International Journal of Information Technology & Decision Making
Vol. 9, No. 6 (2010) 959–978

DOI: 10.1142/S0219622010004111

FUZZY, DISTRIBUTED, INSTANCE COUNTING, AND DEFAULT ARTMAP NEURAL NETWORKS FOR FINANCIAL DIAGNOSIS

ANATOLI NACHEV*, SEAMUS HILL† and CHRIS BARRY‡

Business Information Systems
Cairnes School of Business & Economics
NUI, Galway, Ireland
**anatoli.nachev@nuigalway.ie*
†*seamus.hill@nuigalway.ie*
‡*chris.barry@nuigalway.ie*

BORISLAV STOYANOV

Department of Computer Systems and Technologies
Shumen University, Bulgaria
bpstoyanov@abv.bg

This paper shows the potential of neural networks based on the Adaptive Resonance Theory as tools that generate warning signals when bankruptcy of a company is expected (bankruptcy prediction problem). Using that class of neural networks is still unexplored to date. We examined four of the most popular networks of the class — fuzzy, distributed, instance counting, and default ARTMAP. In order to illustrate their performance and to compare with other techniques, we used data, financial ratios, and experimental conditions identical to those published in previous studies. Our experiments show that two financial ratios provide highest discriminatory power of the model and ensure best prediction accuracy. We examined performance and validated results by exhaustive search of input variables, cross-validation, receiver operating characteristic analysis, and area under curve metric. We also did application-specific cost analysis. Our results show that distributed ARTMAP outperforms the other three models in general, but the fuzzy model is best performer for certain vigilance values and in the application-specific context. We also found that ARTMAP outperforms the most popular neural networks — multi-layer perceptrons and other statistical techniques applied to the same data.

Keywords: Neural networks; data mining; ARTMAP; bankruptcy prediction.

1. Introduction

One of the most significant threats for many businesses today, despite their size and the nature of their operation, is insolvency. The economic cost of business failures is significant. The suppliers of capital, investors, and creditors, as well as management and employees, are severely affected by business failure. The need for

*Corresponding author.

reliable empirical models that predict corporate failure promptly and accurately is imperative to enable decision makers to take either preventive or corrective action.

Estimating potential for insolvency, decision makers usually apply scoring systems, which takes into account factors, such as leverage, earnings, and reputation. Due to lack of metrics and subjectiveness in estimates, sometimes decisions can be unrealistic and not consistent.[1]

Generally, a prediction of corporate insolvency can be viewed as a pattern recognition problem, and as such, it can be solved using one of two approaches: structural and empirical. The former derives the probability of a company for default, based on its characteristics and dynamics, while the latter approach relies on previous knowledge and relationships in that area, learns from existing data or experience, and deploys statistical or other methods to predict failure.

Kumar and Ravi[2] provide a comprehensive survey on empirical techniques used to predict insolvency, all grouped into two categories: statistical and intelligent. Most popular statistical techniques are linear discriminant analysis, multivariate discriminate analysis, quadratic discriminant analysis, logistic regression, and factor analysis. Among the intelligence techniques most common are neural networks, decision trees, case-based reasoning, evolutionary approaches, etc.[2,3] Traditional statistical techniques have often been criticized because of their assumptions about linear separability of training data, multivariate normality, and independence of the predictive variables.[1,4] Such constraints are incompatible with the complex nature, boundaries, and interrelationships of most of financial ratios used for learning and prediction. The intelligent techniques have shown themselves more appropriate for the task. For instance, neural networks do not rely on *a priori* assumptions about the distribution of data and work well in unstructured and noisy environment.[1,5]

Multi-layer perceptron is most common, well known, and widely used model of supervised neural networks. Sharda and coworkers[6–9] used five financial ratios introduced by Altman[10] and multi-layer perceptron to predict bankruptcy. They reported significantly better prediction accuracy than statistical techniques applied to the same data. Rahimian *et al.*[11] compared the performance of multi-layer perceptron, Athena (an entropy-based neural network), and single-layer perceptron on the bankruptcy prediction problem using the same Altman's ratios. Serrano-Cinca[12] also used Altman's ratios and compared his multi-layer perceptron with others' and some statistical techniques. Bell *et al.*,[13] Hart,[14] Yoon *et al.*,[15] and Curram and Mingers[16] also compared the classifying power of different statistical tools and multi-layer perceptron.

Despite their good performance, multi-layer perceptrons have a well-known drawback — unclear structure. Choice of optimal network architecture, in particular number of layers and hidden nodes, is a challenge that has no theoretical answer. A good architecture for an application can be found only by empirical means and experiments.

Many other techniques have been used to predict bankruptcy. Salcedo-Sanz *et al.*[17] propose genetic programming applied to data from insurance companies.

Their results were compared with rough sets approaches. Shin and Lee[18] used support vector machines for modeling business failure prediction. Cielen *et al.*[19] suggested the combined use of linear programming and inductive machine learning.

Our research was motivated by the fact that a class of neural networks — those based on the Adaptive Resonance Theory — is still unexplored as a tool for bankruptcy prediction. Here we explore four of them — fuzzy ARTMAP, distributed ARTMAP, instance counting ARTMAP, and default ARTMAP. We selected a data set that has already been used with other neural network models, in particular multi-layer perceptrons and self-organizing feature maps, which allow us to compare results.

This paper is organized as follows: Sec. 2 provides an overview of the neural network architectures used in this study; Sec. 3 discusses the research design, data set, data preprocessing, and techniques of analysis; Sec. 4 presents and discusses experimental results; and Sec. 5 gives conclusions.

2. Neural Networks

There is a variety of neural network models for clustering and classification, ranging from very general architectures, which are applicable to most of the learning problems, to highly specialized networks that address specific problems. Each model has a particular topology that determines the neurons (nodes) layout and a specific algorithm to train the network or to recall stored information. Among the models, the most common is the *multi-layer perceptron* (MLP), which has a feed-forward topology and error-backpropagation learning algorithm.[20] Authors often call MLP just neural networks, but this is not quite correct as there are other members of the big family of neural networks, such as those with recurrent topology — self-organizing feature maps,[21] Hopfield networks,[22,23] and Adaptive Resonance Theory networks[24] discussed here.

