Comparative evaluation of Multi Layer Perceptrons, to hybrid Multi Layer Percetrons, with Multicriteria Hierarchical Discrimination and Logistic Regression in Corporate Financial Analysis

LOUKERIS N.
C.C.F.E.A.
University of Essex
Wivenhoe park, CO4 3SQ, UK or Ag. Ioannou Xenou 9, 73100 Chania, Crete, Greece
United Kingdom

Abstract:- The net present value of corporations is reflected in the stock price, leading portfolio mangers to maximize their assets, securing stockholders interests. Extended accounting data produced by accounting reports, and financial markets include lucrative hidden information. Econometrics, Neural Networks and Multicriteria Analysis classify companies describing their economic robustness. In a detailed comparison Multi Layer Perceptron classification is compared to neuro-genetic hybrid of Multi Layer Perceptron, Logistic Regressions and Multigroup Hierarchical Discrimination classifications to determine efficient methods in Financial Analysis. Simple Logistic and Logistic Model Trees methods had a fine performance.

Keywords:- MultiLayer Perceptron, Genetic Algorithms, Logistic Regressions, Multigroup Hierarchical Discrimination, Finance

1 Introduction

Portfolio administrators face a constant dilemma on the distribution of their investors' capitals to an optimal number of stocks. The efficiency of corporate governance is partially quantified in accounting statements, financial indices and auditor reports, offering significant, but usually hidden, information on the economic wealth each company owns. Artificial Intelligence provided valuable results in accounting data and financial indices, **[1]**. Neural Networks, Hybrid Systems as Neuro-Genetic Networks, O.R. in the form of Multicriteria Analysis, and Logistic Regressions are strong techniques to provide solutions of corporate financial classifications.

2 Multi-Layer Perceptrons

Multi-Layer Perceptron–MLP is a widely used neural network, **[2]**, where input signals are computed in a number of layers that contain artificial neurons. These neurons are the Processing Elaments-PEs of the network. The number of PEs in input layer is identical to the set of variables and at the output layer PEs equals to the desired number of classes, whilst a number of sets of nonlinear PEs constitute the hidden layers. PEs, except those of input layer, produces a linear combination of the outputs given by previous layers plus a bias. The synaptic weights, between different PEs, when are normalized with the output classes to 0-1 the MLP achieve the optimal performance of maximum ex-post receiver in classifications, **[3]**. Next neurons in the hidden layer process a non-linear sigmoid function $\varphi(x) = 1/(1+exp(-x)))$ of their input. The output neurons produce a result equal to the linear combination. Multi-Layer Perceptrons with 1 hidden layer process a linear combination of sigmoid functions of the inputs. A linear sigmoid function can approximate any continuous function of 1 or more variables obtaining a continuous function fitting a finite set of points when no underlying model is available, and when the approximated function is trained with a desired answer 1 for signal and 0 for background, it has the probability of signal knowing the input values, allowing classifications. MLP can approximate arbitrary functions, whilst they are trained with the backpropagation algorithm **[4]** whilst LMS learning algorithm, **[5]** can not be extended to hidden PEs. Backpropagation begins with an initial value for each weight, and proceed until a stopping criterion is met, such as: i) to cap the number of iterations, ii) to threshold the output mean square error, or iii) to use cross validation which is the most powerful since it stops the training at the point of best generalization. Errors in network are minimized through backpropagation rule, permitting adaptation of the hidden PEs. Multi Layer Perceptron with nonlinear PEs have a smooth nonlinearity as the logistic function and the hyperbolic tangent, whilst their massive interconnectivity permits the computation non linear functions. The Multi Layer Perceptron is trained with error correction learning, where the desired response for the system must be known **[6]**. In case of implementing point estimates the problem of overfitting may appear supported by the flexibility of MLP network. To alleviate overfitting, many of the signal transformation (ST) approaches need to restrict the structure of the MLP network. Training is implemented either by presenting a pattern and adapting the weights on-line, or by presenting all the patterns in the input file (an epoch), accumulate the weight updates, and then update the weights with the average weight update-batch learning. Learning is controlled in any iterative training procedure, as the learning curve. A bias affects biological neurons in extreme weather conditions or in physiological disorders, thus a bias input is given to each one of the artificial neurons, figure 1. The ouput is discrete in {0,1}, whilst the output c of each neuron is

$$c = \varphi\ (\sum_i w_i\, \alpha_i + b)$$

where a_i the inputs of the neuron and w_i the weights of the neuron. The nonlinear activation function φ commands the activation level of the neuron. In the WEKA platform, Multi-Layer Perceptron Function was elaborated with the training set at 50% of overall data. Total data were 16 financial indices of 1411 Greek companies with a 17^{th} classification index by loan executives.

