Multiclassifiers and feature selection methods for success/failure prediction of noninvasive mechanical ventilation in intensive care units

F. Martín[1], J. González[1], F. Sánchez [2] and M. N. Moreno[3*]

[1]Intensive Care Unit. University Hospital of Salamanca, Salamanca, Spain.

[2] School of Nursing and Physiotherapy. University of Salamanca. Prehospital Emergency Services. Salamanca, Spain.

[3]Department of Computing and Automation. University of Salamanca, Salamanca, Spain

[*]Corresponding author: Department of Computing and Automation, Plaza de los Caídos s/n, 37008 Salamanca. Tel. +34294400 ext. 1513. E-mail: mmg@usal.es

Supplementary material

APPENDIX I. Data description

The variables gathered are the following:

- Demographic variables: age, gender.
- Personal antecedents
- RF etiology
- diagnosis
- Admission date (in the hospital and in the ICU), discharge data (Hospital and ICU), NIMV start and finalization date, NIMV hours, NIMV success/failure, cause of the failure, kind of the NIMV (BPAP and/or CPAP), orotracheal intubation (OTI), tracheotomy, existence of inotropic or vasoactive treatment at the beginning of NIMV, need for sedation to tolerate NIMV, establishment of limitation of life support measures, Acute Physiology and Chronic Health Evaluation II score (APACHE II), appearance of major complications, death (Hospital and ICU), causes of death and radiological findings in chest X-ray.
- Physiological variables: systolic blood pressure (SBP), diastolic blood pressure (DBP), heart rate (HR), respiratory rate (RR), Glasgow Coma Scale (GCS) and temperature (T) before the beginning of the NIMV, after 2 hours, 12 hours, 24 hours and at the time of ICU discharge.
- Gasometric variables: pH, $PaCO_2$, PaO_2/FiO_2 ratio, bicarbonate and lactate before the beginning of the NIMV, after 2 hours, 12 hours, 24 hours and at the ICU discharge.
- Biochemical variables: urea, creatinine, albumin, sodium, potassium and total bilirubin, before the beginning of NIMV, and urea, creatinine and albumen after 2 hours, 12 hours, 24 hours and at the time of ICU discharge.
- Hematologic variables: hematocrit and leukocytes before the beginning of NIMV, after 2 hours, 12 hours, 24 hours and at the time of ICU discharge.
- Ventilatory variables: inspiratory positive airway pressure (IPAP) and expiratory positive airway pressure (EPAP) before the beginning of NIMV, after 2 hours, 12 hours, 24 hours and at the time of ICU discharge.
- Fluid balance after 12 hours and 24 hours from the start of NIMV
- With the variables for the beginning of NIMV, a modified Simplified Acute Physiology Score II (SAPS II) was calculated, in which the urinary output was included since it was obtained at a certain point and not with the information of 24 hours after the NIMV start, not even the PaO_2/FiO_2 ratio, since at this moment the patient was not under NIMV. In addition a modified SAPS II was calculated with the same information but adding the PaO_2/FiO_2 ratio value corresponding to the 2 hours after the NIMV start.

The physiological, gasometric, biochemical, hematologic and ventilatory variables were only gathered during the treatment with NIMV.

The patients were divided into five groups according to the RF etiology: Re-worsening of chronic obstructive pulmonary disease (COPD); acute hypercapnic RF or re-worsening of chronic hypercapnic RF without COPD; acute hypoxemic RF (PaO_2/FiO_2 ratio <300 and $PaCO_2$ not raised); acute pulmonary edema in patient diagnosed of acute or chronic heart failure; postextubation RF (defined as the one that appears in 48 hours following extubation).

The chest X-ray at the beginning of the NIMV was classified as: clean parenchyma; injection/ consolidation / unilateral spillage; unilateral consolidation/pulmonary infiltrates/pleural effusion.

Before the data mining study some variables without significance for the study, such as the dates, were manually discarded, as well as those directly related with the class that cannot be used for prediction of success/failure in new patients since they are not early known, such as all the discharge data, the kind of failure, tracheotomy, death (Hospital and ICU) or causes of death.

