TUCS

Adrian Costea

Computational Intelligence Methods for Quantitative Data Mining

Turku Centre *for* Computer Science

TUCS Dissertations
No 67, November 2005

Computational Intelligence Methods for Quantitative Data Mining

Adrian Costea

DOCTORAL DISSERTATION

To be presented, with the permission of the Faculty of Economics and Social Sciences at Åbo Akademi University for public criticism in DataCity, entrance A, third floor, Auditorium 3102, on the 2nd of December 2005, at 10 o'clock a.m.

Åbo Akademi University
Turku Centre for Computer Science /
Institute for Advanced Management Systems Research
Lemminkäinengatan 14A, FIN-20520 Åbo, Finland

Åbo, 2005

Supervised by

Professor Barbro Back
Institute for Advanced Management Systems Research
Department of Information Systems
Åbo Akademi University
Turku, Finland

Reviewed by

Professor Petri Helo
Department of Electrical Engineering and Industrial Management
University of Vaasa
Vaasa, Finland

Assistant Professor (D. Sc. (Tech.)) Esa Alhoniemi
Department of Information Technology
University of Turku
Turku, Finland

Opponent

Professor Max Bramer
School of Computing
University of Portsmouth
Portsmouth, United Kingdom

ISBN 951-29-4013-2
ISSN 1239-1883

Painosalama Oy, 2005
Turku, Finland

To My Parents

Abstract

In this dissertation we investigate the use of different Computational Intelligence (CI) methods to address different business problems. The CI methods employed are from the field of artificial intelligence (decision tree induction – DT, neural networks – ANNs – in the form of self-organizing maps – SOMs – and multilayer perceptrons – MLPs), evolutionary computation (genetic algorithms – GAs) and fuzzy logic (Fuzzy C-Means – FCM). Classical statistical methods (e.g. C-Means, multinomial logistic regression – MLR) are used as comparison methods.

The business problems can be matched with different data-mining (DM) tasks such as clustering, classification and regression. For example, if we simplify, *assessing comparatively economic/financial performance of countries/companies* can be matched with a combination of data-mining clustering and classification task, and *prediction of process control variables* corresponds to the DM regression task.

The dissertation contributes to the related research by exploring and combining (e.g. building *hybrid* systems) the above methods for performing DM tasks. For the clustering task we compare and explore different methods such as SOM, C-Means, FCM and Weighting FCM. We address the problem of validating SOM topology and quantisation error. We introduce linguistic variables to automatically characterize each cluster. For the classification task we introduce a standard method to compare the different approaches such as MLR (statistical method), DT, and ANN (CI methods). We find the most adequate hybrid classification model for the experiment in question. Moreover, we address different technical problems related to the different classification techniques. We extend the applicability of ANNs for the data-mining regression task by applying a retraining procedure for learning the connection weights of the ANN.

As experiments for our study we studied the economic performance of certain Central and Eastern European countries, companies from two important worldwide industrial sectors, telecommunications and pulp and paper manufacturing, with respect to their financial performance and the glass manufacturing process at Schott, a German-based company. We identified groups with similar economic-/financial performance and showed how the countries/companies evolved over time. In the first experiment we analysed Romania's and Poland's economic performance between 1996 and 2000. Overall, Romania was unstable with respect to all the economic variables. Poland, on the other hand, had a stable economic performance that led to EU membership in 2004. The Ukraine had progressed steadily between 1993 and 2000 with respect to its foreign trade balance. In the pulp-and-paper experiment we benchmarked the best three Finnish companies, UPM Kymmene being the best performer. In the telecom experiment we benchmarked the Scandinavian telecom companies and the four largest telecom companies, Nokia achieving the best result. In the last experiment we predicted the temperatures of a Schott glass-smelting tank. Decision-makers, creditors and investors can benefit from this kind of analysis.

Keywords: Business Intelligence, Computational Intelligence Methods, Knowledge Discovery, Data Mining, Economic/Financial Performance Benchmarking, Prediction of Process Control Variables

Acknowledgements

This dissertation has been the result of my research conducted over the last five years at Turku Centre for Computer Science (TUCS). There are many people who have supported me and to whom I owe special thanks.