3 Hybrid Multi- Layer Perceptron with Genetic Algorithms optimization

Multi-Layer Perceptrons-MLP were at a hybrid form optimized by Genetic Algorithms were elaborated by NeuroSolutions software. Genetic Algorithms select the significant inputs by the 16 financial inputs in the hybrid MLP. The network, through multiple training, finds the inputs combination with the lowest error. Genetic Algorithms were used on each layer in MLP with different topologies. On-Line learning were chosen to update the weights of hybrid neuro-genetic MLP, after the presentation of each exemplar. Genetic Algorithms optimized the sub-problems in the: a) number of Processing Elements, b) Step Size, and c) Momentum Rate. Output layer optimized the value of Step size and Momentum through Genetic Algorithms.

4 MHDIS

Multigroup Hierarchical DIScrimination (MHDIS) sorts and classifies problems, such as country risk, bankruptcies, and credit scoring. MHDIS is a non-parametric approach for developing classification models through a hierarchical procedure **[7]**. The MHDIS method implements the preference disaggregation paradigm of MCDA to develop a classification model in parallel with mathematical

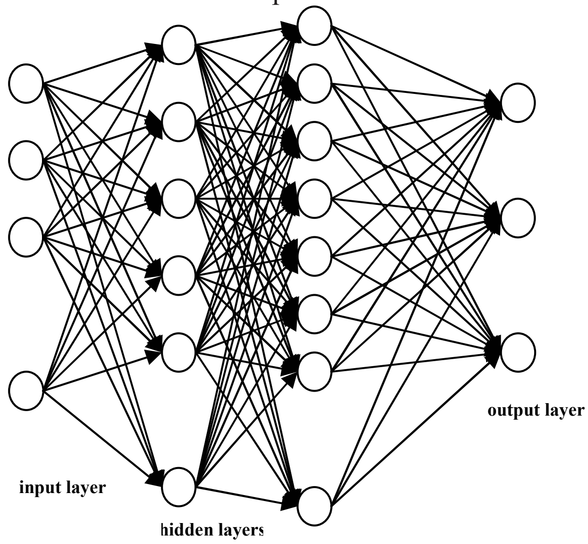

Figure 1. Multi-Layer Perceptron biased with n hidden nonlinear layers

programming; it deploys models to classify a set of alternatives into groups predefined in an ordinal way. The model development process is based on a set of reference alternatives A={α_1, α_2,. . α_n} described over a set of m criteria g ={g_1, g_2, . . . g_m}. Each alternative can be described in terms of the vector g_j ={g_{j1}, g_{j2}, . . . g_{jm}} where g_{ji} denotes the performance of alternative α_j on criterion g_i. The reference alternatives are classified into q groups defined in an ordinal way, such that $C_1 > C_2 >\_ .. > C_q$. The ordinal definition of the groups implies that the alternatives of group C_1 are preferred to the alternatives of C_2, and so on. The development of the appropriate classification model is performed progressively through a sequential hierarchical procedure. The number of stages in this hierarchical discrimination procedure is q - 1. At each stage k of this process, the discrimination of groups is based on the development of a pair of additive utility functions as:

$$U_k(g) = \sum_{i=1}^{m} p_{ki}\, u_{ki}(g_i) \quad and \quad U_{\sim k}(g) = \sum_{i=1}^{m} p_{\sim ki}\, u_{\sim ki}(g_i)$$

where the global utility function $U_k(g)$ describes the alternatives of group C_k, the latter function $U_{\sim k}(g)$ describes the alternatives belonging into the groups C_{k+1}, C_{k+2}, . . . ,C_q. The corresponding marginal (partial) utility functions $u_{ki}(g_i)$ and $u_{\sim ki}(g_i)$ are normalized between 0 and 1. The sum of the weighting parameters p_{ki} and $p_{\sim ki}$ equals to 1, they represent the contribution of each criterion in the corresponding utility function. The classification of the alternatives on the basis of these utility functions is performed as:

$U_k(g_j) > U_{\sim k}(g_j) \Rightarrow \alpha_j \in C_k$

$U_k(g_j) < U_{\sim k}(g_j) \Rightarrow \alpha_j \in \{C_{k+1}, C_{k+2}, \ldots, C_q\}$

Given that the global utility of an alternative according to the utility function $U_k(g_j)$ is higher than the global utility estimated according to the utility function $U_{\sim k}(g_j)$, then the alternative is classified into group C_k. The estimation of additive utility functions in MHDIS involves the estimation of criteria weights and the specification of the form that the criteria marginal utility functions have, and it is fulfilled

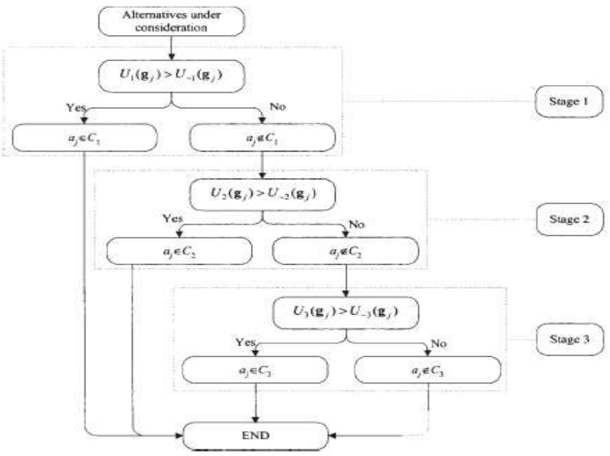

Figure 2. MHDIS algorithm

through mathematical programming. At each stage k of the hierarchical discrimination process of Figure 2, two linear programs and a mixed-integer one are solved to estimate optimally the two additive utility functions $U_k(g_j)$ and $U_{\sim k}(g_j)$, in terms of the total number of misclassifications and the clarity of the obtained classification. The first linear program (LP1) serves as an exploratory stage, and its objective is to develop an initial pair of additive utility functions $U_k(g_j)$ and $U_{\sim k}(g_j)$ minimizing the sum of magnitude of all violations in the classification rules for the reference alternatives. LP1 develops an initial pair of utility functions for the classification of the alternatives and the identification of the alternatives which are difficult to be classified correctly. If the solution of LP1 leads to the misclassification of more than one alternative, then the model development process proceeds to the solution of a mixed-integer programming problem (MIP). MIP minimizes the number of these misclassified alternatives, while retaining all the other correct classifications. Finally, a second linear programming problem (LP2) is solved to further calibrate the pair of utility functions developed by MIP so that the classification of the reference alternatives is as clear as possible. For a reference alternative $\alpha_j \in C_k$, the case where $U_k(g_j) > U_{\sim k}(g_j)$ and $U_k(g_j) - U_{\sim k}(g_j) \approx 0$ indicates a marginally correct classification. On the other hand, the case where $U_k(g_j) > U_{\sim k}(g_j)$ and $U_k(g_j) - U_{\sim k}(g_j) \approx 1$ indicates that the correct classification of α_j in C_k is quite clear.

5 The Regression systems

5.1 Multinomial Logistic Regression

Binomial logistic regression is a form of regression deployed when the dependent is a dichotomy and the independents are of any type, **[8]**. Multinomial Logistic Regression provides solutions to problems with more than two classes of dependents, whilst in case multiple classes of the dependent variable can be ranked, then ordinal logistic regression is preferred. Continuous variables are not used as dependents in logistic regression, permitting only one dependent variable. Logistic regression may predict a dependent variable on the basis of continuous and categorical independents and to determine the percent of variance in the dependent variable explained by the independents, to rank the relative importance of independents, to assess interaction effects, and to understand the impact of covariate control variables. Logistic regression applies maximum likelihood estimation after transforming the dependent into a logit variable. Hence, logistic regression estimates the probability of a certain event. Logistic regression calculates changes in the log odds of the dependent, in contrast to OLS regression which processes the changes in the dependent variables.

