Generalized Additive Neural Networks for mortality prediction using Automated and Genetic Algorithms

Carlos Bras-Geraldes*, Ana Papoila*, Patricia Xufre† and Fernanda Diamantino‡
*New University of Lisbon
Faculty of Medical Sciences
University of Lisbon
Center of Statistics and Applications
Lisbon, Portugal
†New University of Lisbon
Nova School of Business and Economics
Lisbon, Portugal
‡University of Lisbon
Faculty of Sciences
University of Lisbon
Center of Statistics and Applications
Lisbon, Portugal

***Abstract*—The prediction of mortality has shown to be a challenge for hospital management. To help in this task, metrics were developed to predict the evolution of the disease severity. One of the most commonly used metric in Intensive Care Units (ICUs) is the SAPS II, based on Generalized Linear Models (GLMs). However, the use of the more flexible Generalized Additive Models (GAMs) provide better results when the association between the outcome and the continuous covariates is nonlinear. Neural networks have also been used for prediction namely those based in the Multi Layer Perceptron (MLP) architecture, as, in theory, they are universal approximators to any continuous function. Some studies have shown that their performances are equivalent to GLMs and, naturally, inspired by GAMs, Generalized Additive Neural Networks (GANNs) were proposed. Because the construction of a GANN is based in a subjective decision making process through the analysis of the residuals plots, studies to automate this process emerged originating new methodologies (AutoGANN). However, these are not free from problems when the number of variables is large. Some improvements were then introduced for model selection, such as, a multistep algorithm that allows more than one modification at the same time in GANNs's architecture. Methods described above have correspondence to evolutionary programming as the search of a better result is performed by small modifications, closely resembling the mutation operator. AutoGANN method and Genetic Algorithm were used in order to find optimal models for predicting mortality at an ICU. These models, as well as a MLP model, were compared regarding their predictive and discriminative abilities.**

I. Introduction

The admission of critically ill patients in Intensive Care Units constitutes a challenge to the hospital management, taking into account the large budgets needed to maintain the quality of response of these units. Daily decisions must be made by focusing the effectiveness of treatments versus its cost. Metrics for predicting the evaluation of the severity of illness have then been developed, such as Mortality Probability Models (MPMs) [1], [2], [3], APACHE III [4], SAPS II (New Simplified Acute Physiology Score) [5], SAPS 3 [6] and APACHE IV [7], have been developed. Most of these scores were obtained by logistic regression models. However because a nonlinear dependence between the binary response variable and continuous covariates may exist, other methods such as artificial neural networks (ANNs)[8] and generalized additive models, must be considered.

The increasing use of ANNs models in the last decade is undoubtedly a reality. In fact, the flexibility that characterizes the ANNs allows them to model not only complex data from real situations, such as pattern recognition and voice, but also simple data such as, for example, those reflecting relationships between several independent variables and a response (dependent variable). Therefore, we can find applications of this methodology in several areas of knowledge such as engineering, economics and biomedical sciences. They are often used in microbiology to model growth curves and mainly in medicine. Cardiology, orthopedics, geriatrics, trauma, rehabilitation and cancer are only some examples of medical areas where neural networks were implemented [9] [10] [11]. These studies, in most cases, aim to predict a clinical outcome, such as, for example, death, related to a set of independent explanatory variables.

Due to the usual nature of the binary response variable, the ANNs are presented as an alternative to the so popular logistic regression model. Robust, easy to implement and usually with a very satisfactory performance, this model is not free of problems. In fact, the imposition of linear relationships to model the potential association between the dependent variable and each independent variables maybe misleading. To overcome this problem, Generalized Additive Models (GAMs) [12] were proposed. They are, undoubtedly, more flexible than the existing Generalized Linear Models (GLMs), as they allow nonlinear relationships between the

978-1-4673-6165-1/13/$31.00 © 2013 IEEE

independent variables and the response, modelled by smooth functions. There has been, thus, an evolution of regression models in order to improve their performance. Due to the existing parallelism between those models and ANNs, it is, therefore, natural that ANNs suffer too, an evolution. In fact, a new model of ANN architecture, inspired in GAMs was proposed by [13] - Generalized Additive Neural Network (GANN). One of the advantages of this model, when compared to a MLP architecture, is the mitigation of the black box perception regarding the interpretation of results [13]. In fact, with GANNs, the effect of each variable on the output can be interpreted by using partial residual plots [13]. In subsequent studies [14], improvements have been introduced, such as a fully automated algorithm, in order to alleviate both the subjectivity of partial residual plot analysis and the high time-consuming [14], [15].

II. Building an Artificial Neural Network

A. Multi Layer Perceptron

The most used ANN architecture is the MLP with one hidden layer. This architecture is considered to be an universal approximator [16] and, therefore, can serve as a reference for comparative studies with other methods.

