Tables 8 and 9 allow the comparison of both algorithms when solving the datasets with more than 500 features. Again, bold numbers indicate better performance. If we focus again on the last row of these tables, we realize that Algorithm 2 provides better classifiers than Algorithm 1 when solving the datasets with more than 500 features, regardless of the metric considered. In this case, the differences between both algorithms seem to be larger than those for the datasets with a lower number of features. Therefore, from the analysis of these tables we can conclude that, in all the datasets considered, Algorithm 2 generally performs better that Algorithm 1 for all the metrics calculated. However, the density of these four tables makes it difficult to draw quick conclusions. In order to facilitate the interpretation of the results, we have calculated a table of success frequencies similar to a pool and these are presented in Table 10, where each row corresponds to a combination of time and population size, the first row is for 1200 s and a population size of 50, the second row is for 1200 s and population size equal to 100, …, the last is for 10800 s and 100 individuals. The columns are grouped into two blocks, the block on the left compares the two algorithms in terms of the average values of AUC, FSC and KAP obtained in the different executions of the heuristics and the block on the right compares the two algorithms in terms of the average of the best solution obtained among the different executions. Within each block of columns, column "1" indicates the number of times Algorithm 1 outperforms Algorithm 2 and column "2" indicates the number of times Algorithm 2 outperforms Algorithm 1. All comparisons have been taken into account with the rounding to three decimal places shown in Table 10. Obviously, for each measurement (AUC, FSC and KAP) the sum of the values in the row is equal to 13 (except in the cases where both algorithms perform exactly the same): there are 13 datasets and the number of times that one algorithm is better than the other plus the number of times that it is worse must be the number of datasets.

For example, the last row of the main block in Table 10 shows the comparison of both algorithms when the time is 10800 s and the population size is 100 for the different measures computed. If we focus

Table 7
Computational results for Algorithm 2 in datasets with $n \leq 500$.

	TIME	SIZE	$\overline{M}$			$\overline{M}^*$		
			AUC	FSC	KAP	AUC	FSC	KAP
Arrhythmia	1200	50	**0.750**	**0.811**	**0.508**	**0.753**	**0.814**	**0.514**
	1200	100	**0.755**	**0.824**	**0.528**	0.760	**0.827**	0.537
	3600	50	**0.751**	**0.813**	**0.513**	**0.756**	**0.817**	**0.524**
	3600	100	**0.754**	**0.824**	**0.527**	**0.764**	**0.831**	**0.544**
	10800	50	**0.751**	**0.815**	**0.514**	**0.753**	**0.816**	**0.517**
	10800	100	**0.752**	**0.825**	**0.524**	**0.752**	**0.825**	**0.524**
GC	1200	50	**0.706**	**0.593**	0.351	**0.711**	**0.597**	0.373
	1200	100	**0.714**	**0.602**	0.401	**0.717**	**0.605**	0.403
	3600	50	**0.705**	**0.591**	0.360	**0.712**	**0.599**	0.375
	3600	100	**0.715**	**0.602**	**0.414**	**0.717**	**0.603**	0.417
	10800	50	**0.704**	**0.590**	0.346	**0.706**	**0.592**	0.367
	10800	100	**0.720**	**0.610**	**0.427**	**0.721**	**0.611**	**0.431**
Housing	1200	50	**0.864**	**0.861**	**0.725**	**0.864**	**0.861**	**0.725**
	1200	100	**0.863**	**0.861**	**0.725**	**0.866**	**0.863**	**0.730**
	3600	50	**0.872**	**0.873**	**0.743**	**0.875**	**0.876**	**0.749**
	3600	100	**0.869**	**0.868**	**0.735**	**0.869**	**0.869**	**0.737**
	10800	50	**0.871**	**0.871**	**0.742**	**0.873**	**0.872**	**0.744**
	10800	100	**0.869**	**0.868**	**0.737**	**0.874**	**0.872**	**0.746**
IONO	1200	50	**0.858**	**0.915**	**0.742**	0.858	0.916	0.744
	1200	100	**0.867**	0.918	0.750	0.873	0.921	0.760
	3600	50	**0.863**	**0.919**	**0.753**	**0.866**	**0.923**	**0.762**
	3600	100	**0.862**	**0.917**	**0.749**	**0.866**	**0.921**	**0.759**
	10800	50	**0.865**	**0.917**	**0.753**	0.866	0.917	0.754
	10800	100	**0.866**	**0.918**	**0.752**	**0.867**	0.919	0.754
Madelon	1200	50	**0.572**	**0.680**	**0.144**	**0.587**	**0.683**	**0.174**
	1200	100	**0.617**	**0.696**	**0.235**	**0.620**	**0.696**	**0.241**
	3600	50	**0.586**	**0.680**	**0.171**	**0.594**	**0.683**	**0.188**
	3600	100	**0.611**	**0.690**	**0.222**	**0.625**	**0.693**	**0.249**
	10800	50	**0.603**	**0.691**	**0.205**	**0.609**	**0.693**	**0.218**
	10800	100	**0.583**	**0.687**	**0.165**	**0.598**	**0.690**	**0.195**
WBC	1200	50	0.971	0.964	0.943	0.973	0.967	0.947
	1200	100	0.965	0.954	0.925	0.965	0.954	0.925
	3600	50	**0.975**	**0.969**	**0.950**	**0.975**	**0.969**	**0.950**
	3600	100	**0.970**	**0.963**	**0.941**	0.972	0.963	0.941
	10800	50	**0.973**	**0.966**	**0.945**	**0.974**	**0.966**	**0.945**
	10800	100	0.966	0.957	0.931	0.966	0.958	0.932
Average			**0.793**	**0.808**	**0.586**	**0.797**	**0.810**	**0.594**