First, I would like to thank my supervisor Professor Barbro Back for her endless support and patience. Without her continuous encouragement, I would have never had the resources to write this dissertation. She has always shown great care and provided me with guidance throughout my research studies. The discussions that we have had have always been valuable and constructive.

I want to thank the reviewers of my dissertation, Professor Petri Helo from University of Vaasa and Assistant Professor Esa Alhoniemi from University of Turku for their useful comments and suggestions that greatly improved the quality of the manuscript. I wish to thank Christopher Grapes for correcting the language of this dissertation. I am grateful to Professor Max Bramer from Portsmouth University for accepting the invitation to be my opponent.

Special thanks are due to TUCS for the excellent working conditions and financial support provided during my doctoral studies. I am extremely grateful to Professor Christer Carlsson, the head of the Institute for Advanced Management Systems Research (IAMSR), the place where I have studied and done my research. Other institutions have also supported this work financially. I gratefully acknowledge the support of Academy of Finland, Finnish Technology Center, and Tekes. I also gratefully acknowledge the grant provided by Åbo Akademi University. I would like to thank the administrative staff, Monika and Irmeli (TUCS) and Leena, Sirpa, Stina, Pia, and Anne (IAMSR) for their help and support.

I have benefited from working with many excellent co-authors. For this, I would like to thank Francisco Alcaraz, Tomas Eklund, Jonas Karlsson, Antonina Kloptchenko (Durfee), and Iulian Nastac.

I would also want to thank my colleagues, Tomas Eklund, Marketta Hiissa, Piia Hirkman, and Dorina Marghescu for their valuable comments and suggestions in reading the manuscript. Tomas also kindly shared the pulp-and-paper dataset, provided ideas and co-authored with me a number of papers. His perfect knowledge of English language helped improving the quality of our papers. I would like to thank all my colleagues from IAMSR for the enjoyable break times that we shared together.

I am thankful to all my friends in Turku for their moral support and the nice times that we spent together. In particular, Viorel has been my guide in many important decisions that I had faced. He and Professor Ion Ivan from the Academy of Economic Studies in Bucharest were the first persons to suggest me to apply for a TUCS postgraduate position. Alex, Aurel, Catalin, Cristina, Diana, Dragos, Dudu, Irina, Irinel, Marius, Ovidiu, Tibi, and Zoli have been there every time I needed a second opinion about anything. I would especially like to thank Anas Lanedri for his friendship during my first years in Turku. Anas has always provided me with strong motivation to finish my doctoral studies.

Last but not least, I would like to thank my family – my parents, Costea and Joita, my brother, George for their constant support during all these years. Finally, I would like to thank my fiancé, Mirela for her patience, love, and dedication.

Turku, October 5, 2005

Table of Contents

List of Original Research Publications

1 Kloptchenko A, Eklund T, Costea A, Back B. 2003. A Conceptual Model for a Multiagent Knowledge Building System. In *Proceedings of 5th International Conference on Enterprise Information Systems (ICEIS 2003)*, Camp O, Filipe J, Hammoudi S, Piattini M. (eds.), published by Escola Superior de Tecnologia do Instituto Politécnico de Setúbal, Angers, France, April 23-26, 2003, Volume 2: Artificial Intelligence and Decision Support Systems, pp. 223-228. ISBN: 972-98816-1-8.

2 Costea A, Eklund T. 2003. A Two-Level Approach to Making Class Predictions. In *Proceedings of 36th Annual Hawaii International Conference on System Sciences (HICSS 2003)*, Sprague Jr RH. (ed.), IEEE Computer Society, Hawaii, USA, January 6-9, 2003, Track: Decision Technologies for Management, Minitrack: Intelligent Systems and Soft Computing. ISBN: 0-7695-1874-5/03.

3 Costea A, Eklund T. 2004. Combining Clustering and Classification Techniques for Financial Performance Analysis. In *Proceedings of 8th World Multi-Conference on Systemics, Cybernetics and Informatics (SCI 2004)*, Callaos *et al*. (eds.), Organized by IIIS, Orlando, Florida, USA, July 18-21, 2004, Volume I: Information Systems, Technologies and Applications, Track: Management Information Systems, pp. 389-394. ISBN: 980-6560-13-2.