2.1. *Neural networks based on the Adaptive Resonance Theory*

The Adaptive Resonance Theory (ART), introduced by Grossberg in 1970, began with analysis of human cognitive information processing.[24,25] It led to creation of a family of self-organizing neural networks for fast learning, pattern recognition, and prediction. Some popular members of the family are both unsupervised models: ART1, ART2, ART2-A, ART3, fuzzy ART, distributed ART; and supervised models: ARTMAP, instance counting ARTMAP, fuzzy ARTMAP, distributed ARTMAP, and default ARTMAP.[26] Fundamental computational design goals were providing memory stability, fast or slow learning mechanism in an open and evolving input environment, and implementation by a system of ordinary differential equations approximated by appropriate techniques.[24]

A remarkable feature of the ART neural networks is their on-line, one-pass fast learning algorithm. In contrast, MLP offers off-line slow learning procedure that requires availability of all training patterns at once in order to avoid catastrophic forgetting in an open input environment. The adaptiveness makes ART networks

suitable for classification problems in dynamic and evolving domains, whereas MLP are mostly suited for problems related to static environments.

Many applications of the ART networks are classification problems, where the trained system tries to predict a correct category of an input sample.[27–29] In fact, these tasks are pattern recognition problems, as classification may be viewed as a many-to-one mapping task that entails clustering of the input space and then association of the produced clusters with a limited number of class labels.

2.2. *ARTMAP architecture*

ARTMAP is a supervised neural network which consists of two unsupervised ART modules, *ARTa* and *ARTb* and an inter-ART module, called a map-field (see Fig. 1).[30] An ART module has three layers of nodes: the input layer *F0*, the comparison layer *F1*, and the recognition layer *F2*. A set of real-valued weights W_j is associated with the *F1*-to-*F2* layer connections between nodes. Each *F2* node represents a recognition category that learns a binary prototype vector w_j. The *F2* layer is connected, through weighted associative links, to a map-field F^{ab}.

The following algorithm[30,31] describes the ARTMAP learning:

(1) *Initialization.* Initially, all the *F2* nodes are uncommitted, all weight values are initialized. Values of network parameters are set.
(2) *Input pattern coding.* When a training pattern is presented to the network as a pair of two components (a, t) where a is the training sample and t is the class label, a process called *complement coding* takes place. It transforms the pattern into a form suited to the network. A network parameter called *vigilance parameter* (ρ) is set to its initial value. This parameter controls the network

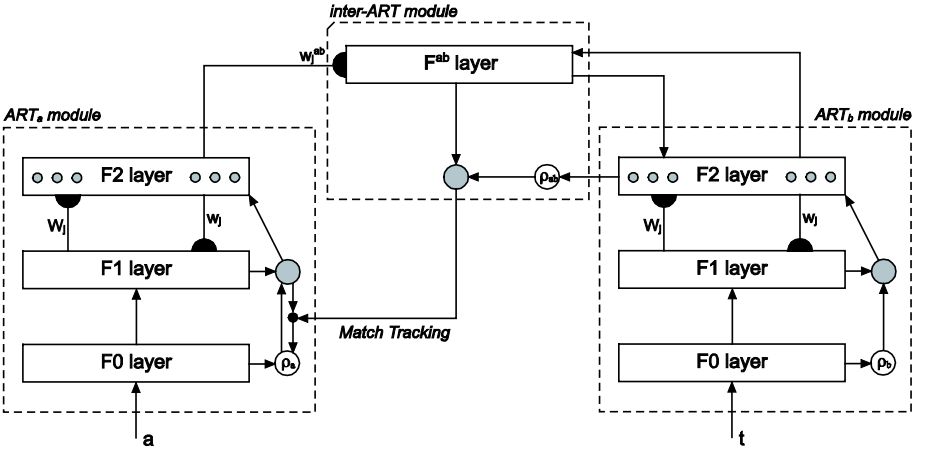

Fig. 1. Block diagram of ARTMAP neural network, adapted from Carpenter *et al.*[30]

'vigilance', i.e., the level of details used by the system when it compares the input pattern with the memorized categories.

(3) *Prototype selection.* The input pattern activates layer *F1* and propagates to layer *F2*, which produces a binary pattern of activity such that only the *F2* node with the greatest activation value remains active, i.e., 'winner-takes-all'. The activated node propagates its signals back onto *F1*, where a *vigilance test* takes place. If the test is passed, then *resonance* is said to occur. Otherwise, the network inhibits the active *F2* node and searches for another node that passes the vigilance test. If such a node does not exist, an uncommitted *F2* node becomes active and undergoes learning.

(4) *Class prediction.* The class label t activates the F^{ab} layer in which the most active node yields the class prediction. If that node constitutes an incorrect class prediction, then another search among *F2* nodes in Step 3 takes place. This search continues until either an uncommitted *F2* node becomes active (and learning directly ensues in Step 5), or a node that has previously learned the correct class prediction becomes active.

(5) *Learning.* The neural network gradually updates its adaptive weights toward the presented training patterns until a convergence occur. The learning dynamic can be described by a system of ordinary differential equations.[30]

2.3. *Fuzzy, distributed, instance counting, and default ARTMAP neural networks*

Fuzzy ARTMAP was developed as a natural extension to the ARTMAP architecture.[32] This is accomplished by using *fuzzy ART* modules instead of ART1, which in fact replaces the crisp (binary) logic embedded in the ART1 module with a fuzzy one. In fact, the intersection operator ($\cap$) that describes the ART1 dynamics is replaced by the fuzzy AND operator ($\wedge$) from the fuzzy set theory ($(p \wedge q)_i \equiv \min(p_i, q_i)$).[33] This allows the fuzzy ARTMAP to learn stable categories in response to either analog or binary patterns, in contrast to the basic ARTMAP, which operates with binary patterns only.

An ART1 module maps categories into *F2* nodes according to the rule winner-takes-all, as discussed above, but this way of functioning can cause category proliferation in a noisy input environment. The explanation of this fact is that the system adds more and more *F2* category nodes in order to meet the demands of predictive accuracy believing that the noisy patterns are samples of new categories. To address this drawback, Carpenter *et al.*[34] introduced a new *distributed ART* module, which features a number of innovations, such as new distributed *instar* and *outstar* learning laws. If the ART1 module of the basic ARTMAP is replaced by a distributed ART module, the resulting network is called *distributed ARTMAP.*[34,35] Some experiments show that a distributed ARTMAP retains the fuzzy ARTMAP accuracy while significantly reducing the network size.[35]

Instance Counting (*IC*) *ARTMAP* adds to the basic fuzzy ARTMAP system new capabilities designed to solve computational problems that frequently arise in prediction. One such problem is inconsistent cases, where identical input vectors correspond to cases with different outcomes. A small modification of the fuzzy ARTMAP match tracking search algorithm allows the IC ARTMAP to encode inconsistent cases and make distributed probability estimates during testing, even when training employs fast learning.[36] IC ARTMAP extends the fuzzy ARTMAP giving a good performance on various problems.