on the AUC average, Algorithm 2 outperforms Algorithm 1 in 11 of the 13 datasets and in the other two, Algorithm 1 is the one which reports better results, these datasets being WBC (0.976>0.966) which is in the group having a small number of features, and MFeat (0.997>0.996) grouped with the datasets having more than 500 features. The values in the rest of the rows can be interpreted in the same way. If we analyze the performance of both algorithms paying attention to the metric AUC (average or maximum), Algorithm 2 outperforms Algorithm 1 in 112 of the 156 comparisons performed. If we focus now on the FSC metric (average or maximum), Algorithm 2 clearly outperforms Algorithm 1 in all the scenarios considered, regardless of the computation time or population size. Specifically, Algorithm 2 gives better results for the FSC metric in 122 of the 156 cases. Similar results are shown if we consider the KAP metric, where Algorithm 2 performs better in 101 of the cases.

The results presented in this section clearly indicate the better performance of the classifiers derived from the alternative model for the soft margin SVM, where the classical objectives have been replaced by the new objectives, minimizing the number of misclassified vectors in each class. Thus, when the goodness of the classification is going to be measured in terms of AUC, FSC or KAP, we would recommend solving the new model proposed as opposed to the classical one.

6. Conclusions and future research

The soft margin SVM has been widely studied in the literature in recent years but most of the works have managed it from a single-objective perspective through a scalarization of the two objectives considered in the problem. When the problem includes feature selection it becomes hard to solve and given that it is a case of non-convex optimization, the only way to obtain the whole set of non-dominated solutions is to use a multi-objective approach that considers the objectives in a separate way. Having all these solutions is important for the decision maker because it provides a wide set of classifiers from where the most appropriate can be chosen. This selection can be made through several metrics that permit the comparison of different classifiers.

In this paper, we propose an alternative model to the soft margin SVM problem with feature selection. The differences between these models are the objectives to be considered. Both models, given the complexity of the problems, are solved through metaheuristics based on NSGA-II. The comparison of the classifiers provided by these models is made through FSC, AUC and KAP analysis. The computational experiment has been carried out employing a stratified hold-out with repetition with 13 datasets in the literature in different scenarios. The results show that, when one of these measures is considered to

Table 8
Computational results for Algorithm 1 in datasets with $n > 500$.

