Comparison of Computational Learning Methods on a Diagnostic Cytological Application

K. Koutroumbas*, G. Paliouras♣, V. Karkaletsis♣, and C.D. Spyropoulos♣

*Institute for Space Applications and Remote Sensing, National Observatory of Athens, Metaxa and V.Pavlou, Palaia Penteli, 15236 Athens, Greece
koutroumbas@space.noa.gr

♣ Software & Knowledge Engineering Laboratory, Institute of Informatics and Telecommunications, NCSR "Demokritos", 153 10 Ag. Paraskevi, Athens, Greece
{paliourg, vangelis, costass}@iit.demokritos.gr

ABSTRACT: In this paper we perform a comparative evaluation of four different computational learning methods on a problem of diagnostic cytology and more specifically on the classification of gastric cells. The methods considered are: Decision Tree Induction, Boosted Decision Trees, Naive Bayesian Classifier, and Radial Basis Function Neural Networks. The performance of each method was assessed on unseen data. Our aim was not to evaluate the quality of the algorithms as such, but to examine which of them are suitable for the specific medical diagnosis task, in order to provide a reliable diagnostic tool to the doctors involved in the area. We compare the performance of the four methods and discuss the results taking into account the characteristics of the methods and the task examined. The dataset that was used in this paper is publicly available, facilitating reproducibility of the results and providing a basis of comparison for future work.

KEYWORDS: machine learning, medical diagnosis, decision trees, neural networks

1. INTRODUCTION

Cytology has been proved a powerful tool for the diagnosis of gastric lesions. In recent years, several machine learning methods have been applied to diagnostic cytology, e.g. [1],[2],[3],[6],[7],[8],[11],[14], adding significant value to the diagnostic process. A notable benefit from the use of these methods is that their response is always objective, since it is based on a specific set of measurements taken for each cell. In addition, machine learning methods guarantee the reproducibility of the results. That is, each cell will always be assigned to the same class, no matter how many times it is presented to the method. This is not always the case for the cytologists, especially when the cells are difficult to identify.

The aim of this paper is to investigate the performance of four machine learning methods, namely, *Decision Tree Induction, Boosted Decision Trees, Naive Bayesian Classifier, and Radial Basis Function Neural Networks* in two classification problems: (a) assignment of gastric cells to one of two categories (benign, malignant), and (b) assignment of gastric cells to one of five categories (ulcer, gastritis, inflammatory displasia, true displasia, cancer). The dataset that was used in this paper is publicly available, facilitating reproducibility of the results and providing a basis of comparison for future work.[1]

The rest of this paper is organized as follows. In section 2 a brief description of the four machine learning methods is given. In section 3, the cytological application is presented, as well as the classification results of

[1] The data set may be found in the address http://www.di.uoa.gr/~makis/projects/CCS/CCS.html

the four methods. Finally, section 4 presents our concluding remarks and the most promising paths for future work.

2. DESCRIPTION OF THE METHODS

The methods used here belong in the family of supervised learning methods, which are suitable ones for classification and prediction tasks. Supervised learning methods require a dataset *S* of the form

$$S = \{(\mathbf{x}_i, c_i)\},$$

where $\mathbf{x}_i$ is a vector of characteristic features x_{ij}, describing an instance of class c_i. This set of preclassified instances is used to learn the classification model and is therefore called the *training set*. A similar dataset, called the *test set*, is used for evaluating the performance of the methods. The classification model assigns each feature vector $\mathbf{x}_i$ of the test set to a class c_i', which is then compared to the true class c_i of the instance. If $c_i' = c_i$ the instance is considered correctly classified, otherwise it counts as a misclassification.

Decision tree induction (J48). The decision tree induction algorithm used here is called J48 and it is an improved version of the C4.5 algorithm [10], as implemented in the WEKA machine learning platform [16]. J48 constructs a decision tree, which can then be used to classify unseen data. J48 generates decision trees, the nodes of which evaluate the existence or significance of individual features x_i in each vector $\mathbf{x}_i$, e.g. a gastric cell. Following a path from the root to the leaves of the tree a sequence of such tests is performed, resulting in a decision about the appropriate class of the gastric cell. The decision trees are constructed in a top-down fashion, by choosing the most appropriate attribute each time. The attributes are evaluated according to an information-theoretic criterion, which provides an indication of the "classification power" of each attribute. This criterion uses information entropy