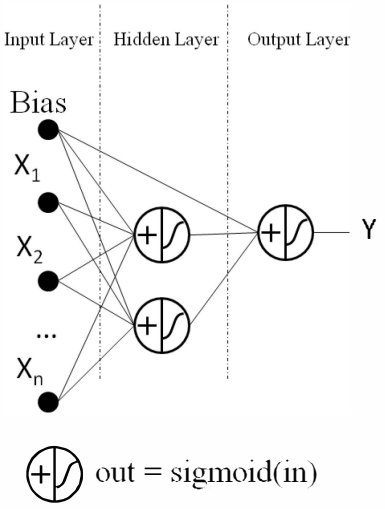

Fig. 1. Multi Layer Perceptron Architecture

The main goal of using a neural network is to build a model that will perform well on new data (generalization). This ability is related to a good non-linear interpolation of the input data [17]. This interpolation depends mainly on two factors - the size of the training dataset and the degree of flexibility. A too low number of nodes and an excessive degree of flexibility may lead to memorization, conducting, in both cases, to a poor generalization.

Therefore, the design of a neural network has shown to be very important to the quality of final results. To calculate the number of hidden nodes or the number of hidden layers, there are no guidelines (this still remains an open problem). At a first approach, heuristics can be used [18][19] as well as constructive and pruning algorithms (e.g. brain damage algorithm) or even Genetic Algorithms (GAs). However there is no direct method to calculate the optimal number of nodes and, despite the existence of several heuristics, many investigators defend that there is not a good way of determining a network topology if only the number of inputs and outputs are taken into account [20].

B. Generalized Additive Neural Networks

Despite the fact a GANN does not be a universal approximator [13], when combined with the correct choice of a link function, it constitutes a good additive model approximator. In fact, a GANN can be approximated to a GAM [15] which, in general, has a better performance compared to a GLM or to a MLP [21].

1) GANN architecture: A first approach of the GANN architecture was done by Sarle[22]. Later, Potts [13] included a method for construction and estimation of a GANN, inspired in the GAM equation:

$$g_0^{-1}(E(y_i)) = \beta_0 + f_1(x_{1i}) + f_2(x_{2i}) + ... + f_k(x_{ki}), \quad (1)$$

where g_0^{-1} represents the link function, E is the expected value, y_i is the target variable (e. g. death), f_1,...,f_k represents the partial functions, x_{1i},...,x_{ki} are the input variables (e.g. patients characteristics) and β_0 the overall bias. The architecture of a GANN, Fig. 2, results from the transposition of the GAM equation, Eq. (1), to a neuronal model [15]; each partial function corresponds to a MLP (due to its property of being an universal approximator to any continuous function) with an input, an output and a single hidden layer. In an improved version of the MLP, a skip layer is inserted so that the GANN

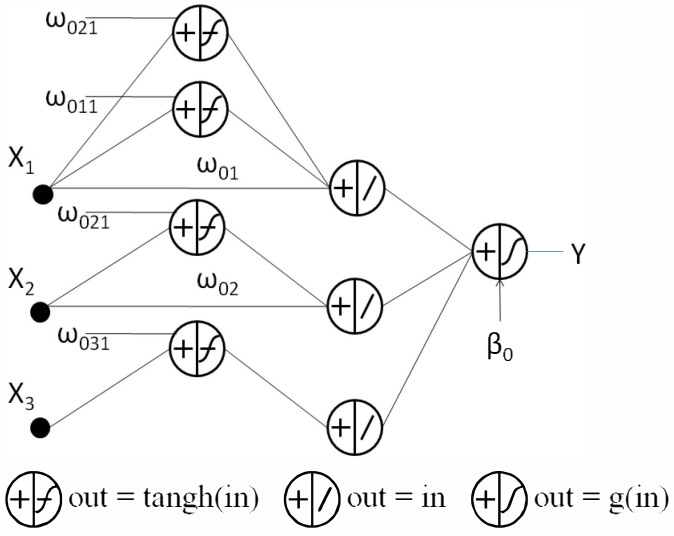

Fig. 2. Generalized Additive Neural Network Architecture

may include the linear model as a special case. The resulting equation for each MLP is given by:

$$f_j(x_{ji}) = \omega_{0j}x_{ji} + \omega_{1j}tanh(\omega_{01j} + \omega_{11j}x_j) + ... \\ +\omega_{hj}tanh(\omega_{0hj} + \omega_{1hj}x_{ji}), \quad (2)$$

where ω represents the network parameter. The flexibility of a MLP depends on the number of nodes in the hidden layer. For a good fit of the model, it will be necessary to find the optimal number of nodes for each of the MLPs. It will be the adequate amount of flexibility in all the MLPs that will guarantee a good fit of the model. Thus, in order to find the appropriate number of hidden nodes, Potts proposed a recipe based on the visual inspection of the partial residual plots defined by the following equations:

$$pr_{ji} = g_0^{-1}(y_i) - \hat{\beta}_i - \sum_{l \neq j} \hat{f}_l(x_{li}), \quad (3)$$

$$pr_{ji} = (g_0^{-1}(y_i) - g_0^{-1}(\hat{y_i})) + \hat{f}_j(x_{ji}), \quad (4)$$

