Annals of the University of Craiova, Mathematics and Computer Science Series
Volume 37(2), 2010, Pages 27–37
ISSN: 1223-6934

A two stage decision model for breast cancer detection

Smaranda Belciug

Abstract. The use of computer technology supporting medical decision is now widespread and pervasive across a broad range of medical areas. Accordingly, computer-aided diagnosis has become an increasingly important area for intelligent computational systems. The aim of this paper is to present a two stage model containing several different neural networks: multi-layer neural perceptron (MLP), radial basis function (RBF) and Probabilistic Neural Networks (PNN), and the effectiveness of this system on a real breast cancer database, to support the medical decision process.

2010 Mathematics Subject Classification. Primary 92B20; Secondary 68T05.
Key words and phrases. machine-learning techniques, neural networks, supervised learning, breast cancer detection.

1. Introduction

Breast cancer is the second most common type of cancer, after lung cancer, (10.4% of all cancer incidences, both sexes counted (World Health Organization International Agency of Research on Cancer, June 2003)). Regarding cancer deaths, breast cancer situates on the fifth place. The statistics show that almost 1% of all deaths and 7% of cancer deaths were caused by breast cancer (World Health Organization, February 2006). The chance of a woman having invasive breast cancer some time during her life is a little less 1 in 8, and the chance of dying from breast cancer is about 1 in 35. Due to earlier detection and better and improved treatment the death rates caused by this disease have been going down.
Breast cancer detection is achieved using non-invasive procedures: conventional imaging, nuclear imaging, such as magnetic resonance imagining, positron emission tomography or invasive procedures: tissue biopsy. The average accuracy of using such modern medical imaging methods for detecting breast cancer or recurrent events ranges from 80% to 90% (Gilbert et al.,[4], Lee et al., [14], Vranjesevic et al., [20]). In breast cancer detection, neural networks (NN) are used widespread, because providing a specific algorithm on how to identify the disease is not necessary. Hsiao *et al.* ([11]) (2009) trained a MLP classifier using the vascularity indices (harmonic and non-harmonic (3D) power Doppler imaging) for determining whether the breast tumors are benign or malign. Revett *et al.* ([17]) and Gorunescu *et al.* ([6]) designed a medical decision support system for breast cancer based on a hybrid model containing rough sets and probabilistic neural networks. Fogel *et al.* ([5]) (1997) trained a NN using evolutionary programming for the detection of breast cancer using radiographic features and patient age. Different types of NNs algorithms, such as MLP, RBF, PNN, combined neural network (CNN), recurrent neural network (RNN), and support vector machine (SVM), have been considered to statistically evaluate their potential contribution to

Received March 19, 2010. Revision received May 01, 2010.

predict recurrent events in breast cancer (Gorunescu et al., [7], [8], [9], Ibeyli, [12]). For the selection of the best predictors of breast malignant lesions among the normalized features NNs and logistic regression analysis were compared (McLaren, [15]). In a survival analysis problem, NN's results on two different breast cancer datasets (both of which using nuclear morphometric features) are compared, showing that NNs can successfully predict recurrence probability and separate patients with good and bad prognoses (Chi et al., [3]).
The aim of this paper is to provide a diagnosing procedure by managing a two stage decision model, a competitive one, and a collaborative one, containg several different NN, for providing a better diagnosis than each NN could accomplish alone. Even if this technique consisting in using both competition and collaboration between methods is not new, in this paper the novelty consists in the way the competitors collaborate. Concretely, a weighted voting system was chosen to provide the final decision. The first phase is represented by the competitive part which establishes the hierarchy of the NN, depending on their diagnosis accuracy. Thus only the best competitors will be considered to make the final decision. The second phase is based on a weighted voting system (WVS) applied on the selected NNs.
The remaining paper is organized as follows: Section 2 gives an overview of the intelligent system, detailing each phase competitive/collaborative. Section 3 gives a description of the breast cancer database on which the system has been applied on. Section 4 presents the results obtained, and a simulation on the system. The paper ends with Section 5 providing the conclusions of the study.

2. Two stage decision model

2.1. Competitive phase. In the competitive phase, each NN is applied on a breast cancer database. The following NNs are applied:
(a) Multi-layer perceptron (MLP);
(b) Radial basis function (RBF);
(c) Probabilistic neural networks (PNN).
Next, we shortly describe the three algorithms subsequently used.