$$Entropy(S) = \sum_{j=1}^{r} - p_j \log_2 p_j$$

to measure the "quantity" of information required to describe the classification of the items in a dataset *S*, into one of *r* classes, distributed according to probability p_j. The criterion is called *information gain* and measures the reduction of the entropy, when splitting the dataset according to a characteristic attribute *A*:

$$Gain(S, A) = Entropy(S) - \sum_{u \in Values(A)} \frac{|S_u|}{|S|} Entropy(S_u),$$

where *Values*(*A*) is the set of values that *A* can take and S_u the set of vectors for which *A* takes the value *u*. Thus, the attribute providing the highest information gain is selected as the "best" discriminator and added to the decision tree. Once an attribute is chosen, the training data are divided into subsets, corresponding to different values of the selected attribute. This process is repeated for each subset, until a large proportion of the instances in each subset belong in a single class.

Boosted decision trees. Boosting is a technique for improving the performance of machine learning algorithms by constructing various classification or prediction models and combining them through a voting process. The various models are generated by the same learning algorithm and each new model aims to learn the weaknesses of the previously generated ones. We used the AdaBoost method [5], which is designed specifically for classification methods, and we have combined it with decision-tree classifiers. This method attempts to diversify the generated classifiers, by training each new classifier on examples that previous classifiers have found difficult to classify. Specifically, training with AdaBoost proceeds as follows: (a) all training instances are uniformly assigned a weight, (b) a new model (e.g. a decision tree) is learned and the weights of the correctly classified instances are multiplied by $e/(1-e)$, where e is the overall error of the model and it should be less than 0.5, otherwise the model is rejected and the process stops, (c) the generation of the next model takes into account the weights of the training instances, which are lower for the correctly classified instances, due to their multiplication by $e/(1-e)$, taking values in the range [0..1) when $e<0.5$. Using this method, a sequence of decision trees are generated and each tree contributes to the

final classification of each instance, according to its performance. After all classifiers have been constructed, a weight ***w*** defined as

$$w = -\log\frac{e}{1-e},$$

is assigned to each one of them. The weight ***w*** takes smaller values as *e* decreases. The final decision is taken by a weighted voting function.

Naive Bayesian classification. The Naive Bayesian classifier [4] is arguably one of the simplest probabilistic classifiers, following from Bayesian decision theory. According to this theory, the most probable class for a vector $\mathbf{x} = (x_1,\ldots, x_n)$ is the class with the maximum aposteriori probability:

$$\arg\max_{j=1,\ldots,r} P(c_j \mid x_1,\ldots, x_n),$$

which becomes:

$$\arg\max_{j=1,\ldots,r} \frac{P(x_1,\ldots, x_n \mid c_j)P(c_j)}{P(x_1,\ldots, x_n)},$$

with the use of the Bayes theorem. Therefore, the vector **x** is assigned to the class that maximises $P(x_1,\ldots, x_n|c_j)P(c_j)$, since $P(x_1,\ldots, x_n)$ is independent of the class assigned.
The apriori probabilities of the classes $P(c_j)$ are estimated by their frequencies in the training set. However, the probabilities $P(x_1,\ldots, x_n|c_j)$ are more difficult to estimate correctly. At this point the simplifying assumption that the features x_i are stochastically independent is made. Under this assumption $P(x_1,\ldots, x_n|c_j)$ is approximated by:

$$P(x_1,\ldots, x_n \mid c_j) = \prod_i P(x_i \mid c_j),$$

The probability $P(x_i|c_j)$ is estimated by the frequency of each feature in the training instances of class c_j[2]. The independence assumption is the reason why the algorithm is called Naive, as the assumption rarely holds in real-world problems. Despite this strong assumption, the algorithm performs surprisingly well on a range of tasks.

Radial Basis Function (RBF) Networks. RBF networks are feed-forward neural networks consisting of two layers of nodes (in addition to the input layer). Each node n of the first (hidden) layer is usually described by the function

$$y_n = \exp(-\|\mathbf{w}_n - \mathbf{x}\|^2/2\sigma_n^2),$$

where **x** is the input vector, $\mathbf{w}_n$ is the parameter vector of the node and σ_n^2 is a parameter related to the variance around $\mathbf{w}_n$. Intuitively, the output of the node decreases as **x** moves away from $\mathbf{w}_n$. Each node *m* of the second (output) layer is characterized by the function