Statistical values of some of these variables are shown below. Table 1 contains the minimum, maximum, mean and standard deviation for continuous variables. Figure 1 shows the distribution of values of some important variables in relation to the two classes: NIMV failure and NIMV success (YES and NO labels respectively). To display this value distribution of continuous variables they were discretized into intervals, however, continuous values are used as input to the algorithms.

Table 1. Distribution of values of continuous variables. PaO_2 and $PaCO_2$ are measured in mmHg and FiO2 in percentage. RR is respiratory rate in breaths a minute, fluid balance is given in ml., heart rate in bpm (beats per minute), albumin in g/dl and bilirubin in mg/dl

Variable	Minimum	Maximum	Mean	Standard deviation
Age	18	88	66.69	13.39
NIMV hours	1	232	18.04	27.09
APACHE II at the start	5	47	20.83	7.45
pH at the start	6.8	7.58	7.32	0.12
$PaCO_2$ at the start	6	148	53.46	26.33
PaO_2/FiO_2 ratio	27	420	133.07	61.04
RR	7	60	29.86	8.28
GCS at the start	6	15	14.04	1.93
Hematocrite at the start	11.1	57.5	34.04	8.03
Leukocytes at the start	10	49000	13038.47	7689.71
Albumin at the start	1	4.7	2.76	0.79
$PaCO_2$ after 2 hours	9	115	48.46	19.64
PaO_2/FiO_2 ratio after 2 hours	50	452	172.8	75.78
Heart rate after 2 hours	60	160	96.62	19.17
Respiratory frequency after 2 hours	11	52	25.92	7.17
$PaCO_2$ after 12 hours	22	122	49.99	20.09
RR after 12 hours	11	49	24.52	6.63
GCS after 12 housr	6	15	14.61	1.12
Fluid balance after 12 hours	-4000	7000	744.88	1469.399
$PaCO_2$ after 24 hours	20	139	49.33	21.93
PaO2/FiO2 ratio after 24 hours	66	306	169.222	58.01
RR after 24 hours	13	41	24.29	5.85
Fluid balance after 24 hours	-6000	10000	1422.65	2309.13
Bilirubin at the start	0.1	22.4	1.1	1.78
SAPS II	6	61	28.04	9.47
Days of hospital stay before NIMV administration	0	88	6.78	10.13
Change of RR 2 hours after admission	-40	19	-3.76	7.2