The Multinomial Logistic Regression-MLR model with a ridge estimator was used to elaborate financial analysis of corporations. In the Multinomial Logistic Regression given k classes for n instances at m attributes, the dimension in matrix B of parameters is $m*(k-1)$. The probability for class j with the exception of the last class is:

$$P_j(X_i) = e^{X_iB_j} / ((\sum_{j=1,..(k-1)} e^{X_i*B_j})+1)$$

The probability of last class is:

$$1-(\sum_{j=1,..(k-1)} P_j(X_i)) = 1/((\sum_{j=1,..(k-1)} e^{X_i*B_j})+1)$$

Consequently the multinomial log-likelihood will have negative values as:

$$L = -\sum_{i=1,..n} [\sum_{j=1,..(k-1)} (Y_{ij} * \ln(P_j(X_i))) + (1 - (\sum Y_{ij})_{j=1,..(k-1)}) * \ln(1 - \sum_{2j=1,..(k-1)} P_j(X_i))] + ridge * B$$

The Quasi-Newton Method is implemented to seek the optimized values of $m*(k-1)$ variables, aiming to find the matrix B for which L is minimised, **[9]**. The initial Logistic Regression algorithm, does not compute instance weights, an adjusted algorithm was implemented through WEKA platform calculate the instance weights.

5.2 Additive Logistic Regression –Logitboost

According to **[10,11]**, Boosting is a considerably important development in the classification domain, where it applies in a sequential order a classification algorithm to reweighted versions of the training data, and on the next step takes a weighted majority vote of the sequence of produced classifiers, causing a accelerating improvement in performance. Additive modeling and maximum likelihood, for the two-class problem with boosting, consist an approximation to additive modeling on the logistic scale using maximum Bernoulli likelihood as a criterion **[12]**. Direct multi-class generalizations based on multinomial likelihood are derived that exhibit performance comparable to other recently proposed multi-class generalizations of boosting in most situations, and far superior in some. The general form of additive models is: $P_j(X_i) = \alpha + \Sigma f(X_i))$, describing the Additive Regression models as well, whilst AdaBoost M1 model is described by: $F(x) = \Sigma c_m f_m(x)$. LogitBoost in WEKA is a form of Additive Logistic Regression, having the ability to boost very simple learning schemes even in cases of multiple classes, **[13]**, performing in a superior manner than AdaBoost M1 algorithm. LogitBoost boosts schemes for numeric prediction, to create a combined classifier that predicts a categorical class, an activity that AdaBoost M1 does not perform. The classification process is implemented through a regression scheme as the base learner in multi-class problems, elaborating efficient internal cross-validation to determine appropriate number of iterations.

5.3 Simple Logistic

Linear Logistic Regression may also be created through SimpleLogistic function of WEKA platform. The SimpleLogistic function as a Linear Logistic Regression deploying a LogitBoost algorithm with regular regression functions as base learners to fit the logistic models. Cross-validation is used to acquire the optimal number of LogitBoost iterations that offer automatic attribute selection, **[14]**.

5.4 Logistic Model Trees-LMT

The Tree induction techniques with linear models were used extended to solve problems of supervised learning, both for the prediction of nominal classes and continuous numeric values, **[14]**. Prediction of numeric quantities, combined the mentioned techniques into model trees, containing linear regression functions at the leaves. Logistic regression, replacing the linear, with Tree Induction was used to formulate Logistic Model Trees-LMT solving classification problems. A stagewise fitting process produces the logistic regression models that can select relevant attributes in the data naturally, creating the logistic regression models at the leaves by incrementally refining those constructed at higher levels in the tree. Hence Logistic Model Trees are classification trees implementing in their leaves logistic regression functions. Logistic Model Trees may process binary and multi-class target variables, numeric, nominal attributes and missing values, **[14]**.