	TIME	SIZE	$\overline{M}$			$\overline{M}^*$		
			AUC	FSC	KAP	AUC	FSC	KAP
Arcene	1200	50	0.612	0.646	0.215	0.633	0.653	0.257
	1200	100	0.585	0.640	0.167	0.634	0.654	0.272
	3600	50	0.532	0.532	0.062	0.541	0.542	0.079
	3600	100	0.600	0.650	0.195	0.659	0.673	0.309
	10800	50	0.595	0.646	0.182	0.623	0.653	0.236
	10800	100	0.572	0.633	0.140	0.580	0.636	0.157
Bioresponse	1200	50	0.500	0.609	0.001	0.501	0.703	0.002
	1200	100	0.516	0.565	0.031	0.540	0.568	0.080
	3600	50	0.500	0.703	0.000	0.500	0.703	0.000
	3600	100	0.502	0.562	0.004	0.503	0.563	0.006
	10800	50	0.519	0.518	0.038	0.550	0.571	0.099
	10800	100	0.501	0.703	0.001	0.501	0.703	0.002
Duke	1200	50	**0.724**	**0.739**	**0.445**	**0.725**	**0.739**	**0.446**
	1200	100	0.736	0.751	0.470	0.779	0.785	0.553
	3600	50	**0.730**	**0.734**	**0.452**	**0.745**	**0.744**	**0.479**
	3600	100	0.740	0.768	0.478	0.761	0.795	0.519
	10800	50	**0.691**	0.704	**0.376**	**0.714**	**0.736**	**0.421**
	10800	100	0.768	0.794	0.532	0.805	0.823	0.605
Gina_agnostic	1200	50	**0.636**	0.630	**0.274**	**0.657**	0.641	**0.315**
	1200	100	**0.645**	0.567	**0.293**	**0.703**	0.649	**0.408**
	3600	50	0.617	0.656	0.235	0.668	0.676	0.337
	3600	100	**0.648**	0.660	**0.298**	**0.695**	0.675	**0.393**
	10800	50	**0.626**	0.562	**0.254**	**0.660**	0.642	**0.323**
	10800	100	0.624	0.638	0.250	**0.695**	0.658	**0.392**
Gisette	1200	50	**0.720**	0.747	**0.441**	**0.737**	**0.778**	**0.475**
	1200	100	0.750	0.785	0.500	**0.784**	**0.806**	**0.567**
	3600	50	**0.767**	**0.799**	**0.534**	**0.857**	**0.860**	**0.714**
	3600	100	0.672	0.712	0.345	0.695	0.726	0.389
	10800	50	**0.762**	**0.797**	**0.524**	**0.842**	**0.849**	**0.685**
	10800	100	0.666	0.688	0.333	0.738	0.734	0.475
Isolet	1200	50	0.641	0.746	0.282	0.646	0.751	0.293
	1200	100	0.674	0.637	0.349	0.715	0.660	0.430
	3600	50	0.593	0.722	0.186	0.593	0.722	0.186
	3600	100	0.718	0.797	0.435	0.843	0.868	0.687
	10800	50	0.630	0.731	0.260	0.648	0.743	0.297
	10800	100	0.684	0.759	0.368	0.717	0.780	0.434
MFeat	1200	50	**0.997**	**0.996**	**0.996**	**0.997**	**0.997**	**0.997**
	1200	100	**0.997**	**0.997**	**0.997**	**0.997**	**0.997**	**0.997**
	3600	50	**0.997**	**0.997**	**0.997**	**0.997**	**0.997**	**0.997**
	3600	100	**0.998**	**0.997**	**0.997**	**0.998**	**0.997**	**0.997**
	10800	50	**0.997**	**0.995**	**0.995**	0.998	0.995	0.995
	10800	100	**0.997**	**0.997**	**0.997**	**0.997**	**0.997**	**0.997**
Average			0.690	0.726	0.379	0.718	0.749	0.436