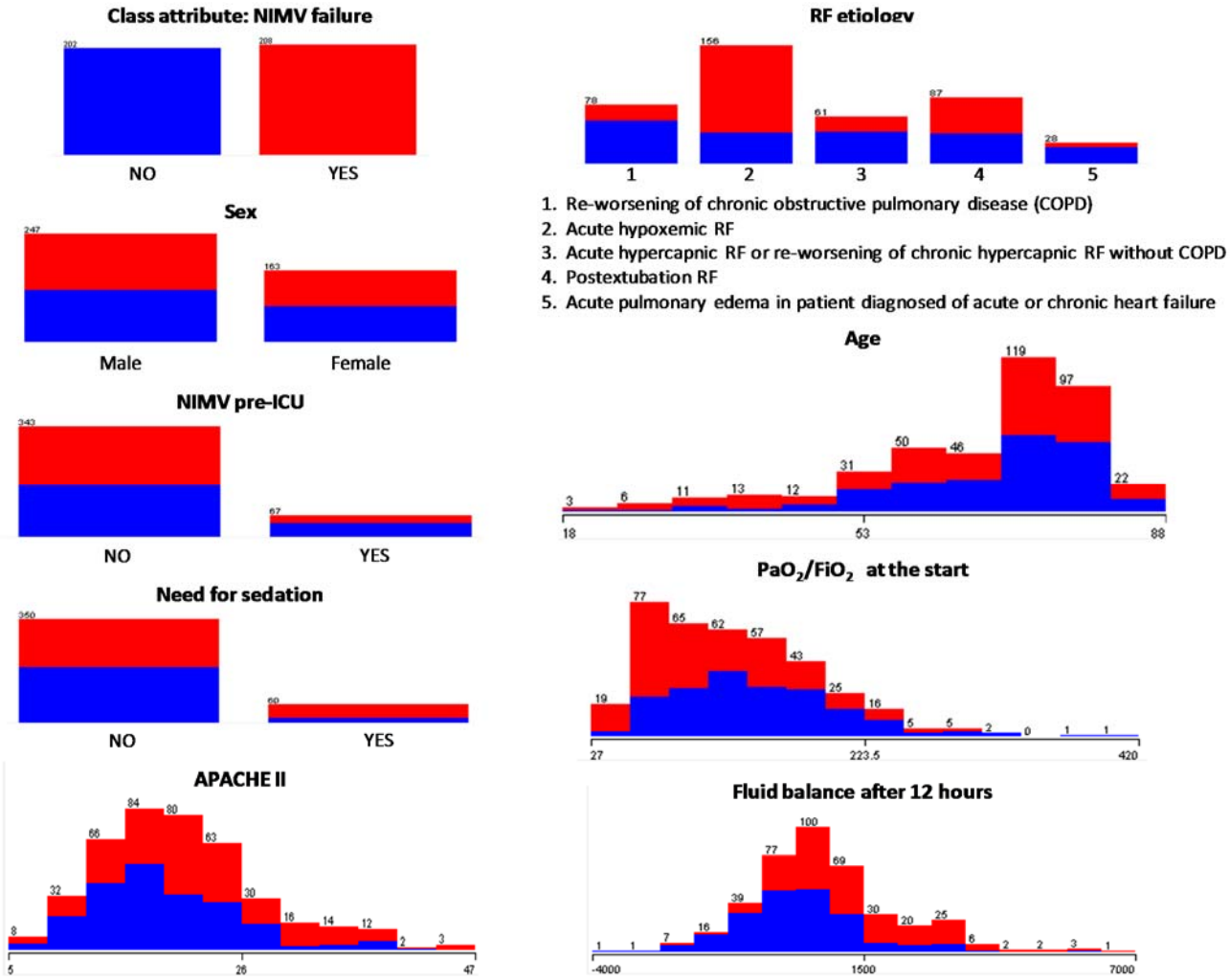

Figure 1. Distribution of values of variables in relation to the classes

APPENDIX II. Feature selection

Table 1. Attributes selected by the CFS method

Rank	Attribute
1	NIMV hours
2	APACHE II at the start
3	$PaCO_2$ at the start
4	PaO_2/FiO_2 ratio at the start
5	Bicarbonate at the start
6	Hematocrit at the start
7	Leukocytes at the start
8	PaO_2/FiO_2 ratio after 2 hours
9	RR after 2 hours
10	Fluid balance after 12 hours
11	Fluid balance after 24 hours
12	RF etiology
13	Need for sedation to tolerate the NIMV
14	Days of hospital stay before NIMV administration
15	Change of RR (respiration rate) 2 hours after admission

Table 2. Attributes selected by the Information Gain method

Rank	Attribute	Information Gain
1	Fluid balance after 24 hours	0.1662
2	Kind of NIMV	0.1283
3	Respiratory frequency after 2 hours	0.1059
4	Fluid balance after 12 hours	0.0846
5	NIMV hours	0.0753
6	$PaCO_2$ at the start	0.0699
7	Hematocrit at the start	0.0529
8	PaO_2/FiO_2 ratio after 2 hours	0.0491
9	Bilirubin at the start	0.0489
10	PaO_2/FiO_2 ratio at the start	0.0489
11	Albumin at the start	0.0407
12	Leukocytes at the start	0.0391
13	Bicarbonate at the start	0.0389
14	Immunosuppression	0.0359
15	Chest X-ray	0.0345
16	APACHE II	0.0318
17	Need for sedation to tolerate NIMV	0.0304
18	Days of hospital stay before NIMV administration	0.0304
19	Heart rate after 2 hours	0.0280
20	Bicarbonate after 2 hours	0.0279
21	Respiratory rate (RR) at the start	0.0278
22	pH at the start	0.0253
23	$PaCO_2$ after 2 hours	0.0238
24	Change of RR 2 hours after admission	0.0232
25	GCS at the start	0.0196
26	RR after 2 hours	0.0147
27	Bicarbonate after 12 hours	0.0128
28	Use of NIMV before ICU	0.0127
29	Use of BPAP before ICU	0.0122
30	$PaCO_2$ after 12 hours	0.0120
31	Domiciliary Oxygen therapy	0.0110