6 Data

Data were offered by 1411 companies in the credit portfolio of a Greek commercial bank, including 16 financial indices form the period 1994 to 1997: 1) EBIT/Total Assets, 2) Net Income/Net Worth, 3) Sales/Total Assets, 4) Gross Profit/Total Assets, 5) Net Income/Working Capital, 6)Net Worth/Total Liabilities 7)Total Liabilities/Total assets, 8) Long Term Liabilities /(Long Term Liabilities + Net Worth), 9)Quick Assets/Current Liabilities 10)(Quick Assets-Inventories)/Current Liabilities, 11)Floating Assets/Current Liabilities, 12)Current Liabilities/Net Worth, 13) Cash Flow/Total Assets, 14)Total Liabilities/Working Capital, 15)Working Capital/Total Assets, 16) Inventories/Quick Assets, and a 17th index with initial classification. done by bank executives. Test set was 50% of overall data, and training set 50%.

7 Results

7.1 Results of Multi Layer Perceptrons

MLP in WEKA software used backpropagation to train, providing a graphical representation of the network, having the ability to be altered only while the network is not running. Random numbers are used for setting the initial weights of the connections between nodes, and also for shuffling the training data. The network can reset with a lower learning rate. If the network diverges from the answer this will automatically reset the network with a lower learning rate and begin training again. The number of epochs to train through trainingTime was 500. The simple Multi Layer Perceptron with 1 hidden layer, had a quite successful convergence which was marginally the best of all regression models. It classified the 582 (82.43%) of initially characterized companies as healthy to the group if healthy, and the companies in distress as in the group of distressed companies in 87 (12.32%) cases, 3.39% of initially characterized as healthy companies were classified as in distress, and 1.84% of distressed companies were characterized as healthy. Kappa statistic which indicates the of interobserver agreement was the highest of all at 0.7939, the Root

Mean Square Error was 0.210, thus MSE is 0.0441 indicating an excellent fitness of the network output to the desired output. The computation time was short, at 58.97 seconds Table 1. The MLP with 2 hidden layers had a slightly different confusion matrix than the previous topology, with lower classifications of healthy companies to healthy at 82.29%, and error classifications of healthy companies to distressed at 3.54%, whilst the rest elements of the confusion matrix are similar to MLP with 1 hidden layer. Kappa statistic is near the previous level but slightly lower at 0.7892, and cost function of MSE was the lowest of all with the optimal fitness of the network output to the desired output for all logistic regression models. MLP with 4 hidden layers had identical confusion matrix with MLP of 2 layers, Kappa statistic and MSE as well.

Finally MLP with 3 hidden layers and MLP with 5 hidden layers were identical, achieving a fine confusion matrix and identical Kappa statistic and MSE, a result that was also achieved by the MLP with number of hidden layers a=(attribs + classes) / 2 = (16+2)/2 = 9. It could be said that a periodic behavior was observed where in the MLPs that the number of layers is produced by 2 were identical, whilst the MLPs with number of hidden layers divided by itself and 1 performed similarly. In general it can be said that the regression models did not have significant differences in their performance.

7.2 Results of hybrid neuro-genetic MLPs

The results from hybrid neuro-genetic MLP failed to converge as they only classified initially healthy companies as healthy at a rate 100%. but they classified all companies in distress as healthy. Analytically neuro-genetic MLP with 1 hidden layer had their cost expressed in MSE 0.375 determines a well fitness of the network output to the desired output, but the moderate value of correlation coefficient r at 0.542 indicates a medium partial correlation between the variables, hence the fit of the model to the data is moderate, and processing time of 6 hours 11 minutes. MLP with 2 hidden layers had the lowest cost, as MSE at 0.202 and a correlation coefficient r at 0.038 with an inadequate fitness of the network output to the desired output, with computation time at 15 hours 13 minutes. MLP with 3 hidden layers had a low MSE at 0.278 with a low partial correlation between the variables and an unsatisfactory fit of the model to the data, whilst r was negative at -0.283 identifying a variation in variables in opposite directions, and a processing time at 19 hours 19 minutes. MLP with 4 hidden layers had a cost, expressed in MSE at 0.325 and a correlation coefficient at -0.0244 that represents a slight a variation in variables in opposite directions with computing time at 47 hours and 44 minutes. Finally MLP with 5 hidden layers had a low MSE at 0.245 and a correlation coefficient at 0.240 providing an inadequate result, in a computation time of 56 hours and 55 minutes, table3.