Assuming that g_0^{-1} is nonlinear, a first order approximation (5) can be used [13]:

$$pr_{ji} = \frac{\partial g_0^{-1}}{\partial y}(\hat{y_i}) \times (y_i - \hat{y_i}) + \hat{f}_j(x_{ji}), \quad (5)$$

where pr_{ji} represents the partial residuals, $\hat{y_i}$ is the estimated output, y_i is the observed output and $\hat{f}_j$ is the estimated output of each MLP. The proposed algorithm by [13] was:

1- Construct a GANN with one neuron and a skip layer for each input (See Eq. (2)) assuming that they are already standardized.
2- This gives 4 parameters (degrees of freedom (df)) for each input. Binary inputs (dummy variables) only have a direct connection (1 df).
3- Fit a generalized linear model to give initial estimates of β_0 and ω_{0j}.
4- Initialize the remaining 3 parameters in each hidden layer using random values from a normal distribution with mean zero and variance equal to 0.1.
5- Fit the full GANN model.
6- Examine each of the fitted partial functions overlaid over their partial residuals.
7- Prune the hidden layers with apparently linear effects an add neurons to hidden layers where the nonlinear trend appears to be under-fitted. If this step is repeated, the final estimates from previous fits can be used as starting values.

However, the performance of this interactive construction algorithm depends on the subjective analysis of partial residual plots. Thus, Du Toit [23] proposed an improved algorithm where the method of partial residual plots is not used in the process of model building. This improved algorithm is based on finding models using objective model selection criteria or cross-validation. Du Toit mentions that, if adequate time to evaluate candidate models is given, this search technique is complete, optimal and has a good performance compared to other non-linear model selection techniques [24].

TABLE I
GANN ENCODING

Symbol	Description
0	No MLP (input removed from model)
1	MLP with a skip layer
2	MLP without skip layer and one hidden node
3	MLP with skip layer and one hidden node
4	MLP without skip layer and two hidden nodes
5	MLP with skip layer and two hidden nodes
6	MLP without skip layer and three hidden nodes
7	MLP witht skip layer and three hidden nodes
8	MLP without skip layer and four hidden nodes
9	MLP with skip layer and four hidden nodes

2) Automated Construction of GANNs: From the point of GANN's view, each MLP has characteritics that will define its final behavior. In the other hand, MLP features are defined by the number of neurons in the hidden layer and the presence or absence of skip layer. These characteristics can be encoded in order to facilitate the process of model selection. As it can be seen later, GAs can be used for model selection by using an encoding system, that, in turn, translates the topological GANN's characteristics. Thus, an architecture with j variables can be encoded as a string with a number of j symbols. Each symbol is limited to a given number. As an example, DuToit and de Waal [23], [14], [15] use a table of 10 symbols for coding some possible GANN's state (any GANN architecture, that corresponds to a structure which includes a number of nodes in each MLP, with or without a skip layer). The proposed encoding schema is given in Table I.

Considering, as an example, the GANN model depicted in Fig. 2: the first variable is represented by a MLP with two hidden neurons with skip layer, the second MLP is represented by a hidden neuron with a skip layer, and the third one has just an hidden neuron, the resulting string that represents the model will, then, be "532".
Concerning the process of model selection, the automated construction of a GANN is an iterative process that includes a set of tasks in each loop and it is described in [14] and [15]. This process aims to achieve an optimal GANN model. Each cycle begins with an initial state, which can start *e.g.* by only a skip layer for each MLP (linear model), or begin a state after an intelligent search. These may use a guess algorithm (a pre-processing step) based on the analysis of results of a stepwise regression, thereby reducing the search space to reach the optimum state.

The displacement in the search space can be arranged in various ways. Du Toit proposes the demand for new states from a parent state (expanded node) through small changes in its string. Those changes should be restricted to slight modifications in each string digit, corresponding to small

changes in GANN architecture (*e.g.* introducing one neuron in one of the MLPs). In this process, given a GANN's state, all possibilities in the next neighborhood are explored. According to a performance indicator, used as model selection criterion (*e.g.* cross-validation error), the best state is selected and will be the next expanded node. The expansion does not include duplicate states, thus, the process may be organized through a search tree where the nodes are organized according to the performance measurement, without duplications. The search ends when all possibilities are covered or when a maximum time is reached. The main disadvantage of this method is time consuming, especially if the number of variables is too large. In this case, a multistep approach can be used in addition to the described above, in which all changes that have been more advantageous may be implemented collectively, conducting to better results [15]. So, in addition, a state that includes collectively the changes that have been proven to be better than the parent node, should also be tested in the same loop. For applications with a large number of variables, the search space is exponentially proportional to the dimensions of the problem and, therefore, exhaustive search methods are very time consuming. Thus, the use of stochastic techniques (*e.g.* GAs) that attempt to discover near-optimal solutions within acceptable time, are highly useful.