A comparative analysis of the ARTMAP modifications, including fuzzy ARTMAP, IC ARTMAP, and distributed ARTMAP, has led to the identification of the *default ARTMAP* network,[26] which combines the winner-takes-all category node activation during training, distributed activation during testing, and a set of default network parameter values that define a ready-to-use, general-purpose neural network for supervised learning and recognition. The default ARTMAP features simplicity of design and robust performance in many application domains.

3. Research Design

Our motivation to conduct this research was to fill a gap in the bankruptcy prediction field by using four models of neural networks, still unexplored in that domain. The main research objective was to investigate how ARTMAP models find common characteristics amongst failing firms and distinguish them from the viable firms in order to predict bankruptcy. Another objective was to investigate how different variants of the ARTMAP networks perform and which one is the most appropriate. We also wanted to compare ARTMAP performance with that of other classification techniques by using the same data sets and experimental conditions. Part of our study aimed to investigate if the ARTMAP models are sensitive to outliers, i.e., observations in the data sets that are numerically distant from the rest of the data. Such data values can often be misleading to classifiers and have a significant effect on the correct classification.

In order to estimate neural network performance we used different metrics and analysis techniques, such as accuracy, true and false-positive rates, receiver operating characteristics analysis, area under the curve, unit cost, and cost analysis.

In order to validate results, we used the k-fold cross-validation method, where $k = 5$. According to Carpenter *et al.*,[30] the value 5 is sufficient to validate the majority of ARTMAP applications and 5 is recommended as a default parameter value.

Finally, we wanted to estimate the ARTMAP neural networks efficiency in terms of training and testing time, and memory consumption.

3.1. *The data*

The data set, we used, has been used in other studies in the domain.[7,9,11,12] It contains financial information from the Moody's Industrial Manual for a number of years for a total of 129 firms, of which 65 are bankrupt and the rest are solvent.

The data entries have been randomly divided into two subsets: one for training, made up of 74 firms, of which 38 are bankrupt and 36 are non-bankrupt; another set for testing, made up of 55 firms, of which 27 are bankrupt and 28 are non-bankrupt. In order to follow the experimental conditions of the studies mentioned above and to be able to compare results, we decided not to use train and test techniques, such as 'leave-one-out' and stick to the train and test data sets as they have been used in the studies.

Brigham and Gapenski[37] suggest that there is an empirical basis for grouping financial ratios into seven categories: return on investment, financial leverage, capital turnover, short-term liquidity, cash position, inventory turnover, and receivables turnover. Altman[10] found, however, that only certain of these ratios have discriminating capability. This research uses the Altman's ratios, namely:

— *R1: Working Capital/Total Assets.* Working capital is defined as the difference between current assets and current liabilities. It can be regarded as net current assets, therefore, a good measure of short-term solvency.
— *R2: Retained Earnings/Total Assets.* Retained profits are obviously higher for older, established firms than for new entrants, other things being equal. This ratio might, therefore, seem to be unfairly weighted in favor of older firms, but actually it is not.
— *R3: EBIT/Total Assets.* EBIT is the abbreviation for 'Earnings Before Interest and Taxes'. This ratio is useful for comparing firms in different tax situations and with varying degrees of financial leverage.
— *R4: Market Value of Equity/Book Value of Total Debt.* This measure examines a firm's competitive marketplace value.
— *R5: Sales/Total Assets.* This ratio, sometimes called the assets turnover ratio, measures the firm's asset utilization.

3.2. *Preprocessing*

Most applications with neural networks transform rough data into form suitable for training and testing. A common form of transformation is linear rescaling of the input variables. This is necessary as different variables may have values which differ significantly because of different units of measurements. Such a disbalance can reduce the predictive abilities of the model as some of the variables can dominate over others. The linear transformation we used arranged for all of the inputs to have similar values. Each of the input variables x_i was treated independently and its mean and variance were calculated using:

$$\bar{x}_i = \frac{1}{N} \sum_{n=1}^{N} x_i^n, \tag{1}$$

$$\sigma_i^2 = \frac{1}{N-1} \sum_{n=1}^{N} (x_i^n - \bar{x}_i)^2, \tag{2}$$

where $n = 1, \ldots, N$ labels the patterns. We then defined a set of rescaled variables given by:

$$\tilde{x}_i^n = \frac{x_i^n - \bar{x}_i}{\sigma_i}. \tag{3}$$

Transformed variables have zero mean and unit standard deviation over the transformed training set. The linear transformation, however, is not sufficient for training or testing the neural networks as they need input in the form of M-dimensional vectors of floating point numbers in between 0 and 1. A second preprocessing step maps the data values into $[0, 1]$ using:

$$\hat{x}_i^n = \frac{(\tilde{x}_i^n - \tilde{x}_i^{\min})}{(\tilde{x}_i^{\max} - \tilde{x}_i^{\min})}, \tag{4}$$

where $x_i^{\max}$ and $x_i^{\min}$ are the max and min values of the variable x_i, respectively.

3.3. *Reduction of dimensionality*

The principal motivation for reduction of dimensionality is that a network with fewer inputs has fewer adaptive parameters to be determined, and these are more likely to be properly constrained by a data set of limited size (as the data set we used), leading to a network with better generalization properties. In addition, a neural network with fewer weights may be faster to train. If too much information is lost after reduction of dimensionality, then the resulting reduction in performance cannot compensate any improvement arising from avoidance of overfitting (or overtraining).[38] In cases where learning is performed too long or where training examples contain too much information, the neural network may adjust to very specific random features of the training data that have no causal relation to the target function. In this case, the performance on the training examples still increases while the performance on unseen data becomes worse because the neural network loses its ability to generalize. Using univariate F-ratio analysis, Serrano-Cinca[12] ranked Altman's ratios and suggested that the second and third variables have a greater discriminatory power in contrast to the fifth one. The univariate analysis, however, does not estimate combinations of variables. Furthermore, the optimal variable selection is different for different classification models, i.e., there is no guarantee that an optimal set for an MLP would perform well with an ARTMAP neural network. Ideally, the optimal subset for a model can be selected by the exhaustive (brutal-force) search approach, which checks whether each variable subset satisfies the requirements of the bankruptcy predictions problem. For n possible variables we have a total of 2^n possible subset, since each variable can be present or absent. We took advantage of the small number of variables (five) to apply exhaustive search using 31 subsets of train and test data sets.

3.4. *Performance metrics*

Binary classifiers, such as the ARTMAP neural nets, map test data set instances to one element of the set of positive and negative class labels $\{p, n\}$. Outcomes from classifications can be summarized into a 2×2 matrix (confusion matrix or contingency table) where the four values are the following:

— true positive (TP), also known as hits;
— true negative (TN), or correct rejection;
— false positive (FP), called also type I error; and
— false negative (FN), or type II error.