4 Alcaraz-Garcia AF, Costea A. 2004. A Weighting FCM Algorithm for Clusterization of Companies as to their Financial Performances. In *Proceedings of the IEEE 4th International Conference on Intelligent Systems Design and Applications (ISDA 2004)*, Rudas I. (ed.), CD-ROM Edition, Budapest, Hungary, August 26-28, 2004, Track: Intelligent Business, pp. 589-594. ISBN: 963-7154-30-2.

5 Costea A, Nastac I. 200x. Assessing the Predictive Performance of ANN-based Classifiers Based on Different Data Preprocessing Methods, Distributions and Training Mechanisms. Submitted to the *International Journal of Intelligent Systems in Accounting, Finance and Management* (in review process).

6 Nastac I, Costea A. 2004. A Retraining Neural Network Technique for Glass Manufacturing Data Forecasting. In *Proceedings of 2004 IEEE International Joint Conference on Neural Networks (IJCNN 2004)*, IEEE, Budapest, Hungary, July 25-29, 2004, Volume 4, Track: Time Series Analysis, pp. 2753-2758. ISBN: 0-7803-8359-1.

List of Figures

List of Tables

List of Abbreviations

A	Average
ANN	Artificial Neural Network
AR	Analytical Review
BI	Business Intelligence
BLR	Binary Logistic Regression
BP	Backpropagation
CART	Classification And Regression Trees
CDF	Cumulative Distribution Function
CI	Computational Intelligence
CR	Current Ratio
CVI	Competitive Intelligence
DA	Discriminant Analysis
DBMS	Database Management System
DM	Data Mining
DT	Decision Tree(s)
EA	Evolutionary Algorithm
EC	Equity to Capital
EP	Evolutionary Programming
ES	Evolutionary Strategy
EUNITE	European Network on Intelligent Technologies
FCM	Fuzzy C-Means
FIFO	First In First Out
FILM	Fuzzy Inductive Learning Algorithm
GA	Genetic Algorithm
GCI	Growth Competitiveness Index
GD	Gradient Descent
GNDP	Gross National Domestic Product
H	High
HCM	Hard C-Means
IC	Interest Coverage
IS	Information Systems
ISDA	Information System Development Approach
ISDM	Information System Development Methodology
IT	Information Technology
KDD	Knowledge Discovery in Databases
KM	Knowledge Management
L	Low
LDA	Linear Discriminant Analysis
LIFO	Last In First Out
LL	Log Likelihood
LM	Levenberg-Marquardt
LR	Logistic Regression
MDA	Multivariate Discriminant Analysis
MLE	Maximum Likelihood Estimation
MLP	Multilayer Perceptron

MLR	Multinomial Logistic Regression
OLAP	Online Analytical Processing
OLS	Ordinary Least Squares
OM	Operating Margin
PCA	Principal Component Analysis
QR	Quick Ratio
ROE	Return On Equity
ROTA	Return On Total Assets
RP	Resilient Backpropagation
RPA	Recursive Partitioning Algorithm
RT	Receivables Turnover
RT-based ANN	Retraining-based ANN
S&P	Standard & Poor's
SA	Simulated Annealing
SCG	Scale Conjugate Gradient
SOFM	Self-Organizing Feature Map
SOM	Self-Organizing Map
TDIDT	Top-Down Induction of Decision Trees
TM	Text Mining
VH	Very High
VL	Very Low
WFCM	Weighting FCM
XBRL	eXtensible Business Reporting Language
XML	eXtensible Markup Language