III. Genetic Algorithms

GAs are part of a set of methods (evolutionary algorithms) where, inspired by natural selection, competition among individuals results in a struggle for survival [25]. The individuals most adapted to the environment, i.e., those having characteristics that give them a greater degree of adaptability (fitness), are able to reach adulthood, becoming parents. These characteristics are determined primarily by their genes. After becoming parents, they may transmit their genes to future generations by exchanging genetic material (recombination). After this stage, a new population is generated. The new individuals will compete in the next generation. Sometimes an individual may mutate, corresponding to a small change in some of the genes. This change may confer greater fitness, contributing to its selection.

From the computational point of view, each individual is represented by a chromosome, which, in turn, is formed by a well-defined array of genes. Each gene may be represented by a symbol or value depending on the used encoding schema (e.g. Table I). From a population with a fixed number of chromosomes, some are selected considering their fitnesses, based on a performance measure (*e.g.* cross-validation error). The selection may have other variants, such as the possibility (under certain conditions), of individuals with a lower fitnesses being selected. After selection, parents should be coupled randomly in pairs and generate offsprings which inherits some of the characteristics of both parents. At last, a new population may be formed, *e.g.*, with the best chromosomes and offsprings [25]. These steps are repeated in several iterations (generations) until a stop criterion is reached.

IV. Methodology

A. Dataset and pre-processing

For this study, data from 996 patients, obtained in an ICU, was used. SAPS II scores were collected in the first 24 hours of admission. The SAPS II is a severity of illness score, used in intensive care units, that has received a lot of attention in Europe for its simplicity and applicability. It includes 17 variables: 12 physiology variables (heart rate, systolic blood pressure, body temperature, the ratio $\frac{Pao_2}{Fio_2}$ for ventilated patients, urinary output, serum urea level, WBC count, serum potassium, serum sodium level, serum bicarbonate level, bilirubin level and Glasgow coma score), age, type of admission (scheduled surgical, unscheduled surgical or medical) and three underlying disease variables (acquired immunodeficiency syndrome, metastatic cancer and hematologic malignancy). Categorical variables were recoded using dummy variables, resulting in a dataset with 19 covariates. Data were normalized with values between 0 and 1.

B. Model Selection

In order to search an optimal neural network model, different methods were implemented with out-of-sample model selection by using the Mean Squared Error (MSE) obtained from a 5-fold cross-validation. Because the MLP architecture is the most popular, it was chosen as benchmark in the experiments. Thus, a MLP model with 19 inputs corresponding to the number of variables and one ouput was also implemented. In order to find out the number of hidden layer nodes corresponding to a good MLP model, a set of models was generated with a topology that varied from one neuron in hidden layer until a maximum of fifty (in order to also include models provided by some heuristics that can be found in [18]). The model that presented a better cross-validation error was chosen. The best MLP model found has 31 nodes in the hidden layer.

Concerning the GANN model, the topology included 19 MLPs corresponding to the number of input variables. In the output, the logit link function was used:

$$g_0^{-1}(E(Y)) = ln\left(\frac{E(Y)}{(1-E(Y))}\right). \tag{6}$$

For finding an optimal GANN model, two methods were compared. The first one is related to the AutoGann in [23], [14], [15]. The second used method was a standard GA [25] applied to the GANN architecture. Both search algorithms applied to the GANN finished after a predetermined time. This time was equal regardless of the used search method. The referred models will be denoted by AutoGANN and Genetic, respectively.

C. AutoGANN

The use of AutoGANN method involved a multistep iteration where a child node was built in each iteration, based on those children whose fitness was better than the one presented by the parent node. The initial state corresponds to a GLM (just a skip layer in each MLP).

TABLE II
NEW GANN ENCODING

Symbol	Description
null	No MLP (input removed from model)
0	MLP with a skip layer
-1	MLP without skip layer and one hidden node
1	MLP with skip layer and one hidden node
-2	MLP without skip layer and two hidden nodes
2	MLP with skip layer and two hidden nodes
-3	MLP without skip layer and three hidden nodes
3	MLP with skip layer and three hidden nodes
-4	MLP without skip layer and four hidden nodes
4	MLP with skip layer and four hidden nodes

D. Genetic Algorithm

The algorithm that was used can be found in [25]. For each generation, the steps listed below were followed:

1) Generation of Population: Initial population is generated with ten chromosomes. This population size is maintained over generations. Each gene can assume random values from {-2,-1,0,1,2}, meaning that the state of GANN can vary from 0 to 2 nodes in each hidden layer and may include or not a skip layer.