The confusion matrix forms the basis for several common metrics defined below:

— true positive rate (TPR), known as hit rate, recall, or sensitivity:
$TPR = TP/(TP + FN) = TP/P$;
— false positive rate (FPR), known as false alarm rate, or fall-out:
$FPR = FP/(FP + TN) = FP/N$;
— true negative rate (TNR), known as specificity:
$TNR = SPC = TN/(FP + TN) = 1 - FPR$;
— false negative rate (FNR), or positive error:
$FNR = FN/(TP + FN)$
— accuracy (ACC):
$ACC = (TP + TN)/(P + N)$; and
— macro-average (MAVG)
MAVG = AVG (TPR, TNR).

3.5. *Receiver operating characteristic analysis*

Despite accuracy is common figure of merit for classifiers, it can be misleading if an important class is underrepresented, e.g., a data set may contain only few instances of bankrupt companies. In that case sensitivity and specificity can be more relevant performance estimators.[29] Second, the accuracy depends on the classifier's operating threshold, such as the vigilance parameter of ARTMAP, and choosing the optimal threshold can be challenging. Finally, different types of misclassifications have different costs. For example, in the bankruptcy prediction domain, errors of type I and type II can produce different consequences and have different costs.

In recent years there is an increased use of the *receiver operation characteristic* (ROC) analysis in the machine learning research due to realization that simple classification accuracy is often a poor metric for measuring performance. ROC curves have properties that make them especially useful for applications with skewed class distribution and unequal classification error costs.[39] An ROC space is defined by FPR and TPR as x and y axes, respectively, which depict relative trade-offs between true positive (benefits) and false positive (costs). Each prediction result or one instance of a confusion matrix represents one point in the ROC space. The

perfect classification would yield a point in the upper left corner or coordinate $(0, 1)$, representing 100% sensitivity (all true positives are found) and 100% specificity (no false positives are found). A completely random guess would give a point along the no-discrimination line from the left bottom to the top right corner.

Soft or probabilistic classifiers, such as naive Bayesian and MLP neural network, produce probability values representing the degree to which class the instance belongs to. For these methods, setting a threshold value will determine a point in the ROC space. In contrast, crisp or discrete classifiers, such as ARTMAP and decision trees, yield numerical values or binary label. When a set is given to such classifiers, the result is a single point in the ROC space. As we want to generate a ROC curve from an ARTMAP classifier, it can be converted to soft classifier by 'looking inside' it. Varying the vigilance parameter generates aggregation of points in the ROC space.

Comparing classifiers we may want to reduce ROC performance to a single scalar value representing expected performance. A common method is to calculate the *area under the ROC curve* (*AUC*).[37] Calculation of AUC in the case of crisp classifiers requires trapezoidal approximation (5). Best classification model maximizes the AUC index.

$$AUC = \frac{1}{2} \sum_{i \in ROCCH} (TPR_i + TPR_{i+1})(FPR_{i+1} - FPR_i). \tag{5}$$

4. Empirical Results and Discussion

In machine learning applications, classification performance is often measured by accuracy as a figure of merit. For a given operating point of a classifier, the accuracy is the total number of correctly classified instances divided by the total number of all available instances.

A series of experiments sought to estimate bankruptcy prediction accuracy of the fuzzy, IC, distributed, and default ARTMAP neural networks. Using 5-fold cross-validation and in accordance with the exhaustive search strategy discussed above, 31 data sets were composed and indexed from 1 to 31 where indexes represent data relevant to subsets of financial ratios, namely: indexes 1 to 5 represent {R1} to {R5}; 6 to 15 for pairs of ratios {R1, R2}, {R1, R3} to {R4, R5}; 16 to 25 for triples — {R1, R2, R3}, {R1, R2, R4} to {R3, R4, R5}; 26 to 30 for quarters; and 31 for the full set {R1, R2, R3, R4, R5}. In order to investigate how vigilance parameter is related to the prediction accuracy, each subset was presented to the four models and iterated 41 times using vigilance parameter values from 0 to 1 with increment of 0.025. Figure 2 shows the results for each ARTMAP model. Axis x of each model represents one subset of financial ratios (from 1 to 31); axis y counts prediction accuracy in % achieved by that model. Each stem (subset) contains a number of circles, where a single circle corresponds to a fixed vigilance parameter value. In that way the figure represents prediction accuracy obtained by varying two components: subsets (collection of financial ratios), and the ARTMAP vigilance.

The figure shows that if an individual financial ratio is used (stems 1–5), second (R2) and fourth (R4) obtain highest accuracy regardless of the model, namely: 80% (R4) for the fuzzy; 78.2% (R2 or R4) for the distributed; 80% (R2) for the IC; and 80% (R4) for the default model. We can summarize that using single financial ratio fuzzy, IC, and default perform better than distributed ARTMAP; second, when ARTMAP works with individual Altman's ratios, two of them have highest discriminatory power — R2 and R4; and finally, all ARTMAP models used with a single Altman's ratio can outperform the statistical technique linear discriminant analysis (74.5% accuracy) used with all ratios.[7]

From the results above we could expect that when R2 and R4 are used together, their combined discriminatory power can lead to even better results. Indeed, Fig. 2 shows that subset 11 {R2, R4} yields the highest score for all ARTMAP models. The subset 24 {R2, R4, R5} is second best. After well-tuned vigilance parameter and using financial ratios R2 and R4, ARTMAP neural networks achieve the following prediction accuracy: 85.5% for the fuzzy and default model; 83% for the IC and distributed. It can be noticed that fuzzy and default ARTMAP (85.5%) outperform the best accuracy reported from MPL[12] (83%). At the same time IC and distributed