Neural Net		0->0	0->1	1->0	1->1	Misclass.	Correct classific.	Kappa Statist.	MAE	MSE	RMSE	RAE	RRSE	Time
MultiLayer		580	26	13	87	39	667	0.7845	0.070	0.0441	0.210	26.79	60.28	58.97
Perceptron Fu		82.1%	3.6%	1.84%	12.3%	5.52%	94.47%							sec
	Layers													
MLP	1	582	24	13	87	37	669	0.7939	0.068	0.043	0.208	5.99%	59.73	4.56 sec
		82.43	3.39%	1.84%	12.32	5.24%	94.75%							
MLP	2	581	25	13	87	38	668	0.7892	0.068	0.043	0.208	25.96	9.52%	6.53 sec
		82.29	3.54%	1.84%	12.32	5.38%	94.61%							
MLP	3	580	26	13	87	39	667	0.7845	0.067	0.043	0.208	25.88	59.7%	9.75 sec
		82.15	3.68%	1.84%	12.32	5.52%	94.47%							
MLP	4	581	25	13	87	38	668	0.7892	0.069	0.044	0.209	26.58%	59.9%	10.94 se
		82.29	3.54%	1.84%	12.32	5.38%	94.61%							
MLP	5	580	26	13	87	39	667	0.7845	0.068	0.043	0.209	26.2%	59.9%	11.94 sec
		82.15	3.68%	1.84%	12.32	5.52%	94.47%							

Table 1. Results of MultiLayerPerceptrons

7.3 Results of Regressions

The Multinomial Logistic Regression had an adequate convergence where the initially categorized as healthy companies by human experts were classified as healthy at a rate of 82.43% (582 companies), table 1, the healthy companies classified falsely as in distress were a rate of 1.84% (13 cases), and the distressed companies classified as healthy were 4.10% (29 cases), whilst the distressed companies that were classified as in distress were 11.67% (82 cases). The model needed 0.39 seconds to be built, producing 664 correctly classified instances (94.051 %) in 42 incorrectly classifications (5.949 %). The Kappa statistic that measures interobserver variability was quite good at

0.7615, and Mean Absolute Error 0.0761, when the Root Mean Squared Error was 0.2194, indicating the cost function in the form of Mean Square Error as 0.0481 revealing a satisfactory fitness of the network output to the desired output, the Relative Absolute Error was 29.0656 %, and the Root Relative Squared Error at 60.2752 %.

Logit Boost had a satisfactory convergence, since the initially characterized healthy companies, by loan experts, were classified through Logistic Regression as healthy in a proportion of 82.43 %(582 companies), with 13 misclassified healthy companies in the category of the distressed companies, 30 companies initially categorized as in distress were put in the healthy category, and 81 distressed companies were classified as in distress. The performed iterations were 10, whilst the time taken to build model was very short at: 0.63 seconds. The evaluation on test split produced 663 correctly classified instances (93.9093 %), and 43 incorrect classifications (6.0907 %), the Kappa statistic was satisfactory at 0.7549, whilst the Mean Absolute Error was 0.0852, the Root Mean Squared Error received 0.2227 with the highest cost function of MSE at 0.04959, between all logistic regressions, in a well fitted network output to the desired output, and Relative Absolute Error reached 32.5318 %, with a Root Relative Squared Error at 61.1839 % given that the training set was on the 50% split of the initial 1411 companies.

	Training set 1997 (%)	1996 (%)	1995 (%)	1994 (%)	Test set 1997	1996 (%)	1995 (%)	1994 (%)
Type I error	0.0	15.0	28.0	32.0	3.4	20.3	22.9	28.8
Type II error	1.0	16.0	13.0	14.0	9.9	17.2	19.3	19.2
Overall error	0.5	15.5	20.5	23.0	6.61	18.8	21.1	24.0

SimpleLogistic function had an accurate convergence with 83.56% of the companies initially characterized as healthy by bankexecutives to be classified as healthy by the model (590 cases), 2.26% of the healthy companies were put in the distress category (16 cases), 2.40% of companies in distress were classified as healthy (17 cases), and 23.51% of in distress companies were classified as in distress (83 cases). The model required 13.78 seconds to be built, revealing 673 correctly classified instances (95.3258 %), with 33 incorrectly classified instances (4.6742 %). The Kappa statistic was quite adequate at 0.807, MAE 0.073, the RMSE 0.2036 offering the cost function as the Mean Square Error at 0.041 was satisfactory, but compared to the other logistic regression models had the highest value, with a satisfactory fitness of the network output to the desired output nevertheless. RAE was 27.8366 %, and RRSE at 58.2119%.