2.1.1. *Multi-layer perceptron (MLP).* The artificial neural network is a collection of units, *neurons* or nodes, which are simple processors whose computing ability is restricted to a rule for combining input to calculate an output signal. Output signals may be sent to other units along connections known as *weights*. The net input of weighted signals received by a unit j is given by the formula: $net_j = w_0 + \sum_{i=1}^{n} w_{ij} \cdot x_i$, where w_0 is the biasing signal, w_{ij} the weight on the input connection ij, x_i the magnitude of signal on input connection ij and n is the number of input connections to unit j. The multi-layer perceptron (MLP) is the most popular neural network in use today.

Once the number of layers and number of units in each layer have been selected, the network's weights and thresholds must be set so as to minimize the prediction error made by the network. The cases belonging to the training dataset are used to automatically adjust the weights and thresholds in order to minimize this error; this process is equivalent to fitting the model represented by the network to the training data available. Thus, the error of a particular configuration of the network can be determined by running all the training cases through the network, comparing the actual output generated with the desired or target outputs. The differences are

combined together by an error function to give the network error. The most common error functions are the sum-squared error, where the individual errors of output units on each case are squared and summed together; this neatly summarizes the error over the entire training set and set of output units. Under these circumstances, the error surface is needed in order to find out its minimum. Concretely, each of the n weights and thresholds of the network (i.e. the free parameters of the model) is taken to be a dimension in space and the $(n+1)$th dimension is the network error. For any possible configuration of weights the error can be plotted in the $(n+1)$th dimension, forming an error surface. The objective of network training is thus to find the lowest point in this many-dimensional surface. Since the neural network error surfaces are complex being characterized by a number of unhelpful features (e.g. local minima, flat-spots and plateaus, saddle-points, and long narrow ravines, etc.), it is not possible to analytically determine where the global minimum of the error surface is, and so neural network training is essentially an exploration of the error surface. From an initially random configuration of weights and thresholds (i.e. a random point on the error surface), the training algorithms incrementally seek for the global minimum. Typically, the gradient (slope) of the error surface is calculated at the current point, and used to make a downhill move. Eventually, the algorithm stops in a low point, which may be a local minimum (but hopefully is the global minimum).
Among the learning algorithms used in the training process the most common one is the back-propagation algorithm, consisting of the following steps:
Step 1. *The gradient descent.* We iteratively update the weight vector such that, at step r we move a short distance in the direction of the greatest rate of decrease of the error, i.e. in the direction of the negative gradient. Denoting the weight vector increment Δw_{ij}, we use the formula: $\Delta w_{ij} = -\frac{\partial E}{\partial w_{ij}} = -\frac{\partial E}{\partial net_i} \cdot \frac{\partial net_i}{\partial w_{ij}}$, where E is the sum of squares error function. The first factor is the error signal for unit i: $\delta_i = -\frac{\partial E}{\partial net_i}$ and the second is $\frac{\partial net_i}{\partial w_{ij}} = \frac{\partial}{\partial w_{ij}} \sum_{k \in A_i} w_{ik} \cdot x_k = x_j$, where $A_i = \{j : \exists w_{ij}\}$ is the set of nodes anterior to unit i. Putting the two together, we get $\Delta w_{ij} = \delta_i x_j$. To compute this gradient, we thus need to know the value of δ_i for each hidden and output unit in the network.
Step 2. *Forward activation.* The activity of the input units is determined by the network's external input $\mathbf{x}$. For all other units, the activity is propagated forward: $y_i = f_i(\sum_{j \in A_i} w_{ij} \cdot x_j)$. Note that before the activity of unit i can be calculated, the activity of all its anterior nodes must be known. Since feedforward networks do not contain cycles, there is an ordering of nodes from input to output that respects this condition.
Step 3. *Calculating output error.* Assuming that we are using the sum-squared loss $E = \frac{1}{2}\sum_o (t_o - y_o)^2$, the error for output unit o is simply $\delta_o = t_o - y_o$.
Step 4. *Error backpropagation.* For hidden units, we must propagate the error back from the output nodes (hence the name of the algorithm).
Using the chain rule, we can expand the error of hidden unit in terms of its posterior nodes, $\delta_j = -\sum_{i \in P_i} \frac{\partial E}{\partial net_i} \cdot \frac{\partial net_i}{\partial y_j} \cdot \frac{\partial y_j}{\partial net_j}$, where $P_j = \{i : \exists w_{ij}\}$ is the set of nodes posterior to unit j. The first factor inside the sum is just the error of node i, the second is $\frac{\partial net_i}{\partial y_j} = \frac{\partial}{\partial y_j} \sum_{k \in A_i} w_{ik} \cdot y_k = w_{ij}$, while the third is the derivative of node j's activation function: $\frac{\partial y_j}{\partial net_j} = \frac{\partial f_j(net_j)}{\partial net_j} = f_j'(net_j)$. For hidden units u that use the activation function, we can make use of the special identity $(\tanh(u))' = 1 - \tanh(u)^2$, giving us