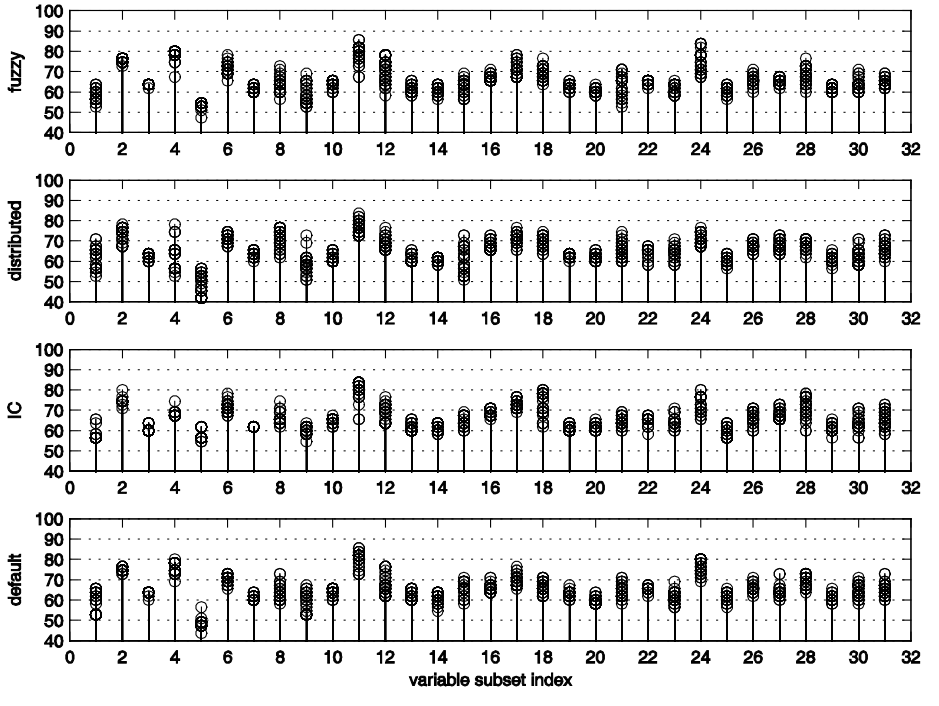

Fig. 2. Prediction accuracy of fuzzy, distributed, IC, and default ARTMAP neural networks by varying Altman's variables (31 possible subsets exist) and varying system's vigilance (41 vigilance parameter values from 0 to 1 with an increment of 0.025 tested). Axis x represents variable subsets; axis y — prediction accuracy in %; each stem shows accuracy values obtained by varying the system vigilance.

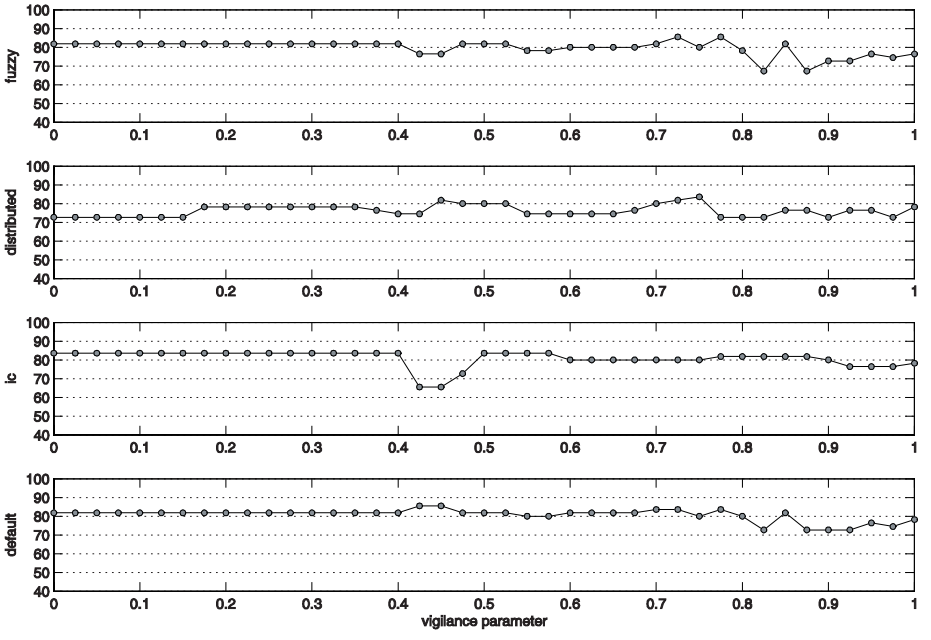

Fig. 3. Prediction accuracy in % obtained by fuzzy, distributed, IC, and default ARTMAP neural networks using variables {R2, R4} and 41 values of the vigilance parameter from 0 to 1 with incremental step of 0.025.

models score 83.6%, which is equal to the MLP's one. Figure 3 provides a more detailed view of prediction accuracy obtained by the four ARTMAP models using the best performing subset of financial ratios {R2, R4} and varying the vigilance parameter from 0 to 1. It can be seen that in the first part of the interval [0, 0.4], the fuzzy, IC, and default ARTMAP achieve a steady accuracy of at least 81.8% (83.6% for the IC), but in the rest of the interval it varies and has peaks at certain parameter values or intervals. Only the distributed model has unsteady and relatively low accuracy in the whole interval.

Table 1 compares misclassified companies (patterns) from the four ARTMAP models discussed here and other techniques: linear discriminant analysis,[7] MLP[7]; MLP[12]; and single-layer perceptron,[11] MLP,[11] and Athena network.[11] All models use the same experimental conditions. The table shows that the fuzzy and default misclassify 8; distributed and IC ARTMAP and MLP[12] — 9, all other models — 10, except linear discriminant analysis which misclassifies 14.

4.1. *ROC analysis results*

Figure 4 represents the ROC space for the fuzzy, IC, distributed, and default models, respectively. Each point in the figure represents a classifier obtained by certain value(s) of the vigilance parameter (text below Fig. 4 shows those values).