$$z_m = \mathbf{v}_m^{\mathrm{T}}\mathbf{y} + v_{m0},$$

where **y** is the vector of the outputs of the first layer nodes, $\mathbf{v}_m$ the parameter (weight) vector of the node, and v_{m0} is the so called *bias* of the node. When RBF networks are used for classification, the number of the output nodes is determined by the number of classes involved. For the two-class case, a single output node is used and a vector is assigned to one or the other class, e.g. benign or malignant, depending on whether the output of the node is greater or less than a prespecified threshold. In the general case of *r* classes, the network has *r* nodes. The node with the maximum output indicates the class to which the vector is assigned.
The training of such a network is usually carried out in two stages. During the first stage, the parameters $\mathbf{w}_n$ and σ_n^2 of the first (hidden) layer nodes are determined using some clustering method. In the second stage, the parameters $\mathbf{v}_m$ and v_{m0} of each second-layer (output) node are determined using the Least Mean Square (LMS) algorithm, see for example [13],[15]. This strategy is followed also in the present paper.

[2] In the case where x_i's take continuous values, $P(x_i|c_j)$'s become point density functions. Their estimation may be carried out using various methods that can be found in the literature (see e.g. [4], [13]).

The clustering algorithm used here is a variation of the BSAS algorithm discussed in [13] that takes into account the true class c_j of each vector $\mathbf{x}_i \in S$. Each cluster produced by the algorithm contains vectors of a single class. The algorithm is fully described in [9] and is briefly outlined here. The algorithm requires one pass through the dataset and it goes as follows: The first vector $\mathbf{x}_1$ is assigned to the first cluster C_1. Then the second vector $\mathbf{x}_2$ is considered. Its distance from C_1 is computed and if it is less than a user-defined parameter Θ, which is a measure of dissimilarity, and $\mathbf{x}_2$ belongs in the same class as $\mathbf{x}_1$, then $\mathbf{x}_2$ is also assigned to C_1, otherwise a new cluster C_2 is created for $\mathbf{x}_2$. Similarly, for any other pattern $\mathbf{x}_i$ its distance from the mean vectors of existing clusters is considered.[3] If the minimum of these distances is less than Θ and $\mathbf{x}_i$ belongs in the same class as all vectors of the corresponding cluster, then $\mathbf{x}_i$ is assigned to this cluster. Otherwise a new cluster is created containing only $\mathbf{x}_i$. The algorithm terminates when all vectors have been considered once.

For each cluster C_i formed by the algorithm, an RBF node is created in the first (hidden) layer of the RBF network. Its parameter vector $\mathbf{w}_n$ is the mean of the vectors that belong in C_i. The parameter σ_n^2 is defined as the variance of the vectors of C_i around the mean vector. After the construction of the first layer, we determine for each $\mathbf{x}_i \in S$ the output $\mathbf{y}_i$ of the first layer and we form the set

$$S' = \{ (\mathbf{y}_i, c_j) \}.$$

The training of the second layer nodes is based on S' and uses the LMS algorithm.

3. EXPERIMENTAL RESULTS

3.1 EXPERIMENTAL SETTING

The aim of this application is the assignment of a gastric cell to one of the following classes: *ulcer, gastritis, inflammatory displasia, true displasia, cancer*. The cells were obtained from brushing cytology smears taken from patients during endoscopy. Each smear contained about 100 cells. For each cell the measurements of 26 characteristic attributes were taken, forming a 26-th dimensional vector that characterizes the corresponding cell. The 26 attributes are divided into two types, as follows:

1. *Geometric characteristics*: area, circularity, major axis, minor axis, perimeter, formAR, formPE, NCI, contour ratio, roundness factor, diameter and nuclei mean radius.
2. *Textural characteristics:*
   a. *Nuclei run length*: short run, long run, grey level, distribution.
   b. *Nuclei histogram*: mean, variance and standard deviation.
   c. *Nuclei coocurence matrix*: maximum, entropy and inertia.
   d. *Differences histogram*: mean, variance, contrast and entropy.

The whole data set consists of 13300 vectors (cells) extracted from 120 patients. 2920 of these belong in the class *"cancer" (C)*, 370 in the class *"true displasia" (TD)*, 6550 in the class *"ulcer" (U)*, 3150 in the class *"gastritis" (G)* and 310 in the class *"inflammatory displasia" (ID)*. The identification of the class of each cell was made by two experienced cytologists and confirmed by the histological examination of biopsies and/or surgical specimens. The available data was split into a training and a test set, at a 30-70 proportion, using stratified random selection, and thus preserving the original class distribution in the two datasets.[4] The following two sections present the results that were obtained in the two tasks, i.e., classifying cells into two classes (malignant and benign) and in the five original classes.