$f'_h(net_h) = 1 - y_h^2$. Putting all the pieces together, we get $\delta_j = f'_j(net_j) \sum_{i \in P_i} \delta_i w_{ij}$ [2].

Step 5. *Learning updates.* The synaptic weights are updated using the results of the forward and backward passes.

2.1.2. *Radial-basis function (RBF).* This intuitive NN consists of a hidden layer of radial units, each one of them modelling a Gaussian response surface. RBF offers an alternative approach to the MLP's use of hyperplanes to divide up the problem space, dividing the space using hyperspheres (quadrics with certain linear constrains on the coefficients), characterized by radii and centres. The activation of a hidden neuron is determined by the distance between the input vector and a prototype vector. This provides a smooth interpolation function in which the number of basis functions is determined by the complexity of the mapping to be represented rather than by the size of the data set. The modifications which are required are as follows:
1. The number M of the basis functions need not equal the number N of data points, and is typically much less than N.
2. The centres of the basis functions are no longer constrained to be given by input data vectors. Instead, the determination of suitable centres become part of the training process.
3. Instead of having a common width parameter σ, each basis function is given its own σ_j whose value is also determined during training.
4. Bias parameters are included in the linear sum. They compensate for the difference between the average value over the data set of the basis function activation and the corresponding average value of the targets.
Basically, considering a number M of basis functions $\phi_j(\mathbf{x}) = \exp(-\frac{||\mathbf{x}-\mu_j||^2}{2\cdot\sigma_j^2})$, where $\mathbf{x}$ is the input vector, μ_j is the vector determining the center of the basis function ϕ_j and σ_j is the width parameter, the RBF mapping is given by $y_k(\mathbf{x}) = \sum_{j=1}^{M} w_{kj}\phi_j(\mathbf{x})$. The training of a RBF is a two-step learning procedure. In the first step we determine the parameters of the basis function ϕ_j (i.e. μ_j and σ_j), and in the second step the weights w_{kj} are found. More details and techniques for optimization are to be found in [2] and [10].

2.1.3. *Probabilistic neural networks.* Different from the above two NNs, a probabilistic neural network (PNN) uses both the naïve Bayes decision methodology and the probability density functions associated with each decision class.

A. Bayes decision methodology.
Bayesian decision theory is a fundamental statistical approach to the problem of pattern classification. Minimizing the probability of error, or the expected risk, represents the traditional goal for decision strategies. The Bayes decision rule can be summarized as follows: *(a)* Let D_k be the decision rule related to the state of nature B_k; *(b)* Given measurement x, the error related to B_k is defined by $P(error|x) = 1 - P(B_k|x)$; Minimize the probability error; Bayes decision rule: *"Choose D_k if $P(B_k|x) > P(B_j|x)$, $\forall j \neq k$"* or, equivalently, *"Choose D_k if $P(x|B_k)P(B_k) > P(x|B_j)P(B_j)$,$\forall j \neq k$"*. The Bayes decision rule is applied to PNN as follows. Consider the general case of the q-category classification problem, in which the states of nature are denoted by $\Omega_1, \Omega_2, ..., \Omega_q$. The goal is to determine the class membership of a multivariate sample data represented by a p-dimensional random vector $\mathbf{x}$ into one of the q possible groups $\Omega_1, \Omega_2, ..., \Omega_q$, that is to make the decision $D(\mathbf{x}) = \Omega_i, i = 1, 2, ..., q$, where $\mathbf{x}$ represents a sample. If the multivariate probability density functions $f_1(\mathbf{x}), f_2(\mathbf{x}), ..., f_q(\mathbf{x})$,