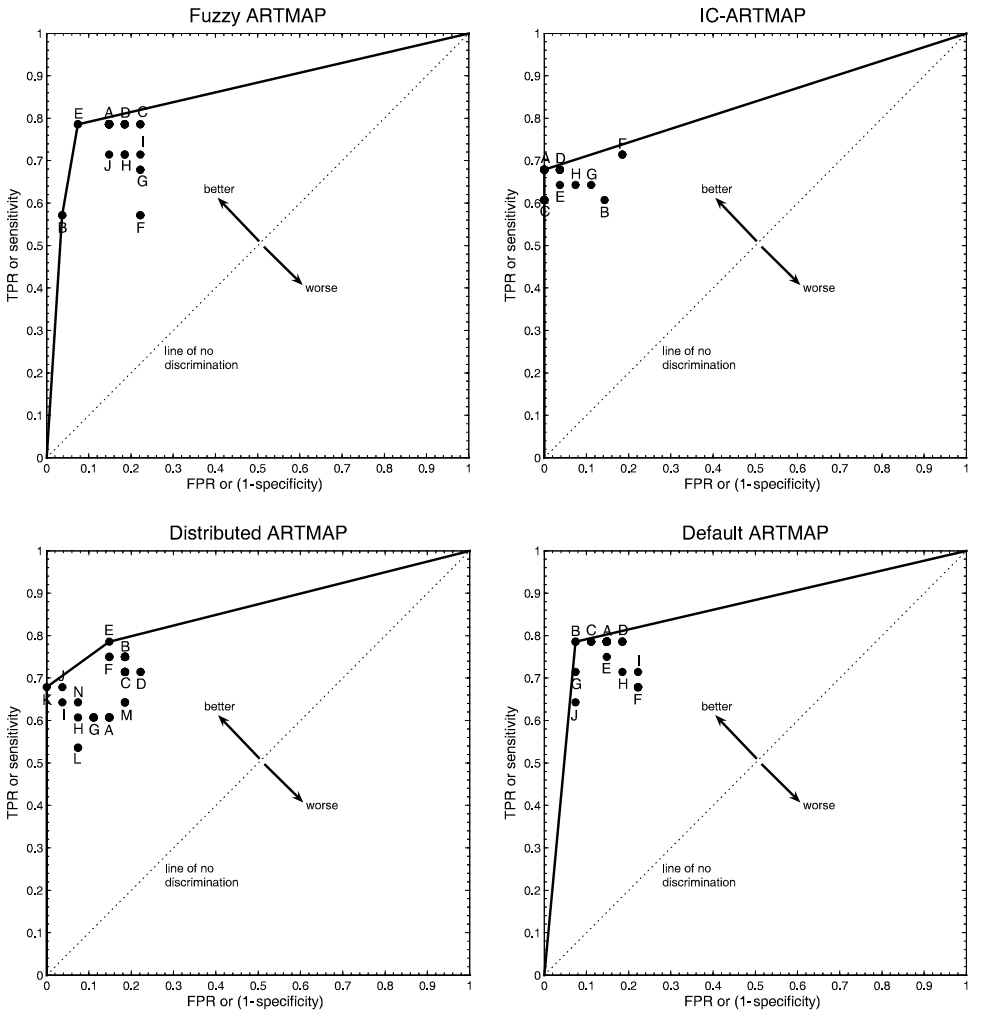

Fig. 4. ROC space for Fuzzy, IC, Distributed, and Default ARTMAP models. Points represent different classifiers associated with vigilance parameter values, as follows: Fuzzy: A{0.0–0.4; 0.475–0.525; 0.7; 0.825}, B{0.425–0.45}, C{0.55–0.575}, D{0.6–0.675; 0.75}, E{0.725; 0.775}, F{0.8; G: 0.825; 0.875}, H{0.9–0.925}, I{0.95; 1}, J{0.975}. IC: A{0.0–0.45; 0.5–0.575}, B{0.475}, C{0.6–0.75}, D{0.775–0.875}, E{0.9}, F{0.925–0.95}, G{0.975}. Distributed: A{0.0–0.15}, B{0.175–0.35}, C{0.375; 0.85–0.875; 0.925–0.95}, D{0.4–0.425}, E{0.45}, F{0.475–0.525}, G{0.55–0.65}, H{0.675}, I{0.7}, J{0.725}, K{0.75}, L{0.775–0.8}, M{0.825; 0.9; 0.975}. Default: A{0.0–0.4; 0.475–0.675}, B{0.425–0.45}, C{0.7–0.725}, D{0.75}, E{0.8}, F{0.825; 0.875–0.925}, G{0.85}, H{0.95}, I{0975}, J{1.0}.

A point in the space is better than another if it is to the northwest (TPR is higher, FPR is lower, or both) of the first. Classifiers appearing on the left-hand side near the x axis may be thought of as more 'conservative' as they make positive classifications only with strong evidence so they make few false positive errors.

Table 1. Misclassified patterns obtained by the four ARTMAP variants: fuzzy (FAM), distributed (DIS), IC (IC), and default (DEF), and the results obtained with the test data set by six different methods: linear discriminant analysis (LDA),[7] MPL,[7] MLP,[11] SLP,[11] Athena Neural Model (ANM),[11] and MLP.[12]

pattern	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28
LDA																	*	*							*			
SLP																	*	*			*				*			
MLP[7]												*					*	*			*				*			
MLP[11]																	*	*			*				*			
ANM												*					*	*							*			
MLP[12]																	*	*			*				*			
FAM																	*	*		*	*				*			*
DIS			*					*								*	*	*		*	*				*			*
IC								*					*				*	*		*	*			*	*			*
DEF																	*	*		*	*				*			*

pattern	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55
LDA							*	*			*	*						*	*		*	*	*			*	*
SLD											*	*						*			*	*				*	
MLP[7]											*	*						*			*	*				*	
MLP[11]							*	*			*	*						*			*					*	
ANM											*	*						*			*	*				*	
MLP[12]											*										*	*				*	
FAM							*																			*	
DIS																											
IC																											
DEF							*																			*	

Classifiers close to the upper right-hand side of an ROC graph may be thought of as more 'liberal' as they make positive classifications with weak evidence so they classify nearly all positives correctly, but they often have high false positive rates.

Given the points in the ROC space of the four ARTMAP models, we can construct ROC convex hull curves connecting the most northwest points (FPR, TPR) as well as the two trivial classifiers (0, 0) and (1, 1). This is possible, because given two classifiers we can construct any intermediate classifier just randomly weighting both classifiers (giving more or less weight to one or the other). This creates a continuum of classifiers between any two classifiers, which allow linking of the two points by a line. The convex hull has a number of useful implications. The classifiers below the convex hull curve can be discarded because there is no combination of class distribution/cost matrix for which they could be optimal. Since only the classifiers on the convex hull are potentially optimal, no others need to be retained. This allows to determine the candidates for optimal classifiers: for the fuzzy model — points B and E; IC — A and C; distributed — E and K; and the default model has a single candidate — B. Selection of the best (optimal) classifier from among candidates depends on the context of application, determined by the class distribution and the error cost (will be discussed in the following section).

4.2. *AUC results*

Table 2 shows the AUC values of the fuzzy, IC, distributed, and default ARTMAP models. The values show that the distributed ARTMAP is best performer, followed by fuzzy, default, and IC models. It should be pointed out that for crisp classifiers the AUC metric provides an overall estimation of the model performance and does not find optimal classifiers of the ROC convex hull.

4.3. *Cost analysis*

The classifier with lowest error rate is frequently not the best classifier. In many applications not all the errors produced by a predictor have the same consequences.

Table 2. Figure of merits for fuzzy, IC, distributed, and default ARTMAP models: AUC and unit cost of ROC convex hull (ROCCH) points. Letters in bold show the optimal classifier for a model.

Model	AUC	ROCCH points	Unit cost
Fuzzy ARTMAP	0.8624	**B**	0.786
		E	0.929
IC ARTMAP	0.8392	**A**	0.321
		C	0.393
Distrib. ARTMAP	0.8690	E	1.643
		K	0.321
Default ARTMAP	0.8558	**B**	0.929

Table 3. Cost-benefit matrix.