3.2 TWO-CLASS CASE

First, we consider the two-class problem, in which the cells of the classes *C* and *TD* are characterized as *"malignant"* and the cells of the remaining three classes, *G, U, ID*, as *"benign"*. The results of the four methods described above are shown in table 1.

[3] Specifically, the distance of $\mathbf{x}_i$ from the mean vectors of the clusters is considered.

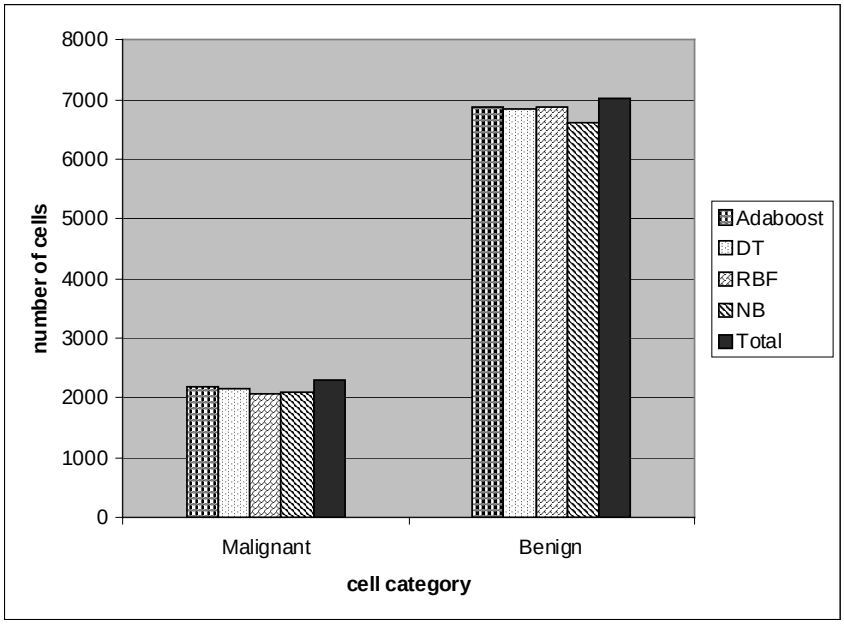
[4] For a thorough description of the data set se [7].

	Training set accuracy	Test set accuracy
AdaBoost	100%	97.25%
Decision Trees	98.45%	96.61%
RBF network	~98%	96.24%[5]
Naive Bayesian Classifier	94.44%	94.17%

Table 1. The results of the four methods for the two-class case.

The results presented in table 1, show that all methods discriminate very well among the two cell classes. This is an indication that there are clear distinctions between the two classes, in terms of the 26 attributes that were used. In particular, the AdaBoost method gives slightly better results than the other three methods, while the Naive Bayesian classifier has the lowest performance. The performance of all four methods on the training set is similar to that on the test set, indicating that there is no overfitting on the training data. The fact that AdaBoost achieves 100% accuracy on the training set is a characteristic of the method and not an indication of overfitting, which is proven by the high accuracy on the test set. AdaBoost is able to model the training data exactly and still achieve high accuracy, by assigning the appropriate weights to the various classifiers that it constructs. As a means of comparison to previous work, a multi-layered perceptron has been trained with backpropagation in [7], using a subset of the dataset that we used here (~11000 cells) and the same stratified 30-70 division between training and test data. The results of that method were similar to the ones obtained here (95.7% - 97.3% accuracy on the test set).

As in all medical diagnosis tasks, the types of misclassification made by different methods are particularly interesting. We illustrate this by means of the relative performance of the classifiers in the two different classes (figure 1), as well as the corresponding confusion matrices (table 2). The presented results are on the test set.



[5] These results were obtained by ignoring the clusters with less than three elements produced during the training of the first layer of the network.

Figure 1. Relative performance of the four methods on the two classes (*malignant* and *benign*). The darkest shaded columns show the true number of instances of each class.