determine the goodness of the classifier, the model proposed gives, in general, better results. Therefore, we can conclude that if the goal of the decision maker is to find the best classifier for a dataset with regard to one of these metrics, the new metaheuristic proposed should be employed and then the best of the Points on the Pareto front according to the desired metric could be selected. The reason for the improvement using the new metaheuristic lies only in the change of the objectives considered in the multi-objective search space. This is an important conclusion because it would allow other techniques to be easily adapted by employing classical objective functions to operate along with the new ones. Linked to this idea, a future research line could consist on designing and developing new metaheuristics, based on different paradigms (SPEA2, PAES, etc.) that implement these objectives or even new objectives based on the metrics. It would also be interesting to study some other variants of the problem and analyze the impact of the objectives considered.

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

The datasets can be downloaded from the repositories cited in the work.

Acknowledgments

The authors thank the grants PID2021-122344NB-I00 and PID2019-105952GB-I00 funded by Ministerio de Ciencia e Innovación/Agencia Estatal de Investigación /10.13039/501100011033. This work was partially also supported by the Generalitat Valenciana under grants PROMETEO/2021/063 and ACIF/2020/155.

Table 9
Computational results for Algorithm 2 in datasets with $n > 500$.

	TIME	SIZE	$\overline{M}$			$\overline{M}^*$		
			AUC	FSC	KAP	AUC	FSC	KAP
Arcene	1200	50	**0.717**	**0.676**	**0.435**	**0.742**	**0.720**	**0.478**
	1200	100	**0.726**	**0.706**	**0.448**	**0.754**	**0.734**	**0.502**
	3600	50	**0.724**	**0.683**	**0.455**	**0.751**	**0.723**	**0.505**
	3600	100	**0.736**	**0.715**	**0.471**	**0.762**	**0.738**	**0.522**
	10800	50	**0.731**	**0.698**	**0.458**	**0.778**	**0.761**	**0.546**
	10800	100	**0.746**	**0.720**	**0.492**	**0.777**	**0.749**	**0.555**
Bioresponse	1200	50	**0.586**	**0.720**	**0.175**	**0.609**	**0.725**	**0.221**
	1200	100	**0.693**	**0.749**	**0.386**	**0.721**	**0.758**	**0.442**
	3600	50	**0.571**	**0.719**	**0.144**	**0.589**	**0.724**	**0.179**
	3600	100	**0.660**	**0.738**	**0.323**	**0.670**	**0.744**	**0.342**
	10800	50	**0.598**	**0.725**	**0.195**	**0.631**	**0.736**	**0.259**
	10800	100	**0.677**	**0.748**	**0.357**	**0.693**	**0.757**	**0.388**
Duke	1200	50	0.617	0.623	0.236	0.627	0.627	0.255
	1200	100	**0.793**	**0.794**	**0.584**	**0.843**	**0.838**	**0.682**
	3600	50	0.555	0.558	0.112	0.629	0.623	0.257
	3600	100	**0.774**	**0.792**	**0.548**	**0.839**	**0.854**	**0.678**
	10800	50	0.681	**0.712**	0.362	0.705	0.731	0.410
	10800	100	**0.790**	**0.817**	**0.581**	**0.848**	**0.874**	**0.702**
Gina_agnostic	1200	50	0.611	**0.678**	0.221	0.619	**0.679**	0.238
	1200	100	0.642	**0.683**	0.285	0.651	**0.683**	0.303
	3600	50	**0.650**	**0.683**	**0.300**	**0.688**	**0.694**	**0.378**
	3600	100	0.641	**0.684**	0.283	0.664	**0.688**	0.330
	10800	50	0.597	**0.687**	0.195	0.613	**0.687**	0.226
	10800	100	**0.664**	**0.710**	**0.329**	0.682	**0.719**	0.363
Gisette	1200	50	0.684	**0.758**	0.368	0.699	0.769	0.399
	1200	100	**0.762**	**0.789**	**0.524**	0.778	0.801	0.556
	3600	50	0.711	0.766	0.423	0.773	0.797	0.545
	3600	100	**0.746**	**0.783**	**0.492**	**0.769**	**0.801**	**0.537**
	10800	50	0.741	0.788	0.483	0.778	0.815	0.556
	10800	100	**0.736**	**0.775**	**0.473**	**0.757**	**0.788**	**0.515**
Isolet	1200	50	**0.980**	**0.980**	**0.960**	**0.982**	**0.982**	**0.964**
	1200	100	**0.988**	**0.988**	**0.975**	**0.989**	**0.989**	**0.978**
	3600	50	**0.983**	**0.983**	**0.966**	**0.983**	**0.983**	**0.966**
	3600	100	**0.977**	**0.977**	**0.955**	**0.981**	**0.981**	**0.962**
	10800	50	**0.984**	**0.983**	**0.967**	**0.986**	**0.986**	**0.972**
	10800	100	**0.985**	**0.985**	**0.970**	**0.986**	**0.986**	**0.972**
MFeat	1200	50	0.996	0.988	0.986	0.996	0.990	0.988
	1200	100	0.993	0.978	0.975	0.993	0.978	0.975
	3600	50	0.995	0.983	0.981	0.997	0.985	0.983
	3600	100	0.995	0.988	0.987	0.996	0.991	0.990
	10800	50	0.996	0.988	0.986	**0.998**	**0.995**	**0.995**
	10800	100	0.996	0.989	0.988	0.996	0.991	0.990
Average			**0.772**	**0.797**	**0.544**	**0.793**	**0.814**	**0.586**