2) Evaluation: Each chromosome is evaluated in order to find its fitness.

3) Selection of parents: In this step, among the top five chromossomes with better fitness, four should be randomly chosen to an empty set of parent chromosomes. Among the five with worst fitness, two should be chosen to the same set.

4) Passage of genes from parents to children: The fundamental principle is that identical characteristics between parents should pass to the children. Before applying this principle, it is essential to find an encoding system that facilitates the development of the algorithm, namely the process of children generation as well as the mutation process. The proposed encoding schema (Table II), different from the one proposed by DuToit and de Waal (Table I), focuses on the main characteristics that define a GANN's state - the existence of a skip layer, and the number of nodes in each hidden layer.

The first feature is boolean and the second one is numeric. Therefore, the system used in this study combines these two features in the same symbol, assigning a positive number if the MLP includes a skip layer or negative in the opposite case. The absolute part of the number represents the number of hidden nodes in a MLP. When a MLP only includes a skip layer, zero will be used. Thus, highlighting the two characteristics in the same gene will allow, for example, in the case of both parents having the same number of nodes in the hidden layer (absolute part of gene's value), but one of the parents has a skip layer and the other not, the children inherits the number of nodes, but the existence of a skip layer (the signal of gene's value) will be determined randomly. The same will happen if they both have the last characteristic in common, for example, no skip layer (negative signal) with a different number of nodes in the hidden layer. In this case, the corresponding children's gene will have a negative signal with a number of nodes randomly chosen. Because the encoding schema includes negative numbers, it is desirable, from now on, to represent the GANN state by using a vector rather than a string. Using the example of the GANN architecture in Fig. 2, according to the enconding schema proposed in Table II, the GANN state is represented by the vector (2,1,-1). When variable selection takes place and the input is removed from the model, a non-numerical value could be used, such as "null". However, because all SAPS II variables will have to be considered in the models, this feature will not be used in the present study. In each generation, the six selected parent chromosomes are randomly coupled in pairs. Each couple is combined resulting in two offsprings. Six children result from each generation.

5) Mutation: the probability of inherited mutation of a gene in children was considered 0.1 (the value of the gene is modified to a random value).

6) Generation of new Population: The fittest chromosomes from previous population and the six children above generated, will form together the new population for the next generation.

7) Steps from 2 to 6 are performed until the number of iterations corresponds to the predefined time or a maximum number of generations.

E. Statistical methods for model comparison

In order to compare the models under study, predictive and discriminative abilities were studied. Predictive performance was evaluated through the analysis of the agreement between the estimated probabilities of death and the actual observed mortality by using MSE. The discriminative performance (*i.e.* ability of the model to distinguish between patients who live from patients who die) was measured by the area under the Receiver Operating Characteristic (ROC) curve [26]. A value of 0.50 is obtained when a model discriminates no better than chance, and a value of 1.0 means perfect accuracy [27]. To complement the study of model's performance, predictiveness curves were also calculated [28]. This curve is a graphical representation of the distribution of risk given a marker (in our case, marker values correspond to the estimated probabilities of death). This risk is defined by the following conditional probability: $risk(y) \equiv P[D = 1|Y = y]$, where D denotes the binary outcome (in our case, death) and $Y = y$ a marker value (in our case, predicted probabilities of death). A marker that is useless assigns equal risk to all individuals and hence, the corresponding predictiveness curve is an horizontal line at the prevalence of the outcome; on the other hand, a marker that is highly informative about risk, originates a predictive curve that is close to a step function.

To compare the several models, continuous Net Reclassification Improvement (continuous NRI) and Integrated Discrimination Improvement (IDI) with corresponding bootstrap 95% confidence intervals were also calculated. NRI is a prospective measure which quantifies the correctness of upward and downward reclassification or movement of predicted probabilities as a result of adding a new marker (or

using a new estimation method) and IDI may be viewed as a difference between improvement in average sensitivity and any potential increase in average 1-specificity (it measures the increase in the difference in average predicted risks between the individuals with and without the outcome) [29].

Data analysis was performed on a personal computer with an Intel processor icore3, operating at 3.33 GHz with 4 GB of RAM, running Microsoft Windows 7, 32 bit. All algorithms have been implemented in Java using the Eclipse IDE. However, some statistical functions that can be found in R software [30] were also integrated into the Java program by using the library RJava (*e.g.* the software that was used to implement the GLMs). All the statistical methods that were used to compare the performances of the three models were implemented by using R software [30]. A p-value < 0.05 was considered significant and 95% confidence intervals were calculated whenever appropriate.