	Confusion matrix		
AdaBoost		Malignant	Benign
	Malignant	2181	122
	Benign	134	6873
Decision Trees		Malignant	Benign
	Malignant	2157	146
	Benign	170	6837
RBF network		Malignant	Benign
	Malignant	2079	224
	Benign	126	6881
Naive Bayesian Classifier		Malignant	Benign
	Malignant	2107	136
	Benign	407	6600

Table 2. The confusion matrices produced by the four methods for the two-class case. The (1,2) element of the above confusion matrices corresponds to the number of cells that belong in the class *"malignant"* and have been identified as *"benign"*. The opposite holds for the (2,1) element. For example, in the matrix of AdaBoost 134 benign cells have been classified as malignant and 122 malignant cells have been classified as benign.

Reading figure 1, it is clear that AdaBoost does equally well in both classes. On the other hand, the RBF network performs slightly better in the largest class (benign), while the decision tree performs better than the RBF network in the smaller class (malignant). This can be attributed to the attribute selection function of J48, which treats all classes as equally important. Focusing on table 2 one interesting observation is that *different methods capture different issues of the problem and miss others.* For example the Naive Bayesian classifier misclassifies several benign cells (about three times more from the other methods) as malignant. Also, the RBF network misclassifies several malignant cells (about 1.5 to 2 times more than the other methods) as benign.

3.3 FIVE-CLASS CASE

Here we consider the original five-class problem. The results of the four methods in this problem are shown in table 3.

	Accuracy on the training set (%)	**Accuracy on the test set (%)**
AdaBoost	100%	80.37%
Decision Trees	91.00%	78.59%
RBF network	N/A	77.41%
Naive Bayesian Classifier	68.77%	68.74%

Table 3. The results of the four methods for the five-class case.

In this case, the results are worse that in the two-class case. This is an indication that the five-class problem is harder than the two-class one. That is the classes are not well separated from each other. This is supported by the nature of the application, since the *TD* (*True Displasia*) class is very close to the *C* (*Cancer*) class and *ID* (*Inflammatory Displasia*) class is very close to the classes *U* (*Ulcer*) and *G* (*Gastritis*). The AdaBoost method gives once more the highest success rates of the four methods examined here. The Desicion Trees and the RBF network exhibits lower success rates but comparable to the AdaBoost method, while the Naive Bayes classifier gives much lower success rate. This can be explained

by the fact that the attribute independence hypothesis, on which the Naive Bayesian classifier is based, is not valid in the five class case.

In order to confirm our intuition about the similarity between some of the classes, we examine again the types of error made by the four methods. Figure 2 presents the relative performance of the classifiers in the five different classes, while table 4 presents the corresponding confusion matrices. The presented results are on the test set.