The Logistic Model Tree classified the healthy companies according to bank experts in the class of healthy at a rate of 83.56% (590 cases), whilst some healthy companies were classified as in distress at a rate 2.26% (16 cases), the misclassifications included companies in distress which were categorized as healthy at a rate 2.40% (17 cases), finally the distressed companies were classified as in distress at a rate 23.51% (83 cases). With split 50% for the training data the Logistic Model Tree had 6 leaves and its size was 11, whilst the time taken to build model was 54.92 seconds. The correctly classified instances were 673 (95.3258 %) and the incorrectly 33 (4.6742 %). Kappa statistic was very good at 0.807, with a MAE at 0.073, RMSE 0.2036 supplying the cost function with MSE at 0.0414 at an adequate level, in the lowest values among the four different regression types, providing an excellent fitness of the network output to the desired output, RAE was 27.8366 %, and RRSE 58.2119 %. Hence Logistic Model Trees and Simple Logistic had the optimal performance between all logistic regression models, with a slight difference.

7.4 Results of M.H.DIS.

Multicriteria Hierarchical Discrimination (M.H.DIS.), provided the following results, using criteria of 11 financial indices, **[15[.** The overall error is the average of type I and type II errors. The cost with type I errors is higher than the cost of type II errors. The possibility apriori that a company belongs to the group of low risk is significantly smaller than the possibility to belong to the high risk group. The overall error for the training set was 0.5% in 1997, 15.5% in 1996, 20.5 in 1995, 23% in 1994, with a total of 59.5% for all years and an average of 14.875% per year.

Table 5. Results of MHDIS

8 Conclusions

It is obvious that the simple Multi Layer Perceptrons performed much better than neuro-genetic MLPs. They converged with a high number of successful classifications, producing a very low Mean Square Error. The hybrid MLP failed completely to converge as they all classified companies in distress as healthy companies, with low MSE and much lower correlation coefficient, indicating a medium fitness of the network output to the desired output, and a medium partial correlation between the variables, emphasizing that the fit of the model to the data was moderate. Although simple MLP was proven to be better than hybrid MLPs, it still needs further development as it produces the groups of classification only, with no further analytical result of each company. The simple Multi Layer Perceptron with 1 hidden layer, offered the optimal results in all MLPs. But the SimpleLogistic function and Logistic Model Trees had better convergence in the confusion matrix than all types of MLPs, since Kappa statistic which indicates the of interobserver agreement was the highest of all at 0.807, MSE is 0.0441 indicating an excellent fitness of the network output to the desired output, the computation time was slightly longer than MLP at 4.56: 13.78 for Simple Logistic, and 59.94 for LMT . We dint have the precice time of MHDIS, which needed some minutes.

In MHDIS the overall error for the training set was 0.5% in 1997, 15.5% in 1996, 20.5 in 1995, 23% in 1994, with a total of 59.5% for all years and an average of 14.875% per year, which is higher than the number of incorrect classifications (misclassifications). Simple Logistic and Logistic Model Trees had the lowest number of misclassifications in the period of 1994 to 1997 with a 4.67% totally, followed by Multi Layer Perceptron of 1 hidden layer at a 5.24%, MHDIS had an average of 14.5% incorrect classifications per year. Also the Mean Absolute Error of the Logistic regressions and the MLP, is much lower than the average error of 14.5% per year of MHDIS. Consequently Simple Logistic and Logistic Model Trees are the most excellent methods for corporate financial analysis than the rest methods examined in this research.

Bibliography

[1]. Loukeris N., Matsatsinis N. (2006) -Corporate Financial Evaluation and Bankruptcy Prediction implementing Artificial Intelligence methods-, WSEAS Transactions in Business and Economics, (4), Vol. 3, April

[2]. Lippman R. (1987)-An introduction to computing with neural nets-, IEEE Trans, ASSP Magazine 4, 4-22.