V. Results

The mean of patients ages was 60.3 (95% confidence interval: 59.3 - 61.4) years. The median SAPS II was 41 (interquartile range 20-60) and a mortality rate of 36%.

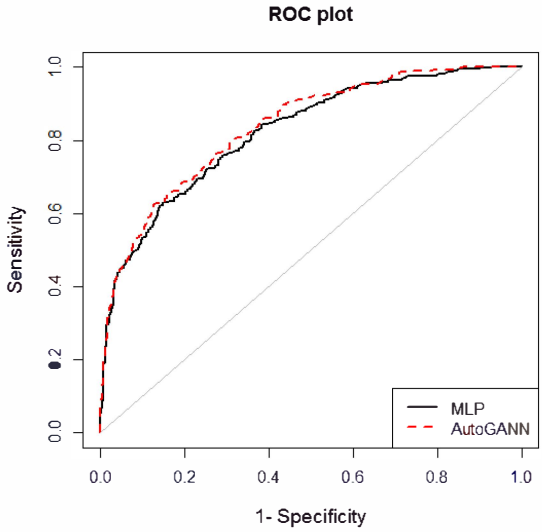

Fig. 3. ROC MLP AutoGANN

Concerning the predictive ability of the models, measured by the 5-fold cross-validation MSE, it may be seen that the MLP model had a higher value than the other two. Regarding the discriminative ability of the models, results showed that all models had a better performance than the MLP model. By Fig. 3 and Table III it can be seen that the model obtained by the AutoGANN had the highest AUC and was clearly different from MLP model's AUC (p-value=0.004). Comparing MLP model with the model obtained by Genetic, statistical differences remained (p-value=0.022). No differences were found between the genetic method and the AutoGANN (p-value=0.19).

TABLE III
PREDICTIVE AND DISCRIMINATIVE PERFORMANCE MEASURES

Model	MSE	AUC	95%C.I.
MLP	0.1600	0.821	0.794 - 0.848
AutoGANN	0.1545	0.835	0.810 - 0.861
Genetic	0.1559	0.832	0.806 - 0.857

TABLE IV
NRI AND IDI

Model comparison	Total NRI(95%C.I.) $(p-value)$	Total IDI (95%C.I.) $(p-value)$
MLP vs AutoGANN	54.6%(41.7, 67.6) $(p < 0.001)$	0.0298(0.0199, -0.0397) $(p < 0.001)$
AutoGANN vs Genetic	-7.18%(-20.1, 5.8) $(p = 0.277)$	-0.0021(-0.0074, 0.0032) $(p = 0.442)$

TABLE V
NRI EVENTS/NO-EVENTS

Model comparison	NRI events(95%C.I.) $(p-value)$	NRI no-events(95%C.I.) $(p-value)$
MLP vs AutoGANN	25.9%(15.6,36.2) $(p < 0.001)$	28.7%(21.0,36.5) $(p < 0.001)$
AutoGANN vs Genetic	5.85%(-4.5, 16.2) $(p = 0.268)$	-13.03%(20.8,-5.3) $(p = 0.001)$

Results concerning NRI and IDI also confirm that AutoGANN has the best performance (see Table IV and Table V). When compared to MLP, a huge increase in total NRI was found (NRI=54.6%; $p < 0.001$). This reflects the correct changes in the relative order of the calculated risk of those with and without events (NRI events= 25.9%; $p < 0.001$ and NRI no-events= 28.7%; $p < 0.001$, respectively). This means that, when AutoGANN was used, an increase of the calculated risk in 25.9% of those who died was observed as well as a reduction of the calculated risk in 28.7% of those who didn't died. The value of the IDI, which considers the size of the referred changes, was also significant (IDI=0.0298; $p < 0.001$). When comparing the AutoGANN with the Genetic, although NRI and IDI total value was negative indicating a worse performance of this algorithm, no overall significant differences were detected (Table IV). However, if events/no-events NRI components are analysed, a significant reduction of NRI for no-events was obtained (NRI no-events= -13.0%; $p = 0.001$). By the analysis of the predictiveness curves in Fig. 4, a confirmation of previous results was found. In fact, the MLP model originated the worse results (higher risk estimates for low percentiles of risk and lower risk estimates for high percentiles of risk) and, AutoGANN and Genetic models were similar.