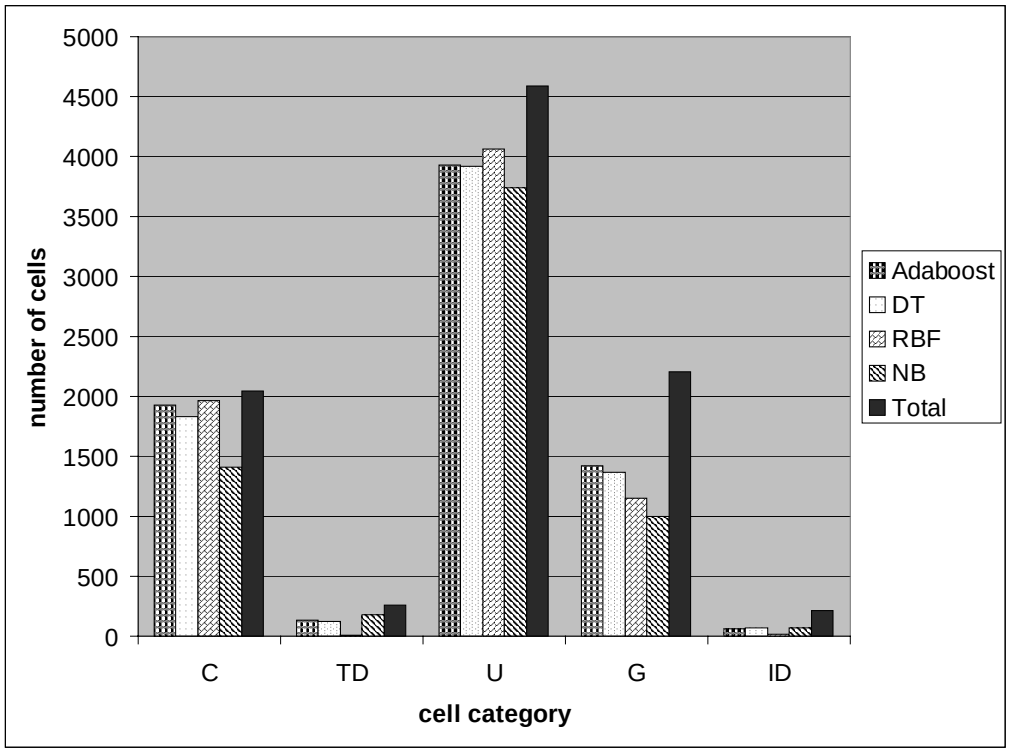


Figure 2. Relative performance of the four methods on the five classes (*TD*, *C*, *ID*, *U*, *G*). The darkest shaded columns show the true number of instances of each class.

The first interesting observation in figure 2 is the very unbalanced distribution of instances in the five classes. The *TD* and *ID* the classes are very small, i.e., they contain few instances, in contrast to the large *U* class. The RBF networks seem to be doing much worse in the smaller classes than in the larger ones. This may be explained either by the unbalanced distribution of the classes or by the close similarity of the two smaller classes with larger classes. Examining the confusion matrices for the RBF network it becomes clear that the larger classes are favoured. For instance, 52% of the *ID* cases are misclassified to class *U* and 36% to class *C*. Similarly, 63% of the *TD* cases are misclassified to class *C*. In contrast to the RBF networks, AdaBoost and the Decision Trees seem to provide better models of the smaller classes. Perhaps the most difficult class for all methods is the *G* class, which is larger than the *C* class, but apparently very difficult to identify correctly. The reason seems to be that the characteristics of the *G* class are similar to the largest class (*U*), as can be seen by the high level of misclassification between the two classes in the confusion matrices.

Another interesting observation in figure 2 and table 4 is again the fact that different classifiers focus on different aspects of the problem, i.e., their misclassifications differ. Thus, the major issue in obtaining even better classification results seems to be the efficient combination of the results of different classifiers for a given problem rather than the implementation of the "best" single classifier.

	Confusion matrix					
AdaBoost		**C**	**TD**	**U**	**G**	**ID**
	C	1931	35	33	22	23
	TD	92	138	2	26	1
	U	46	6	3933	564	36
	G	11	26	738	1420	10
	ID	39	5	99	14	60
Decision Trees		**C**	**TD**	**U**	**G**	**ID**
	C	1831	64	69	19	61
	TD	96	121	7	30	5
	U	54	15	3924	530	62
	G	16	37	760	1366	26
	ID	41	3	86	12	75
RBF network		**C**	**TD**	**U**	**G**	**ID**
	C	1967	4	27	36	10
	TD	163	11	6	73	6
	U	112	0	4065	406	2
	G	72	2	980	1150	1
	ID	78	0	112	13	14
Naive Bayesian Classifier		**C**	**TD**	**U**	**G**	**ID**
	C	1415	362	14	22	231
	TD	47	181	2	18	11
	U	41	34	3739	553	218
	G	28	94	1028	998	57
	ID	6	19	112	13	67

Table 4. The confusion matrices produced by the four methods for the five-class case. The (1,2) element of the above confusion matrices corresponds to the number of cells that belong in the class *C* and have been identified as *TD*. The opposite holds for the (2,1) element. For example, in the matrix of AdaBoost 92 cells of class *TD* have been classified to class *C* and 35 cells of class *C* have been classified to class *TD*. The other elements of the confusion matrices are defined similarly.

4. CONCLUSIONS

In this paper, we evaluated four different machine learning methods on the task of diagnosing gastric lesions, based on cytological evidence. The four methods were evaluated on two diagnostic problems of different level of difficulty: the easier task of distinguishing between benign and malignant cells and the harder task of assigning a cell to one of five diagnostic categories (ulcer, gastritis, inflammatory displasia, true displasia, cancer). The experiments were done with the use of a public domain corpus, facilitating reproducibility of the results and providing a basis of comparison for future work.

The first interesting observation is the very high accuracy of all methods on the two-class case. In particular the AdaBoost method achieved 97.25% accuracy on unseen data, thus providing a very good model of the diagnostic process and a valuable tool for the cytologist. In the harder task, the performance of all methods was reduced, but AdaBoost remained the favourable option, achieving 80.37% accuracy on unseen data. However, the difference in the performance of the three best methods, i.e., AdaBoost, Decision Trees and RBF networks, was small and perhaps statistically insignificant. As a result, it is within our immediate future plans to repeat our experiments, using stratified cross-validation, in order to obtain an unbiased estimate of the performance of the methods and measure the statistical significance of their differences.