List of Notations

$x(t)$	t^{th} sample in a time series
n	the number of: inputs/observations/subsets/elements of the input delay vector/distinct intervals
m_i	i^{th} weight vector
x	an input observation (sample)
m_c	the best matching unit
$\alpha(t)$	learning rate function
N_c	a set of neurons in the vicinity of the winner
$\alpha(0)$	initial learning rate
C	a constant/the number of clusters/the number of subsets/a class
$rlen$	training length
$N(t)$	neighbourhood function
$N(0)$	initial radius length
qe	average quantisation error
x_i	i^{th} input vector/observation
$m_c^{(i)}$	the closest weight vector for the input vector x_i
c_j, v_i	centre of cluster j, i
X	a (fuzzy) set
A	a (fuzzy) subset
χ_A	the characteristic function of the subset A
μ_A	the membership function of the fuzzy subset A
U	the matrix of membership values
V	the matrix which contains the centres of the clusters
$J(U, v)$	the objective function of the Fuzzy C-Means algorithm
u_{ik}	the membership degree of observation x_k in cluster i
C	the number of clusters/the number of classes/a confidence factor
d_{ik}	the Euclidean distance between observation x_k and cluster centre v_i
P	number of input variables
m	FCM weighting exponent/minimum cases for a leaf node/number of elements of the output delay vector
ε	small positive number – used in stopping criterion for FCM
i, j, k, l, s	indices used in different contexts
$perc_{ij}$	the vector of percentages of input variable j in cluster i
VC_{ij}	the variation coefficient of $perc_{ij}$
SVC_{ij}	the standardised variation coefficient of $perc_{ij}$
$\sigma(perc_{ij})$	the standard deviation of $perc_{ij}$
$\overline{perc_{ij}}$	mean of $perc_{ij}$
I	set of “certain” observations
I'	set of “uncertain” observations
$odds(x \in C)$	the likelihood that observation x belongs to class C
w_i	i^{th} regression coefficient
v_i	i^{th} input variable/cluster centre
$logit(odds\ ratio)$	logistic transformation of *odds ratio*
$R^2_{Cox\ and\ Snell}$, $R^2_{Nagelkerke}$	correlation measures for logistic regression
E	the base of natural logarithm
X	a test/horizontal dimensionality of the SOM/an eigenvector
Y	vertical dimensionality of the SOM
T	A training set
T_i	i^{th} training subset

$gain(X)$, $gain_{Gini}(X)$	information gain of splitting the data based on text X based on the information gain criterion
$info(T)$, $info_{Gini}(X)$	expected amount of information to specify the class of a case in T
$info_X(T)$	expected amount of information to specify the class of a case in T given that the case reaches the node with test X
$freq(C_j, T)$	the frequency of cases which belong to the class C_j in the set T
$\lvert T \rvert$	the number of observations (cases) in T
$entropy(p_1, p_2, ..., p_n)$	the entropy function
$gain\ ratio(X)$	the information gain of splitting the data based on text X based on the gain ratio criterion
$split\ info(X)$	the potential information of dividing T into n subsets
θ	a threshold (a constant value)
NH	number of hidden neurons
NI	number of input neurons
NO	number of output neurons
TR	training set
TRe	effective training set
TS	test set
VAL	validation set
N_{TRN}	number of training samples
N_w	number of weights
NH_i	number of neurons for the i^{th} hidden layer
L	number of runs of the experiment
γ	scaling factor
RTi	i^{th} retraining mechanism
ACR_A	accuracy rate of set A
MSE_A	mean square error of set A
N_{gen}	number of generations for the genetic algorithm
PS, PS', PS''	population size
N_{elite}	number of elite chromosomes
Max_split	maximum number of splitting points
P_i	the probability of selection for the chromosome i
P_c	the probability of crossover
P_m	the probability of mutation
α	a randomly generated number between 0 and 1
max_lim	maximum number of duplicate chromosomes
λ	eigenvalue
$prepca$	Matlab PCA transformation function
$transMat$	PCA transformation matrix
min_frac	minimum fraction of the total variation in the data set – the criterion of retaining the eigenvectors in the PCA transformation function
$O_i(t)$, $output\_i(t)$, $Y(t)$	output i at moment t
$I_i(t)$, $input\_i(t)$, $X(t)$	input i at moment t
$Vect\_In$	the delay vector for the inputs
$Vect\_Out$	the delay vector for the outputs
ERR	EUNITE competition model evaluation error
O_{Rki}	the real output i at time step k
O_{Fki}	the forecasted output i at time step k
$f(k)$	a weight decreasing function with the number of time step k
N	number of observations
ERR_A	the average training error
ERR_T	the test error