the *a priori* probabilities $h_i = P(\Omega_i)$ of occurrence of patterns from categories Ω_i and the *loss* parameters l_i associated with all incorrect decisions given $\Omega = \Omega_i$, then, according to the Bayes decision rule, $\mathbf{x}$ is classified into the category Ω_i if the following inequality holds true: $l_i h_i f_i(\mathbf{x}) > l_j h_j f_j(\mathbf{x}), i \neq j$.
The accuracy of the decision depends straight on the accuracy of estimating the corresponding p.d.f.'s.
The way to using the Bayes decision rule to PNNs is represented by the technique chosen to estimate the p.d.f's $f_i(\mathbf{x})$ corresponding to each decision class Ω_i, based upon the training patterns set. The classical approach uses a sum of small multivariate Gaussian distributions, centered at each training sample, that is: $f_i(\mathbf{x}) = \frac{1}{(2\pi)^{n/2}\sigma^n} \cdot \frac{1}{m_i} \cdot \sum_{j=1}^{m_i} \exp(-\frac{d(\mathbf{x},x_j)^2}{2\sigma^2}), i = 1, 2, ..., q$, where m_i is the total number of training patterns in Ωi, p is the input space dimension and σ is the adjustable *"smoothing"* parameter using the training procedure. The smoothing or scaling parameter σ defines the width of the area of influence and should decrease as the sample size increases. The key factor in PNNs is therefore the way to determine the value of σ, since this parameter need to be estimated to cause reasonable amount of overlap. Commonly, the smoothing factor is chosen heuristically. For more details, see [20].
B. PNN algorithm
Step 1. For each class Ω_i compute the (Euclidian) distance between any pair of vectors and denote these distances by $d_1, d_2, ..., d_{r_i}$, where $r_i = C_{m_i}^2 = \frac{m_i!}{2!(m_i-2)!}$. For each class Ω_i compute the corresponding average distances and standard deviations $D_i = \frac{\sum_{j=1}^{r_i} d_j}{r_i}$, $SD_i = \sqrt{\frac{\sum_{j=1}^{r_i}(d_j - D_i)^2}{r_i}}$. For each class Ω_i compute the corresponding confidence intervals $I_{\Omega_i} = (D_i - 3SD_i, D_i + 3SD_i)$ for the average distances.
Step 2. For each decision class Ω_i consider the decision functions $f_i(\mathbf{x}) = \frac{1}{(2\pi)^{n/2}\sigma^n} \cdot \frac{1}{m_i} \cdot \sum_{j=1}^{m_i} \exp(-\frac{d(\mathbf{x},x_j)^2}{2\sigma^2})$. Assign (σ, D_i), $i = 1, 2, ..., q$.
Step 3. In each decision class Ω_i (randomly) choose a certain vector x_i^0, fix it and compute all distances $D_{\Omega_i,j} = d(x_i^0, x_j)$, $j = 1, 2, ..., m_i$.
Compute $f_i(x_i^0) = \frac{1}{(2\pi)^{n/2} D_i^n} \cdot \frac{1}{m_i} \cdot \sum_{j=1}^{m_i} \exp(-\frac{D_{\Omega_i j}^2}{2D_i^2})$.
Step 4. (*Bayes decision rule*) Compare $f_i(x_i^0)$ and $f_j(x_i^0)$, for all $i \neq j$, following the algorithm: "IF $l_i h_i f_i > l_j h_j f_j$ (for all $i \neq j$) THEN $x_i^0 \in \Omega_i$ ELSE IF $l_i h_i f_i \leq l_j h_j f_j$ (for some $i \neq j$)THEN $x_i^0 \notin \Omega_i$". For each (fixed) decision class Ω_i consider the 3-valued logic: TRUE - if $l_i h_i f_i > l_j h_j f_j$ (for all $j \neq i$) UNKNOWN - if $l_i h_i f_i = l_j h_j f_j$ (for some $j \neq i$)and FALSE - otherwise. l_i, h_i are the model parameters (costs and prior probabilities).
Step 5. Repeat step 3 for another choice for x_i^0 in Ω_i until all of them are chosen. Repeat step 3 for all vectors x_j^0 in Ω_j for all $j \neq i$. Obtain the classification accuracy in percentage.
Step 6. *(Estimating optimal smoothing parameter)* Divide each confidence interval I_{Ω_i} by N dividing knots into $(N+1)$ equal sectors with length $\Delta_i = 6SD_i/N$. Repeat step 3 by assigning $\sigma = D + k\Delta_i$ and $\sigma = D - k\Delta_i$, $k = 1, 2, ..., N$. If the current value of σ exceeds I_{Ω_i}, then STOP. Obtain the corresponding classification accuracy. Compute the maximum value MAX of the variable corresponding to TRUE out of the N cases.
Step 7. The smoothing parameters σ, of each class, corresponding to MAX represent