APPENDIX III. Data Mining Methods

III.1. Feature selection techniques

Most of the machine learning algorithms usually do not present a good behavior when they use high dimensional datasets as input. In these situations a suitable way of increasing accuracy is to preprocess datasets by means of feature selection methods in order to obtain, from the original dataset, subsets of the features that most contribute to increase the accuracy. This is an efficient way to reduce the dimensionality and to increase the accuracy of the induced classifiers.

CFS (Correlation-based Feature Subset Selection) and IG (Information Gain) are two popular and reliable methods that are used in this work.

CFS evaluates the significance of a subset of features (attributes) taking into account the individual predictive ability of each feature and the degree of redundancy between them. This method selects the subsets of attributes that are highly correlated with the class while having low inter-correlation between them. Feature subsets are ranked according to a correlation based heuristic evaluation function.

The acceptance of a feature will depend on the extent to which it predicts classes in areas of the instance space not already predicted by other features. To do that, the following feature subset evaluation function is used:

$$M_S = \frac{k\overline{r_{cf}}}{\sqrt{k + k(k-1)\overline{r_{ff}}}} \quad \text{(Eq. 1)}$$

where M_S is the heuristic "merit" of a feature subset S containing k features, $\overline{r_{cf}}$ is the mean feature-class correlation ($f \in S$), and $\overline{r_{ff}}$ is the average feature-feature inter-correlation. The numerator of equation 1 is an indicator of the predictive power of the features with respect to the class and the denominator indicates the redundancy degree among the features.

The information gain method evaluates the importance of an attribute by means of obtaining the information gain (IG) with respect to the class according to the following equation:

$$IG(Class, Attribute) = H(Class) - H(Class \mid Attribute) \quad \text{(Eq. 2)}$$

Where H(Class) is the Shannon entropy for the class and H(Class | Attribute) is the class entropy given the attribute. In the method the gain values for each attribute are evaluated and then they are ordered according to

these values. Those providing more information gain are the most relevant and therefore the most influential in the classification.

III.2. Classification methods

III.2.1. Decision trees

A decision tree consists of a set of conditions that are organized in a hierarchical structure, so that the final decision can be determined following the true conditions from the root to the leafs of the tree.

The induction of the tree is carried out by means of a process in which the examples are separated depending on the evaluation of certain conditions related to the values of the attributes. The algorithm starts with the identification of the most significant attribute, that is, the most influential in the classification. his attribute is placed in the root of the tree. After that, all the examples of the training set are checked against a condition regarding the values of this attribute and those satisfying the condition are placed in the left branch of the tree while the remaining ones are placed in the right branch, in the case of binary trees. In the next step the following significant attribute is selected and the process is repeated until all the examples are classified or a stopping criterion is fulfilled.

The mathematical model used to select the attributes influencing the classification as well as the attribute values involved in the conditions is based on the entropy provided by the attribute.

In this study, two tree induction algorithms have been used, J48 and REPTree. J48 is an advanced version of C4.5, one of the most known and used algorithms. J48 is a information gain based method with pruning procedures that use rules. REPTree is a fast decision tree learning algorithm, also based on information gain, which uses reduced-error pruning with back fitting. Moreover, REPTree can be used for inducing regression trees but in this option, variance is considered for selecting attributes and generating the partitions instead of information gain.