VI. Conclusion

After carrying out several performance measurements, it was found that the AutoGANN obtained a better model than

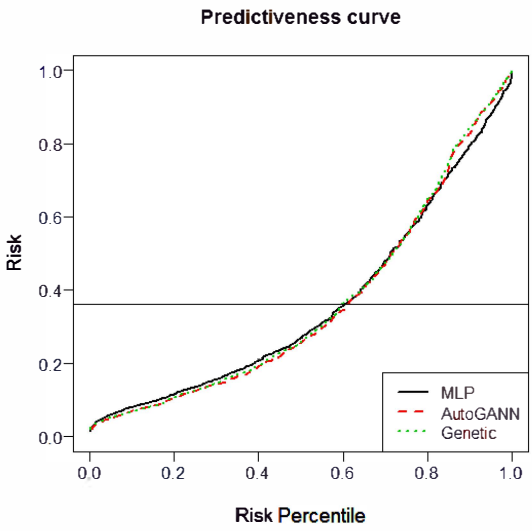

Fig. 4. Predictiveness curves of the estimated probabilities of death by the three models

the best MLP model attained with 31 nodes in the hidden layer. Indeed, comparing their predictive and discriminative indicators, the GANN model presented better results than the MLP model, despite the search methods that were used. In this study it was also shown that the architecture GANN is surely a good alternative to the MLP. After comparing the two GANN search methods, the Genetic method emerges as a potential alternative to AutoGANN. AutoGANN and GAs have in common the fact that both methods perform a search in a population of candidates through an iterative process and, using a selection criterion, both retain the characteristics most promising for future generations. In the other hand, the AutoGANN is based on small changes along generations, meaning that the convergence process is very slow and more subject to falling into local minima. Using a GA, the search begins for a more spreaded throughout the search space and may found alternative solutions inside the population of candidates. However, given a convergence zone, it may result in a good solution but not in the best one. AutoGANN may be used in situations that require a local search, while the GA is more suited to a global search, founding possible alternatives. These roles may be complemental and, because it was shown that the GA and the AutoGANN are very similar regarding predictive and discriminative performances, we propose, as future work, the use of hybrid methods that combine the two approaches together in a single search method.

ACKNOWLEDGMENT

Research was partially sponsored by national funds through the Fundação Nacional para a Ciência e Tecnologia, Portugal - FCT under the projects PEst-OE/MAT/UI0006/2011.

REFERENCES

[1] S. Lemeshow, D. Teres, H. Pastides, J. S. Avrunin, and J. S. Steingrub, "A method for predicting survival and mortality of icu patients using objectively derived weights." *Crit Care Med*, vol. 13, no. 7, pp. 519–25, 1985. [Online]. Available: http://www.biomedsearch.com/nih/method-predicting-survival-mortality-ICU/4006490.html

[2] A. J. Lemeshow S, Teres D and G. RW., "Refining intensive care unit outcome prediction by using changing probabilities of mortality." *Crit Care Med*, vol. 1, pp. 470–477, 1988.

[3] S. Lemeshow, "Mortality Probability Models (MPM II) based on an international cohort of intensive care unit patients," *Jama-journal of The American Medical Association*, vol. 270, pp. 2478–2486, 1993.

[4] W. A. Knaus, D. P. Wagner, E. A. Draper, J. E. Zimmerman, M. Bergner, P. G. Bastos, C. A. Sirio, D. J. Murphy, T. Lotring, and A. Damiano, "The apache iii prognostic system. risk prediction of hospital mortality for critically ill hospitalized adults." *CHEST Journal*, vol. 100, no. 6, pp. 1619–1636, 1991. [Online]. Available: +http://dx.doi.org/10.1378/chest.100.6.1619

[5] J. R. L. Gall, "A new Simplified Acute Physiology Score (SAPS II) based on a European/North American multicenter study," *Jama-journal of The American Medical Association*, vol. 270, pp. 2957–2963, 1993.

[6] P. Metnitz, R. Moreno, E. Almeida, B. Jordan, P. Bauer, R. Campos, G. Iapichino, D. Edbrooke, M. Capuzzo, J.-R. Le Gall, and , "Saps 3from evaluation of the patient to evaluation of the intensive care unit. part 1: Objectives, methods and cohort description," *Intensive Care Medicine*, vol. 31, pp. 1336–1344, 2005, 10.1007/s00134-005-2762-6. [Online]. Available: http://dx.doi.org/10.1007/s00134-005-2762-6

[7] J. Zimmerman, A. Kramer, D. McNair, and F. Malila, "Acute physiology and chronic health evaluation (apache) iv: hospital mortality assessment for today's critically ill patients." *Crit Care Med*, vol. 34, no. 5, pp. 1297–310, 2006.

[8] C. Bishop, *Neural networks for pattern recognition*. Clarendon Press, 1995.