the optimal values of the smoothing parameters σ 's for each decision category Ω_i, $i = 1, 2, ..., q$.
The results, diagnosing accuracies, obtained after the above NNs were applied were statistically evaluated in order to arrange them in a hierarchical order.

2.2. Collaborative phase. It is common practice in the application of neural networks to train many different candidate networks and then to select the best, on the basis of performance on an independent validation set for instance, and to keep only this network and to discard the rest. There are two disadvantages with such an approach. First, all of the effort involved in training the remaining networks is wasted. Second, the generalization performance on the validation set has a random component due to the noise on the data, and so the network which had the best performance on the validation set might not be the one with the best performance on new test data. These drawbacks can be overcome by combining the networks together to form a *committee* ([16], [17]). The importance of such an approach is that it can lead to significant improvements in the predictions on new data, while involving additional computational effort. In fact, the performance of a committee can be better than the performance of the best single network used in isolation, [2].
Taking into account the above comments we considered the notion of competitive/ collaborative system as a committee.
After the competitive phase is finished, and the hierarchy established, we retain only the best classifiers. In general terms, when using an initial number of n competitors, only a certain number k (best) competitors will be retained for the collaborative step. The number k is chosen by the user, depending on the result of the competitive phase, and on the concrete problem to solve etc. In this paper, since only three NN techniques have been used, we have kept all of them.
In the WVS applied in this study, the values of the weights are directly proportional with the testing performances. Finally, after applying the WVS, the overall automatic diagnosis can be established for a new patient. ([11])

3. Breast cancer database

The model has been tested on the Wisconsin Prognostic Breast Cancer - WPBC, consisting of 683 cases with two decision classes: benign 444 (65%) and malign 239 (35%) instances. The database contains nine ordinal (categorical) attributes (for details concerning this database, see http://archive.ics.uci.edu/ml/machine-learning-databases/breast-cancer-wisconsin/).
Below, two samples from the database, one from each class, are presented, in order to visualize the two typical situations:

Table 1. Example of benign/malign samples

Attributes	**Patient 1**	**Patient 2**
Diagnosis	benign	malign
Clump Thickness	5	8
Uniformity of Cell Size	1	10
Uniformity of Cell Shape	1	10
Marginal Adhesion	1	8
Single Epithelial Cell Size	2	7
Bare Nuclei	1	10
Bland Chromatin	3	9
Normal Nucleoli	1	7
Mitoses	1	1