Table 10
Success frequency table.

TIME	SIZE	$\overline{M}$						$\overline{M}^*$					
		AUC		FSC		KAP		AUC		FSC		KAP	
		1	2	1	2	1	2	1	2	1	2	1	2
1200	50	5	8	3	10	6	7	6	7	5	8	7	6
1200	100	3	10	3	10	5	8	6	7	4	9	7	6
3600	50	3	10	3	10	4	9	3	10	3	10	4	9
3600	100	2	11	1	12	2	11	3	10	2	11	4	9
10800	50	4	9	2	11	5	8	4	8	3	10	5	8
10800	100	2	11	2	11	2	11	3	10	3	10	4	9
Total		19	59	14	64	24	54	25	53	20	58	31	47

References

Abdunabi, T., Basir, O., 2014. Predicting a biological response of molecules from their chemical properties using diverse and optimized ensembles of stochastic gradient boosting machine. In: 2014 IEEE International Conference on Big Data. Institute of Electrical and Electronics Engineers Inc., pp. 10–17.

Aladeemy, M., Tutun, S., Khasawneh, M.T., 2017. A new hybrid approach for feature selection and support vector machine model selection based on self-adaptive cohort intelligence. Expert Syst. Appl. 88, 118–131.

Alcaraz, J., Labbé, M., Landete, M., 2022. Support vector machine with feature selection: A multiobjective approach. Expert Syst. Appl. 204, 117485.

Aranha, C., Camacho Villalón, C.L., Campelo, F., Dorigo, M., Ruiz, R., Sevaux, M., Sörensen, K., Stützle, T., 2022. Metaphor-based metaheuristics, a call for action: The elephant in the room. Swarm Intell. 16 (1), 1–6.

Aytug, H., 2015. Feature selection for support vector machines using generalized benders decomposition. European J. Oper. Res. 244 (1), 210–218.

Benítez-Peña, S., Blanquero, R., Carrizosa, E., Ramírez-Cobo, P., 2019. Cost-sensitive feature selection for support vector machines. Comput. Oper. Res. 106, 169–178.

Bouraoui, A., Jamoussi, S., BenAyed, Y., 2018. A multi-objective genetic algorithm for simultaneous model and feature selection for support vector machines. Artif. Intell. Rev. 50 (2), 261–281.