[9] P. J. G. Lisboa, "A review of evidence of health benefit from artificial neural networks in medical intervention," *Neural Netw.*, vol. 15, no. 1, pp. 11–39, Jan. 2002. [Online]. Available: http://dx.doi.org/10.1016/S0893-6080(01)00111-3

[10] M. Suka, S. Oeda, T. Ichimura, K. Yoshida, and J. Takezawa, "Neural networks applied to medical data for prediction of patient outcome," in *Trends in Intelligent Systems and Computer Engineering*, ser. Lecture Notes in Electrical Engineering, O. Castillo, L. Xu, and S.-I. Ao, Eds. Springer US, 2008, vol. 6, pp. 309–325. [Online]. Available: http://dx.doi.org/10.1007/978-0-387-74935-8_23

[11] M. Cacciafesta, F. Campana, G. Piccirillo, P. Cicconetti, I. Trani, R. Leonetti-Luparini, V. Marigliano, and P. Verico, "Neural network analysis in predicting 2-year survival in elderly people: a new statistical-mathematical approach," *Archives of Gerontology and Geriatrics*, vol. 32, no. 1, pp. 35–44, Jan-Feb 2001. [Online]. Available: ⟨GotoISI⟩://000168006600004

[12] T. Hastie and R. Tibshirani, *Generalized additive models*, ser. CRC Monographs on Statistics & Applied Probability. Chapman & Hall/CRC, 1990.

[13] W. J. E. Potts, "Generalized additive neural networks," in *Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining*, ser. KDD '99. New York, NY, USA: ACM, 1999, pp. 194–200. [Online]. Available: http://doi.acm.org/10.1145/312129.312228

[14] D. de Waal and J. du Toit, "Generalized additive models from a neural network perspective," in *Data Mining Workshops, 2007. ICDM Workshops 2007. Seventh IEEE International Conference on*, oct. 2007, pp. 265 –270.

[15] D. A. de Waal and J. V. du Toit, "Automation of generalized additive neural networks for predictive data mining." *Applied Artificial Intelligence*, vol. 25, no. 5, pp. 380–425, 2011. [Online]. Available: http://dblp.uni-trier.de/db/journals/aai/aai25.html

[16] K. Hornik, M. Stinchcombe, and H. White, "Multilayer feedforward networks are universal approximators," *Neural Netw.*, vol. 2, pp. 359–366, July 1989. [Online]. Available: http://dl.acm.org/citation.cfm?id=70405.70408

[17] S. Haykin, *Neural Networks: A Comprehensive Foundation*. Prentice Hall PTR, 1998.

[18] S. Walczak and N. Cerpa, "Heuristic principles for the design of artificial neural networks," *Information and Software Technology*, vol. 41, no. 2, pp. 107–117, 1999.

[19] E. B. Baum and D. Haussler, "What size net gives valid generalization?" *Neural Computation*, vol. 1, no. 1, pp. 151–160, 1989.

[20] A. K. Sharma, "Effectiveness of heuristic rules for model selection in connectionist models to predict milk yield in dairy cattle," *TECHNIA - International Journal of Computing Science and Communication Technologies*, vol. 2, no. 1, pp. 384–386, July 2009.

[21] A. Papoila, C. Rocha, C. Geraldes, and P. Xufre, "Generalized linear models, generalized additive models and neural networks: Comparative study in medical applications," in *Advances in Regression, Survival Analysis, Extreme Values, Markov Processes and Other Statistical Applications*, dec. 2012.

[22] W. S. Sarle, "Neural networks and statistical models," 1994.

[23] J. V. Du Toit, "Automated construction of generalized additive neural networks for predictive data mining." Ph.D. dissertation, School for Computer, Statistical and Mathematical Sciences, North-West University, South Africa, 2006.

[24] J. V. du Toit and D. A. de Waal, "Spam detection using generalized additive neural networks," in *Southern Africa Telecommunication Networks and Applications Conference (SATNAC) 2010*, september 2010.

[25] F. Hillier and G. Lieberman, *Introduction To Operations Research*. McGraw Hill Higher Education, 2005.

[26] A. P. Bradley, "The use of the area under the roc curve in the evaluation of machine learning algorithms," *Pattern Recognition*, vol. 30, pp. 1145–1159, 1997.

[27] H. JA and M. BJ, "The meaning and use of the area under a receiver operating characteristic (roc) curve," *Radiology*, vol. 143, pp. 29–36, 1982.

[28] Y. Huang, M. Pepe, and Z. . Feng, "Evaluating the predictiveness of a continuous marker," *Biometrics*, vol. 63, 1181-1188.

[29] M. Pencina, R. D'Agostino, and E. . Steyerberg, "Extensions of net reclassification improvement calculations to measure usefulness of new biomarkers," *Statistics in Medicine*, vol. 30, pp. 11–21.

[30] R Development Core Team, *R: A Language and Environment for Statistical Computing*, R Foundation for Statistical Computing, Vienna, Austria, 2012, ISBN 3-900051-07-0. [Online]. Available: http://www.R-project.org