4. Results

For MLP we have used a three layered network, the first hidden layer contained 8 neurons, the second hidden layer contained 4 neurons. The initial weights were randomly chosen from the interval $[0, 0.1]$. For RBF we have used one hidden layer which contained 49 neurons. Since the first phase of the system must compare the diagnosing performances of the three machine learning (ML) techniques, an *a priori* statistical power analysis (*two-tailed* type of null hypothesis) has been performed because the appropriate sample size needed to be determined in order to achieve the adequate power. A sample of 100 different computer runs is considered for each ML technique, providing a statistical power equaling 99% with type I error $\alpha \in 0.05$ for the comparison test subsequently used.
For the evaluation of the classification efficiency, two metrics have been computed: the *training performance* (i.e. the proportion of cases that are correctly classified in the training phase) and the *testing performance* (i.e. the proportion of cases that are correctly classified in the testing phase. As a verification method, the 10-fold cross-validation is used ([13]). Thus, the classification accuracy (both training and testing) is computed 10 times, each time leaving out one of the sub-samples from the computation and using it as a test sample for cross-validation, so that each sub-sample is used 9 times as training sample and just once as testing sample (complete cross-validation cycle). The NN correct classification rates, computed for each of the 10 runs of the NN model, are then averaged to give the 10-fold estimate of the classification accuracy. Basically, each NN model is run 100 times in a complete cross-validation cycle, and the correct classification of both training and testing classification rates are recorded. Finally, six samples, two for each NN model (training/testing) with size = 100, were selected for subsequent statistical analysis.
The results of the diagnosing performances of the three NN models, in terms of average and standard deviation (SD), averaged over 100 computer runs of a complete cross-validation cycle, are displayed in Table 2.

Table 2. NN average training/testing performances over 100 runs

NN type	**Training perf. avg./SD(%)**	**Testing perf. avg./SD(%)**
MLP	97.66 / 3.06	95.88 / 3.57
RBF	96.79 / 3.30	95.29 / 3.81
PNN	87.05 / 5.18	82.51 / 5.52

From the table above we can see that:

- The training/testing performances depend on the NN type, the poorer performance being obtained by PNN and the higher by MLP. Obviously, this result and corresponding NNs hierarchy is decisively determined by this particular database.
- There is no statistical significant difference (two-sided t-test for difference between two means, with p-value > 0.05) between the training and the testing performances, regardless of the NN model, denoting the fact that all the system cross-validate well.
- The diagnosis accuracy of all the models is in accordance to the reported modern medical imaging experience, ranging from 80% to 95%.

Thus, approximately the same performance has been obtained using there intelligent systems fed with categorical data, unlike the modern and very expensive medical imaging techniques.

For the performance statistical comparison between the three NNs models we have considered the testing performance only, since it illustrates the accuracy of each model in real-world situations (generalization feature), that is for new, previously unknown patients. The t-test for independent samples, based on Student's t distribution, is the most commonly used method to evaluate the differences in means between two independent groups of observations.With independent groups of observations, we are interested in the mean difference between the two groups, focusing also on the variability between observations. Theoretically, the t-test for independent samples can be used as long as the variables are normally distributed within each group and the variances of the two groups are not significantly different. The *Kolmogorov-Smirnov & Lilliefors* test (since the mean and the standard deviation are computed from the actual data) has been used to test the normality of data, and the homogeneity of variances was tested with the *Levene*'s test. Thus, the testing performances were not normally distributed (p-level < 0.05), regardless of the NN model. However, due to the Central Limit Theorem, since the sample size is fairly large (100), the deviation from normality observed in the data does not matter much. Note that in practice, unequal variances of two independent samples are less problematic when the samples have the same size ([1]).

Remark. In case of small sample size, the *Mann-Whitney U* test is a nonparametric alternative to the t-test for independent samples. This nonparametric test requires all observations to be ranked as if they were from a single sample. The interpretation of the test is essentially identical to the interpretation of the result of a t-test for independent samples. To avoid the use of this alternative, we have considered from the beginning a number of 100 computer runs, so the hypotheses of the t-test for independent samples are fulfilled. The table below illustrates the results of the use of the t-test for independent sample:

Table 3. t-test for comparing difference in means

NN type	p**-level**
MLP vs. RBF	0.12
MLP vs. PNN	0.04
RBF vs. PNN	0.01

Using the difference between two proportions test (z-value/two-sided) to compare the average testing performances, we obtained the following results, illustrated in the table below Table 4.

Table 4. Comparing average testing performances (two-sided z-test)

NN type	**Testing performance (p-level)**
MLP vs. RBF	0.840
MLP vs. PNN	0.002
RBF vs. PNN	0.006

From the above table we can see that:

- There is no statistical significant difference (p-level > 0.05) between the average proportions of correctly classified cases for MLP and RBF (p-level > 0.05), showing that the two NN models perform similarly.
- There is statistically significant difference (p-level < 0.05) between the average proportions of correctly classified cases for PNN and both MLP and RBF. This means that in this particular case, PNN has a lower performance than MLP and RBF.