Bradley, P.S., Mangasarian, O.L., 1998. Feature selection via concave minimization and support vector machines. In: ICML '98: Proceedings of the Fifteenth International

Conference on Machine Learning, Vol. 98. Morgan Kaufmann Publishers Inc., pp. 82–90.
Burges, C.J., 1998. A tutorial on support vector machines for pattern recognition. Data Min. Knowl. Discov. 2 (2), 121–167.
Candelieri, A., Giordani, I., Archetti, F., Barkalov, K., Meyerov, I., Polovinkin, A., Sysoyev, A., Zolotykh, N., 2019. Tuning hyperparameters of a SVM-based water demand forecasting system through parallel global optimization. Comput. Oper. Res. 106, 202–209.
Carrizosa, E., Martin-Barragan, B., Morales, D.R., 2014. A nested heuristic for parameter tuning in support vector machines. Comput. Oper. Res. 43, 328–334.
Chang, C.-C., Lin, C.-J., 2011. LIBSVM: A library for support vector machines. ACM Trans. Intell. Syst. Technol. 2 (3), 1–27.
Cohen, J., 1960. A coefficient of agreement for nominal scales. Educ. Psychol. Meas. 20 (1), 37–46.
Cortes, C., Vapnik, V., 1995. Support-vector networks. Mach. Learn. 20 (3), 273–297.
Deb, K., Pratap, A., Agarwal, S., Meyarivan, T., 2002. A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Trans. Evol. Comput. 6 (2), 182–197.
Dua, D., Graff, C., 2017. UCI Machine Learning Repository. University of California, Irvine, School of Information and Computer Sciences, URL http://archive.ics.uci.edu/ml.
Dudzik, W., Nalepa, J., Kawulok, M., 2021. Evolving data-adaptive support vector machines for binary classification. Knowl.-Based Syst. 227, 107221.
Ezugwu, A.E., Shukla, A.K., Nath, R., Akinyelu, A.A., Agushaka, J.O., Chiroma, H., Muhuri, P.K., 2021. Metaheuristics: A comprehensive overview and classification along with bibliometric analysis. Artif. Intell. Rev. 54 (6), 4237–4316.
Faris, H., Hassonah, M.A., Al-Zoubi, A., Mirjalili, S., Aljarah, I., 2018. A multi-verse optimizer approach for feature selection and optimizing SVM parameters based on a robust system architecture. Neural Comput. Appl. 30 (8), 2355–2369.
Fawcett, T., 2006. An introduction to ROC analysis. Pattern Recognit. Lett. 27 (8), 861–874.
Florian, E., Sgarbossa, F., Zennaro, I., 2021. Machine learning-based predictive maintenance: A cost-oriented model for implementation. Int. J. Prod. Econ. 236, 108114.
García-Pedrajas, N., de Haro-García, A., Pérez-Rodríguez, J., 2014. A scalable memetic algorithm for simultaneous instance and feature selection. Evolut. Comput. 22 (1), 1–45.
Gaudioso, M., Gorgone, E., Labbé, M., Rodríguez-Chía, A.M., 2017. Lagrangian relaxation for SVM feature selection. Comput. Oper. Res. 87, 137–145.
Gu, Q., Zhu, L., Cai, Z., 2009. Evaluation measures of the classification performance of imbalanced data sets. In: International Symposium on Intelligence Computation and Applications. Springer, pp. 461–471.
Guyon, I., Saffari, A., Dror, G., Cawley, G., 2007. Agnostic learning vs. prior knowledge challenge. In: 2007 International Joint Conference on Neural Networks. Institute of Electrical and Electronics Engineers Inc., pp. 829–834.
Halder, A., Chatterjee, S., Dey, D., 2022. Adaptive morphology aided 2-pathway convolutional neural network for lung nodule classification. Biomed. Signal Process. Control 72, 103347.
Huang, C.-L., Wang, C.-J., 2006. A GA-based feature selection and parameters optimization for support vector machines. Expert Syst. Appl. 31 (2), 231–240.