However, the use of random selection in the cross-validation experiments is likely to make the results more difficult to reproduce.

Another interesting conclusion from the work presented in this paper is that the various classifiers tend to make different types of error. This result suggests that higher performance can be achieved by combining different classifiers, e.g. using a stacked generalization approach [17]. The high performance of the AdaBoost method, which is a form of combination of different classifiers, further supports this intuition. Finally, we are interested in evaluating different classifiers on the same data, such as proportional-odds logistic regression, which has proven quite effective in other medical diagnosis problems with ordered classes [12]. The ultimate goal of this work is to provide the cytologist with a reliable tool facilitating accurate diagnosis.

ACKNOWLEDGEMENTS: The authors would like to thank: (a) the team of the project VALUE-CCS, partially funded by ECC (Contract number CTT-471), for the kind provision of the data, (b) Mr A. Grigoriadis for the assistance in the experiments with AdaBoost, the Decision Trees and the Naïve Bayesian classifier, (c) Mr A. Agathos and Mr S. Pantazis for the software implementation of the RBF networks.

REFERENCES

[1] Astion M. L., Wilding P. "The Application of Back Propagation Neural Networks to Problems in Pathology and Laboratory medicine" Arch. Pathol. Lab. Med. 116:995:1001. 1992.

[2] Astion M. L., Wilding P. "Application of Neural Networks to the Interpretation of Laboratory Data in Cancer Diagnosis" Clin. Chem. 38(2):34:38. 1992.

[3] Dawson A. E. et al. " Nuclear Grading of Breast Carcinoma by Image Analysis". A.J.C.P. 95(4) S29:S37. 1991.

[4] Duda, R. O. and Hart, P. E.: Pattern Classification and Scene Analysis, John Wiley, (1973)

[5] Freund, Y. and Schapire, R. E.: Experiments with a new boosting algorithm. In Proceedings of the International Conference on Machine Learning, 1996.

[6] Haykin Simon. "Neural networks a comprehensive foundation". Macmillan College Publishing Company, Inc. ISBN 0-02-352761-7. 1994.

[7] Karakitsos P., Botsoli-Stergiou E., Pouliakis A., Tzivras M., Archimandritis A., Liossi A., Kyrkou K., "Potential of the Back Propagation Neural Network in the Discrimination of Benign from Malignant Gastric Cells", Analytical Quantitative Cytology and Histololy 1996, 18:3: 245-250.

[8] Karakitsos P., Pouliakis A., Koutroumbas K., Botsoli-Stergiou E., Tzivras M., Archimandritis A., Liossi A., "Neural Network Application in the Discrimination of Benign from Malignant Gastric Cells", Analytical and Quantitative Cytology and Histololy 2000, Vol. 22, No. 1, 63-69.

[9] Koutsoulis N., Kapogiannopoulos G., Koutroumbas K., Kalouptsidis N., “A new Greek Optical Character Recognition System”, under review.

[10] Quinlan, J. R.: C4.5: Programs for machine learning, Morgan - Kaufmann (1993).

[11] Rosenthal D. L., Mango L. J. “Applications of Neural Networks for Interactive Diagnosis of Anatomic Pathology Specimens” pp. 173-184. In Compedium on the Computerized Cytology and Histology Laboratory ed. Wied G. L. et al. ToC, Chicago, Illinois USA 1994.

[12] S.M. Rudolfer, G. Paliouras and I. Peers, "A Comparison of Logistic Regression to Decision Tree Induction in the Diagnosis of Carpal Tunnel Syndrome," Computers and Biomedical Research, v. 32, pp. 391-414, 1999.

[13] Theodoridis S., Koutroumbas K., Pattern Recognition, Academic Press, 1998.

[14] Truong H. et al. "Neural Networks as an Aid in the Diagnosis of Lymphocyte-Rich Effusions. Anal. Quant. Cytol. And Histol. 17(1):48:54. 1995.

[15] Widrow B., Lehr M.A., "30 years of Adaptive Neural Networks: Perceptron, Madaline and Backpropagation", proc. of IEEE, vol. 78. No. 9, Sept. 1990.

[16] Witten I., Frank E., Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations, Morgan Kaufmann, 1999.

[17] Wolpert, D. 1992. “Stacked generalization.” Neural Networks, 5 (2), 241-260.