As an overall conclusion of the above statistical comparison of testing performances, the experiment shows that in this situation -breast cancer detection- MLP and RBF models performs similarly, having performances in accordance with sophisticated medical imaging techniques, which are more expensive and time consuming.

The steps of the collaborative stage concern:

- The distribution of the number of votes (weights) among the 3 NN models. By default, the number of votes is directly proportional to the testing performance;
- The choice of the quota between 50% and 100% of the total number of votes ('legitimate' WVS);
- Choice of voters power;
- Computation of the voting process output to establish the overall computing system diagnosis;

In this way, we use in a collaborative way the decision power of each member of the team, previously chosen in the competitive phase. Thus, the weighted combination of each vote optimizes the final decision, leading to a more trustworthy diagnosis, surpassing individual verdict.

Technically, the collaborative phase will use a system of weights w_i, $i = 1, 2, 3$, directly proportional to the diagnosis accuracy of the three NN models, that is $w_1 = 0.345$ for the MLP, $w_2 = 0.343$ for the RBF and $w_3 = 0.297$ for the PNN. Next, applying the WVS with the above parameters, the overall automatic diagnosis can be established for a new patient. A simulation consisting in three different cases is displayed in the table below:

Table 4. Simulation of the competitive/collaborative diagnosing system

MLP	**RBF**	**PNN**	**WVS**	**Diagnosis**
0	1	0	$0 \cdot 0.345 + 1 \cdot 0.343 + 0 \cdot 0.297 = 0.343$	0
0	1	1	$0 \cdot 0.345 + 1 \cdot 0.343 + 1 \cdot 0.297 = 0.64$	1
1	0	1	$1 \cdot 0.345 + 0 \cdot 0.343 + 1 \cdot 0.297 = 0.642$	1
0	0	1	$0 \cdot 0.345 + 0 \cdot 0.343 + 1 \cdot 0.297 = 0.297$	0

While, in the first case, MLP and PNN provided the correct diagnosis, RBF, as the second best failed. In the second case, the best competitor (the winner) failed to establish the correct result, but the overall collaboration has remedied the situation.

The third case is similar to the first one, and, in the last case, the collaboration between the first two best competitors lead to the correct diagnosis. To conclude, this example illustrates the effectiveness of a competitive/collaborative diagnosis system in comparison to separate standalone networks.

5. Conclusions

Automatic medical diagnosis, as a collaborative paradigm involving both medical knowledge and Artificial Intelligence methods, has become a very important interdisciplinary technology in health care, yielding non-invasive accurate diagnoses with low costs and high speed.
In this context, the aim of this paper was to demonstrate the suitability of the machine learning methodology, used in a both competitive and collaborative way for the diagnosis of breast cancer. The classification results were consistent with some of the highest results obtained by using sophisticated and expensive imaging medical techniques.
Future work would use more machine learning techniques in order to asses the potential of this Artificial Intelligence domain in computer-aided diagnosis.

Acknowledgement. The WPBC database was obtained from the University of Wisconsin Hospitals, Madison. Thanks go to Dr. William H. Wolberg, W. Nick Street and Olvi L. Mangasarian for providing the data.