	Actual positive	Actual negative
Predicted positive	0	FPcost = X
Predicted negative	FNcost = αX	0

The important thing is not to obtain the classifier with fewer errors but the one with lowest cost. ROC graphs have been criticized because of their inability to handle example-specific costs as they are based on the assumption that the costs of all true positives and false positives are equivalent. Cost analysis requires transformation of the confusion matrix into cost–benefit matrix as shown in Table 3. It encounters cost of misclassifications, both false positive and false negative. In the table X is the lost investment caused by insolvency of a company; αX, the lost profit of investment in a solvent company, where α can be an investor's annual profit rate.

Using the cost matrix, we evaluated all points belonging to the convex hull curves, as they are candidates for optimal classifiers. To calculate unit cost for a classifier we first calculated the slope of that point (5), (6).

$$slope = (FPcost/FNcost) \times (Neg/Pos) \tag{6}$$

$$cost\_per\_unit = FNR + slope \times FPR. \tag{7}$$

Outcomes from calculation are presented in Table 2. As an optimal classifier minimizes the cost per unit, the best classifiers for the models are as follows: fuzzy — B; IC — A; Distributed — K; and default — B. That means that the ARTMAP models have best performance with vigilance parameter values associated with those points.

The analysis also helps to rank the four ARTMAP models. Table 2 shows that the two best classifiers are K of the distributed and A of the IC, followed by B of the fuzzy, and K of the default model. Comparing the results with those of the AUC metric, we can conclude that the distributed model is not only best performer after accurate tuning, but it also provides best overall performance. The IC model has poor overall performance, but after careful tuning it can show best result. The fuzzy model shows moderate results, which exceed those of the default ARTMAP model. These conclusions, however, are application-specific rather than general. The conclusions also demonstrate that different performance metrics, such as accuracy, AUC, and unit cost show different results.

4.4. *Further experiments*

A series of experiments sought to investigate if the four ARTMAP models are sensitive to outliers, i.e. data points that could be excluded because of inconsistency with the statistical nature of the bulk of the data. We marked data points as outliers

if their values are more than three times the standard deviation value away from the mean of the relevant variable. The four models were trained and tested without outliers and results showed no difference from those obtained by the experiments discussed before. The four ARTMAP models showed no sensitivity to the outliers in the context of the domain and data sets discussed here.

Other experiments also led to the conclusion that best values for certain network parameters are those proposed by Carpenter *et al.*,[30] namely: baseline vigilance $\rho_{\text{test}} = 0$; signal rule $\alpha = 0.01$; and learning fraction $\beta = 1.0$.

We also examined efficiency of the four in terms of train time, test time, and RAM used for the long-term memory of the networks. Results show that for all trainings and testings, a session time is less than 0.02 sec. Consumed computer memory was less than 2.9 kb. An explanation for the instance responsiveness is that the ARTMAP models feature one-pass learning. In contrast, the widely used MLP require multiple presentations (epochs) of the training set to learn. Some studies report that MLPs with similar size data sets achieve convergence after 1400 training iterations,[40] 100,000 iterations,[9] and 191,400 iterations over 24 hours.[6] These figures illustrate once again some of the advantages of the ARTMAP models.

5. Conclusions

Today, financial institutions are paying heavy price for their indiscriminate practices. Corporate bankruptcies have put many institutions on the brink of insolvency and many others have been merged with or acquired by other financial institutions. Decision-making problems in the area require efficient analytic tools and techniques, most of which involve machine learning in order to predict future financial status of firms.

Our research was motivated by a gap in the studies on bankruptcy prediction methods, namely using some still unexplored techniques based on the Adaptive Resonance Theory neural networks. Here we examine four of them — fuzzy ARTMAP, distributed ARTMAP, instance counting ARTMAP, and default ARTMAP. In order to illustrate the network performance and compare results from other techniques, we used data, financial ratios, and experimental conditions identical with those published in previous studies.

Our experiments show that financial ratios *Retained Earnings/Total Assets* and *Market Value of Equity/Book Value of Total Debt* provide highest discriminatory power and ensure best prediction accuracy for all the ARTMAP networks. We also found that with appropriate network parameters ARTMAP provides 85.5% accuracy, which outperforms all MLP networks and other classification techniques applied to the same data. In order to avoid bias from the prediction accuracy and estimate the overall classifier performance, we used fivefold cross-validation and exhaustive search of input variables, receiver operating characteristic analysis, and area under curve metric. The figures show that the distributed ARTMAP is best performer followed by fuzzy, default, and instance counting ARTMAP.

We also did application-specific cost analysis to find optimal network parameters. The unit cost metric shows that by proper tuning of the network vigilance, fuzzy ARTMAP appears to be best application-specific classifier followed by instant counting, distributed, and default ARTMAP. Our experiments also showed that the ARTMAP classifiers are not sensitive to outliers in the data set. The classifiers' efficiency in terms of train and test time was also confirmed experimentally.

In conclusion, we find that ARTMAP neural network is a promising technique for bankruptcy prediction that outperforms the most popular MLP networks not only in terms of prediction accuracy, but also as training time and adaptiveness in a changing environment.

References

1. A. Vellido, P. Lisboa and J. Vaughan, Neural networks in business: A survey of applications, *Expert Syst. Appl.* **17** (1999) 51–70.
2. P. Kumar and V. Ravi, Bankruptcy prediction in banks and firms via statistical and intelligent techniques, *Eur. J. Oper. Res.* **180**(1) (2007) 1–28.
3. H.-F. Wang and Z.-W. Chan, A pruning approach to pattern discovery, *Int. J. Inform. Technol. Decis. Making (IJITDM)* **7**(4) (2008) 721–736.
4. R. Lacher, P. Coats, S. Sharma and L. Fant, A neural network for classifying the financial health of a firm, *Eur. J. Oper. Res.* **85** (1995) 53–65.
5. Y. Peng, G. Kou, Y. Shi and Z. Chen, A descriptive framework for the field of data mining and knowledge discovery, *Int. J. Inform. Technol. Decisi. Making (IJITDM)* **7**(4) (2008) 639–682.
6. M. Odom and R. L. Sharda, A neural network model for bankruptcy prediction, in *Proc. IEEE Int. Conf. on Neural Networks* (San Diego, 1990), pp. 163–168.
7. M. Odom and R. Sharda, A neural network model for bankruptcy prediction, in *Neural Networks in Finance and Investing*, eds. R. R. Trippi and E. Turban (Probus Publishing Company, Chicago, 1993).
8. R. Sharda and R. Wilson, Performance comparison issues in neural network experiments for classification problems, in *Proc. 26th Hawai Int. Conf. System Scientists* (1993).
9. R. Wilson and R. Sharda, Bankruptcy prediction using neural networks, *Decis. Support Syst.* **11** (1994) 31–447.
10. E. Altman, Financial ratios, discriminant analysis, and the prediction of corporate bankruptcy, *J. Fin.* **23**(4) (1968) 598–609.
11. E. Rahimian, S. Singh, T. Thammachacote and R. Virmani, Bankruptcy prediction by neural networks, in *Neural Networks in Finance and Investing*, eds. R. Trippi and E. Turban (Probus Publ., Chicago, 1993).
12. C. Serrano-Cinca, Self organizing neural networks for financial diagnosis, *Decis. Support Syst.* **17** (1996) 227–238.
13. T. Bell, G. Ribar and J. R. Verchio, Neural nets vs. logistic regression: A comparison of each model's ability to predict commercial bank failures, in *Proc. 1990 Deloitte & Touche/University of Kansas Symposium on Auditing Problems* (1990), pp. 29–53.
14. A. Hart, Using neural networks for classification task. Some experiments on datasets and practical advice, *J. Oper. Res. Soc.* **43**(3) (1992) 215–266.
15. Y. Yoon, G. Swales and T. Margavio, A Comparison of discriminant analysis versus artificial neural networks, *J. Oper. Res. Soc.* **44**(1) (1993) 51–60.