III.2.2. Bayesian networks

Bayesian networks (BN) are probabilistic graphical models where nodes represent random variables and edges represent conditional dependencies between the variables. Two nodes representing conditionally independent variables are not connected each other. Each node has a conditional probability distribution given its parents:

$$P(X_i \mid Parents\ (X_i)) \quad \text{(Eq. 3)}$$

The learning process for a dataset D lies in finding, among all possible graphs, the graph G that better represents the set of dependencies/independencies between data. The problem is NP-hard, so that is not feasible an exact solution and it is necessary to resort to heuristic search methods. They consist of establishing a quality metric, which represents the adaptation of a Bayesian network to a specific dataset, and finding a solution that maximizes this metric by means of an optimization procedure. Some search algorithms are TAN, BAN, K2, etc. TAN (Augmented Naïve -Bayes) and BAN (Bayesian Networks Augmented Naïve-Bayes) are based in the simpliest BN, Naïve Bayes, but they involves sophisticated graphical model to deal with the no realistic assumption of attribute independence. K2 is a function based on uniform prior scoring for learning and evaluating Bayesian networks. K2 measures the joint probability of a BN G given a dataset D using the following formula:

$$P(G,\mathbf{D}) = P(G)\prod_{i=1}^{n}\prod_{j=1}^{q_i}\frac{(r_i-1)!}{(N_{ij}+r_i-1)!}\prod_{k=1}^{r_i}N_{ijk}!, \quad \text{(Eq. 4)}$$

Where r_i is the number of possible values of the variable X_i , q_i is the number of possible configurations for the variables in Parents (X_i), N_{ijk} is the number of cases in D in which variable x_i has its kth value and Parents (X_i), is instantiated to its jth value, and $N_{ij} = \sum_{k=1}^{r_i} N_{ijk}$.

III.2.3. k- Nearest Neighbors

The k-Nearest Neighbors algorithm (k-NN) can be used for solving both classification and regression problems. The classification of an item is the result of the majority vote of its neighbors, that is, the class most frequent among its k nearest neighbors, being k a positive integer preferably odd to avoid ties.

The search of the neighbors (the k closest training points) is carried out according to some distance metric such as euclidean, manhattan, etc.

III.2.4. Multiclassifiers

Multiclassifiers are methods that combine several individual classifiers induced with different basic methods or obtained from different training datasets with the purpose of improving the precision of the predictions. Another additional advantage of these techniques is the reduction of the overfitting problem, which takes place when the learning process finds a regularity in the data that is distinctive of the training set but it cannot be extended to other datasets.

The methods for building multiclassifiers are divided in two groups. The first methods, such as Bagging and Boosting, induce models that merge classifiers with the same learning algorithm, but introducing modifications in the training data set. The second type of methods, named hybrids, such as Stacking and Cascading create new hybrid learning techniques from different base learning algorithms.

The Bagging (Bootstrap AGGregatING) method allows to induce different classifiers using different training sets, which are created by random selection with replacement (there can be duplicated instances) of a sample of instances with the same size than the original training set. A boostrap sample with *n* size is generated by selecting *n* instances from the training set in a random way, so that so many bootstrap samples with the same size are created as the number of classifiers to be induced and every classifier is trained with a bootstrap sample. The prediction with the highest number of votes is selected between all prediction given by the classifiers.

Boosting is based on assigning different weighs to the examples. In an iterative way, models minimizing errors of the previous ones are built by means of assigning greater weighs to the instances incorrectly classified in the previous iteration. The errors in every iteration are used for updating the weight of the training set examples, so that the weight of the wrong classified examples is increased and the weight of the correctly classified examples is reduced. When they are used for classifying real examples, its behavior in the test phase is also considered.

AdaBoost is the most known variant of boosting. In a cycle, a model is learned across the weighted evidence, the error of the model is estimated and depending on its value the algorithm is stopped or the process continues repeating the cycle. In each iteration the weight of the correctly classified examples is updated, the model is stored and the normalization of the weight of all the examples is carried out.

Random Forest can be considered a multiclassifier similar to Bagging since it involves the induction of an ensemble of tree classifiers, each of which produces its own output. The induction of each tree is produced from a subset of the original data set chosen independently (with replacement) and with the same distribution for all trees in the forest. For classification problems, the most popular class obtained by simple vote is chosen as the final outcome.