[3]. Makhoul J.(1992)-Pattern recognition properties of neural networks." Proc. 1991 IEEE Workshop on Neural Networks for Signal Processing, pp 173-187

[4]. Rumelhart, D., Hinton, G. & Williams, R. (1986)-Learning internal representations by error back-propagation, in *Parallel distributed processing: explorations in the microstructure of cognition* (Rumelhart, D. & McClelland, J., eds.), MIT Press, Cambridge, MA.

[5]. Widrow B. and Lehr M. (1990) – 30 years of adaptive neural networks: perceptron, madaline and backpropagation- Proc. IEEE 78, 1415-1442, 1990.

[6]. Principe J., Euliano N., and Lefebvre W. (2000)-Neural and Adaptive Systems: Fundamentals Through Simulations-ed. Wiley

[7]. Zopounidis C. and Doumpos M. (2000)-Building additive utilities for multi-group hierarchical discrimination: The MHDIS method-, Optimization Methods and Software 14(3), 219–24

[8]. Garson D. (1998) http://www2.chass.ncsu.edu/garson/PA765/logistic.htm

[9]. Witten I. and Frank E., (2000), Data Mining, Morgan Kaufmann Publ., London, **[10].** Freund Y. and Schapire R. (1995)- A decision theoretic generalization of online learning and application to boosting-, In European Conference on Computational Learning Theory, 23-37, 1995 and Journal of Computer and System Sciences, 55(1):119-139, 1997.

[11]. Freund Y. and Schapire R. (1996)- Experiments with a new boosting algorithm-, Machine Learning: Proceedings of the Thirteenth International Conference, 148-156

[12]. Friedman J., Hastie T., and Tibshirani R. (2000)-Additive logistic regression, a statistical view of boosting - Annals of Statistics, 28:307–337, 2000.

[13]. Friedman J., Hastie T. and Tibshirani R. (1998)-Additive Logistic Regression: a Statistical View of Boosting-, Annals of Statistics, vol. 28, no. 2, 2000

[14]. Landwehr N., Hall M. and Frank E. (2003)-*Logistic model trees*. In Proc. of the 14th ECML, 241-252

[15[. Doumpos M., Kosmidou K., Baourakis G., Zopounidis C. (2002)- Credit risk assessment using a multicriteria hierarchical discrimination approach: A comparative analysis-European Journal of Operational Research 138, 392–412

Hybrid NN with GA	*Layers*	Active Confusion Matrix 0->0	0->1	1->0	1->1	Performance MSE	NMSE	r	%error	AIC	MDL	Time
MLP	1	100	0	100	0	0.37	1.457	0.542	9402632	407.53	331.13	6 h 11’
MLP	2	100	0	100	0	0.20	1.096	0.038	15415127	117.84	91.08	15 h 13’
MLP	3	100	0	100	0	0.27	1.082	-0.283	23083110891.488		69.307	19 h 19’
MLP	4	100	0	100	0	0.32	1.266	-.0244	1693938	2123.7	1744	47 h 44’
MLP	5	100	0	100	0	0.24	0.951	0.240	16839440	1834.02	1504	56 h 55’

Table 3. Results of hybrid MLP with Genetic Algorithms optimization

Regressions	0->0	0->1	1->0	1->1	Misclass	Correct class.	K-stat	MAE	MSE	RMSE	RAE %	RRSE %	Time (sec)
Logistic	582 82.43%	13 1.84%	29 4.10%	82 11.67%	42 5.94%	664 94.05%	0.761	0.076	0.048	0.2194	29.06%	60.27	0.39
Logit boost	582 82.43%	13 1.84%	30 4.24%	81 11.47%	43 6.097%	663 93.9%	0.754	0.085	0.049	0.2227	32.53	61.18	0.63
Simple Logistic	590 83.56%	16 2.26%	17 2.40%	83 23.51%	33 4.67%	673 95.3%	0.807	0.07	0.041	0.2036	32.75%	62.60	13.7
Logistic Model Trees	590 83.56%	16 2.26%	17 2.40%	83 23.51%	33 4.67%	673 95.3%	0.807	0.07	0.041	0.2036	27.83%	58.21	54.9

Table 4. Results for the regressions as training set splitted to 50% of overall data