16. S. P. Curram and J. Mingers, Neural networks, decision tree induction and discriminant analysis: An empirical comparison, *J. Oper. Res. Soc.* **45**(4) (1994) 440–450.
17. S. Salcedo-Sanz, J. Fernandez-Villacanas, M. Segovia-Vargas and C. Bousono-Calzon, Genetic programming for the prediction of insolvency in non-life insurance companies, *Comput. Oper. Res.* **32** (2005) 749–765.
18. K. Shin and Y. Lee, A genetic algorithm application in bankruptcy prediction modelling, *Expert Syst. Appl.* **23**(3) (2002) 321–328.
19. A. Cielen, L. Peeters and K. Vanhoof, Bankruptcy prediction using a data envelopment analysis, *Eur. J. Oper. Res.* **154**(2) (2004) 526–532.
20. D. Rumelhart and J. McClelland, *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, Vol. 1 (MIT Press, Cambridge, MA, 1986).
21. T. Kohonen, *Self-organizing Maps*, Series in Information Sciences, 2nd edn., Vol. 30 (Springer, Heidelberg, 1997).
22. T. Keiji and T. Tetsuzo, Improved projection Hopfield network for the quadratic assignment problem, *Int. J. Inform. Technol. Decis. Making (IJITDM)* **7**(1) (2008) 53–70.
23. J. Hopfield, Neural networks and physical systems with emergent collective computational abilities, *Proc. Natl. Acad. Sci.* **79** (1982) 2554–2558.
24. S. Grossberg, Adaptive pattern recognition and universal encoding II: Feedback, expectation, olfaction, and illusions, *Biol. Cybernet.* **23** (1976) 187–202.
25. G. Carpenter and S. Grossberg, A massively parallel architecture for a self-organizing neural pattern recognition machine, *Comput. Vis. Graph. Image Process.* **37** (1987a) 54–115.
26. G. Carpenter, Default ARTMAP, in *Proc. Int. Joint Conference on Neural Networks* (IJCNN'03) (2003).
27. A. Nachev and B. Stoyanov, Default ARTMAP neural networks for financial diagnosis, in *Proc. Int. Conf. Data Mining DMIN'07* (Las Vegas, 2007), pp. 1–9.
28. A. Nachev, Forecasting with ARTMAP-IC neural networks. An application using corporate bankruptcy data, in *Proc. 10th Int. Conf. on Enterprise Information Systems* (*ICEIS'08*) (Barcelona, 12–16 June, 2008), pp. 167–172.
29. S.-S. Park, K.-K. Seo and D.-S. Jang, Fuzzy art-based image clustering method for content-based image retrieval, *Int. J. Inform. Technol. Decis. Making (IJITDM)* **6**(2) (2007) 213–233.
30. G. Carpenter, S. Grossberg and J. Reynolds, ARTMAP: Supervised real-time learning and classification of non-stationary data by a self-organizing neural network, *Neural Netw.* **6** (1991a) 565–588.
31. E. Granger, A. Rubin, S. Grossberg and P. Lavoie, A what-and-where fusion neural network for recognition and tracking of multiple radar emitters, *Neural Netw.* **3** (2001) 325–344.
32. G. Carpenter, S. Grossberg, N. Markuzon, J. H. Reynorlds and D. B. Rosen, Fuzzy ARTMAP: Neural network architecture for incremental supervised learning of analog multidimensional maps, *IEEE Trans. Neural Netw.* **3**(5) (1992) 698–713.
33. G. Carpenter, S. Grossberg and D. Rosen, Fuzzy ART: Fast stable learning and categorization of analog patterns by an adaptive resonance system, *Neural Netw.* **4**(6) (1991b) 759–771.
34. G. Carpenter, Distributed activation, search, and learning by ART and ARTMAP neural networks, in *Proc. Int. Conf. Neural Networks* (*ICNN'96*) (1996), pp. 244–249.
35. G. Carpenter, Distributed learning, recognition, and prediction by ART and ARTMAP neural networks, *Neural Netw.* **10**(8) (1997) 1473–1494.

36. G. Carpenter and N. Markuzon, ARTMAP-IC and medical diagnosis: Instance counting and inconsistent cases, *Neural Netw.* **11**(2) (1998) 323–336.
37. E. F. Brigham and L. C. Gapenski, *Financial Management Theory and Practice* (Dryden Press, New York, 1991).
38. J. P. Li, Z. Chen, L.-W. Wei, W. Xu and K. Gang, Feature selection via least squares support feature machine, *Int. J. Inform. Technol. Decis. Making (IJITDM)* **6**(4) (2007) 671–686.
39. T. Fawcett, An introduction to ROC analysis, *Pattern Recogn. Lett.* **27**(8) (2006) 861–874.
40. P. Coats and L. Fant, Recognizing financial distress patterns using neural network tool, *Fin. Manage.* **November 1993** (1993) 142–155.
41. A. P. Bradley, The use of the area under the ROC curve in the evaluation of machine learning algorithms, *Pattern Recogn.* **30** (1997) 1145–1159.
42. W. Huang, K. K. Lai, Y. Nakamori, S. Wang and L. Yu, Neural networks in finance and economics forecasting, *Int. J. Inform. Technol. Decis. Making (IJITDM)* **6**(1) (2007) 113–140.