References

[1] D.G. Altman ,*Practical statistics for medical research*, Chapman and Hall, 1991.
[2] C. Bishop, *Neural Networks for Pattern Recognition*, Oxford University Press, 2004.
[3] C.L. Chi, W.N. Street and W.H. Wolberg, Application of artificial neural network-based survival analysis on two breast cancer datasets, *AMIA Annu. Symp. Proc.* **11** (2007), 130–134.
[4] F.J. Gilbert, S.M. Astley, M.G.C. Gillan, O.F. Agbaje, M.G. Wallis, J. James, C.R.M. Boggis and S.W. Duffy, Single Reading with Computer-Aided Detection for Screening Mammography, *N. Engl. J. Med.* **359** (2008), no. 16, 1675–1684.
[5] D.B. Fogel, E.C. Wasson, E.M. Boughton and V.W. Porto, A step toward computer-assisted mammography using evolutionary programming and neural networks, *Cancer Lett.* **119** (1997), no. 1, 93–97.
[6] M. Gorunescu, F. Gorunescu and K. Revett, Investigating a Breast Cancer Dataset Using a Combined Approach: Probabilistic Neural Networks and Rough Sets,*Proc. 3rd ACM International Conference on Intelligent Computing and Information Systems -ICICIS07, Cairo, Egypt, Police Press* (2007), 246–249.
[7] F. Gorunescu, M. Gorunescu, E. El-Darzi and S. Gorunescu, A statistical evaluation of neural computing approaches to predict recurrent events in breast cancer,*Proc. 4th International IEEE Conference on "Intelligent Systems-ISO8, Varna, Bulgaria, IEEE Press* (2008), 38–43.
[8] F. Gorunescu, M. Gorunescu, E. El-Darzi and S. Gorunesc, A statistical evaluation of neural computing approaches to predict recurrent events in breast cancer,*International Journal of General Systems*,(in press), (2009).
[9] F. Gorunescu, M. Gorunescu, A. Saftoiu, P. Vilmann and S. Belciug, Competitive/Collaborative Neural Computing System for Medical Diagnosis in Pancreatic Cancer Detection,*Expert Systems*, (in press), (2009).
[10] S. Haykin, *Neural Networks, a comprehensive foundation (2nd Edition)*, Prentice Hall, 1999.
[11] Y.H. Hsiao, Y.L. Huang, W.M. Liang, S.J. Kuo and D.R. Chen, Characterization of benign and malignant solid breast masses: harmonic versus nonharmonic 3D power Doppler imaging, Ultrasound *Med. Biol.* **35** (2009), no. 3, 353–359.
[12] E.D. Ibeyli, Implementing automated diagnostic systems for breast cancer detection,*Expert Systems with Applications: An International Journal* **33** (2007), no. 4, 1054–1062.

[13] R. Kohavi, A study of cross-validation and bootstrap for accuracy estimation and model selection, *Fourteenth International Joint Conference on Artificial Intelligence* **2** (1995), no. 12, 1137–1143.

[14] J.H. Lee, E.L. Rosen and D.A. Mankoff, The Role of Radiotracer Imaging in the Diagnosis and Management of Patients with Breast Cancer: Part 1-Overview, Detection, and Staging*, *Journal of Nuclear Medicine* **50** (2009), no. 4, 569–581.

[15] C.E. McLaren, W.P. Chen, K. Nie and M.Y. Su, Prediction of malignant breast lesions from MRI features: a comparison of artificial neural network and logistic regression techniques, *Acad. Radiol.* **16** (2009), no. 7, 842–851.

[16] M.P. Perrone, General averanging results for convex optimization. In M.C. Mozer et al. (Eds.), *Proceedings 1993 Connectionist Models Summer School* (1994), 364–371.

[17] M.P. Perrone and L.N. Cooper, When networks disagree: ensemble methods for hybrid neural networks. In *Artificial Neural Networks for Speech and Vision*, Chapman & Hall, London (1993), 126–142.

[18] K. Revett, F. Gorunescu, M. Gorunescu, E. El-Darzi and M. Ene, A Hybrid Breast Cancer Diagnosis System: A Combined Approach Using Rough Sets and Probabilistic Neural Networks, *Proceedings of the IEEE International Conference on Computer as a tool-Eurocon 2005*, Belgrade, Serbia, IEEE Press, (2005), 1124–1127.

[19] D.F. Specht, *Probabilistic neural networks, Neural Networks*, (1990).

[20] D. Vranjesevic, J.E. Filmont, J. Meta, D.H. Silverman, M.E. Phelps, J. Rao, P.E. Valk and J. Czernin, Whole-Body 18F-FDG PET and Conventional Imaging for Predicting Outcome in Previously Treated Breast Cancer Patients, *Journal of Nuclear Medicine* **43** (2002), no. 3, 325–329.

[21] P. Tannenbaum, *Excursions in Modern Mathematics (6 Ed.)*, Prentice Hall, 2006.

(Smaranda Belciug) Faculty of Mathematics and Computer Science, University of Craiova, Al.I. Cuza Street, No. 13, Craiova RO-200585, Romania, Tel. & Fax: 40-251412673
E-mail address: smaranda.belciug@inf.ucv.ro