Ierimonti, L., Venanzi, I., Ubertini, F., 2021. ROC analysis-based optimal design of a spatio-temporal online seismic monitoring system for precast industrial buildings. Bull. Earthq. Eng. 19 (3), 1441–1466.
Ismael, A.M., Şengür, A., 2021. Deep learning approaches for COVID-19 detection based on chest X-ray images. Expert Syst. Appl. 164, 114054.
Kim, M., Hiroyasu, T., Miki, M., Watanabe, S., 2004. SPEA2+: Improving the performance of the strength Pareto evolutionary algorithm 2. In: International Conference on Parallel Problem Solving from Nature. Springer, pp. 742–751.
Knowles, J.D., Corne, D.W., 2000. Approximating the nondominated front using the Pareto archived evolution strategy. Evolut. Comput. 8 (2), 149–172.
Kwiatkowski, J., Sotor, J., 2022. Laser wavelength shift and dual-wavelength generation in continuous-wave operation of Ho:YAG laser pumped by thulium-doped fiber laser. Opt. Laser Technol. 146, 107544.
Labbé, M., Martínez-Merino, L.I., Rodríguez-Chía, A.M., 2019. Mixed integer linear programming for feature selection in support vector machine. Discrete Appl. Math. 261, 276–304.
Loh, L.X., Lee, H.H., Stead, S., Ng, D.H., 2022. Manuka honey authentication by a compact atmospheric solids analysis probe mass spectrometer. J. Food Comp. Anal. 105, 104254.
Maldonado, S., Pérez, J., Weber, R., Labbé, M., 2014. Feature selection for support vector machines via mixed integer linear programming. Inform. Sci. 279, 163–175.
Ponmalar, A., Dhanakoti, V., 2022. An intrusion detection approach using ensemble support vector machine based chaos game optimization algorithm in big data platform. Appl. Soft Comput. 116, 108295.
Raman, M.G., Somu, N., Kirthivasan, K., Liscano, R., Sriram, V.S., 2017. An efficient intrusion detection system based on hypergraph-genetic algorithm for parameter optimization and feature selection in support vector machine. Knowl.-Based Syst. 134, 1–12.
Roshani, M., Phan, G.T., Ali, P.J.M., Roshani, G.H., Hanus, R., Duong, T., Corniani, E., Nazemi, E., Kalmoun, E.M., 2021. Evaluation of flow pattern recognition and void fraction measurement in two phase flow independent of oil pipeline's scale layer thickness. Alex. Eng. J. 60 (1), 1955–1966.
Sörensen, K., 2015. Metaheuristics—the metaphor exposed. Int. Trans. Oper. Res. 22 (1), 3–18.
Srinivas, N., Deb, K., 1995. Multiobjective function optimization using nondominated sorting genetic algorithms. Evolut. Comput. 2, 221–248.
Vapnik, V., 2013. the Nature of Statistical Learning Theory. Springer Science & Business Media.
Vijayan, N.M., Johnson, M.S., Jacob, J., 2021. Convex-optimization-based constrained control strategy for 3-dof tandem helicopter using feedback linearization. J. Optim. Theory Appl. 191 (2), 736–755.
Xue, Y., Tang, Y., Xu, X., Liang, J., Neri, F., 2021. Multi-objective feature selection with missing data in classification. IEEE Trans. Emerg. Top. Comput. Intell. 6 (2), 355–364.
Yang, Y., Hu, X., Jiang, H., 2022. Group penalized logistic regressions predict up and down trends for stock prices. North Am. J. Econ. Finance 59, 101564.
Zhao, M., Fu, C., Ji, L., Tang, K., Zhou, M., 2011. Feature selection and parameter optimization for support vector machines: A new approach based on genetic algorithm with feature chromosomes. Expert Syst. Appl. 38 (5), 5197–5204.