PART ONE: RESEARCH SUMMARY

Chapter 1 Introduction

1.1 Research Context

More and more the international business environment is being considered a war zone in which the most important weapon is *fore*knowledge (McNeilly, 2000). McNeilly's *Sun Tzu and the Art of Business* is one of the books that convert ancient war experience tactics into business rules and advice. Indeed, increasingly attention is being paid to the business intelligence units within organisations.

Business intelligence (BI) is the process of legally and ethically obtaining and analysing raw data in order to make relevant observations about a company's competitive environment, and to determine actionable, strategic and tactical options (Maag & Flint, 2004). Maag & Flint (2004, p. 404) state that within the broader knowledge management (KM) process, BI can be metaphorically considered as the keyboard of the KM piano. As keys, BI comprises: data mining (DM), qualitative and quantitative market research, competitive intelligence (CVI), functions that are not mutually exclusive. For Maag & Flint data mining focuses on the past because it relies on historical data; market research (the process of planning, collecting, and analysing data relevant to marketing decision-making) is centred on the present, not necessarily the future, while BI as a whole confronts decisions involving the future by gathering competitive intelligence. For other authors (e.g. Barth, 2004) BI and CVI are two terms describing the same concept. For Barth CVI is information not only about competitors "but about any factor in the market environment that could impact your competitiveness as a business". Among these factors, the author mentions: partners, investors, employees, suppliers, customers, government regulators, and critics, as well as competitors. To refer specifically to competitors, the author uses the concept *competitor intelligence*. Whatever it is named, increasing number of managers understand the need for an intelligence unit within their organisations.

The importance of such an intelligence unit within organisations is obvious and comes from its goal, i.e. to cut costs, to save time and money for the organisations. As a consequence of this importance, almost all of the best ranked 500 companies by *Fortune* magazine (Fortune, 2005) employed a BI or CVI function internally. At the 2001 conference of the Society of Competitive Intelligence Professionals (SCIP) many companies were represented: Philip Morris, Bayer, Pfizer, Coca-Cola, Nestle, etc (Maag & Flint, 2004). There is an increasing understanding that old and traditional ways of gathering, analysing and presenting data have to be changed. Kolb (2000) cites a 1998 Future Group survey that suggested that by the end of 2001 about 60 per cent of Fortune's 1000 companies will establish and organise their BI units, having as main targets the markets, competitors, the technology that they use, and their products and services. At the time of the survey nearly 82 per cent of US companies with annual revenues of at least 10 billion and

60 per cent of companies with revenues of at least 1 billion had organised a CVI or BI unit.

The time factor is also of extreme importance when it comes to making decisions. For example, there was and still is a huge debate about the CIA's responsibility in preventing the 9/11 terrorist attacks in New York. At the same time, even though intelligence is gathered about some particular event/entity, the way it is presented, and the paths of sharing that information (i.e. the dissemination of information) has to be clearly defined from the bottom up to the head of the organisation. Figure 1-1 depicts the CIA's Intelligence cycle.

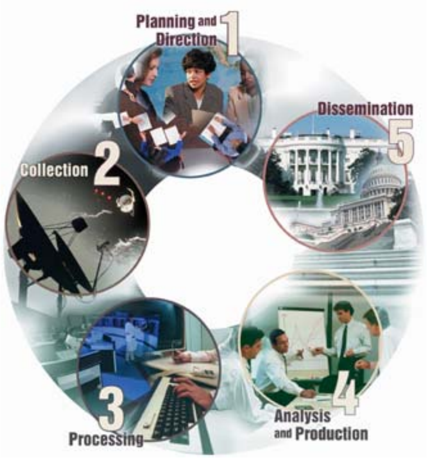


Figure 1-1 The CIA's Intelligence Cycle
(Source: Central Intelligence Agency, 2005)

As in the case of the CIA, the BI units within organisations have to perform the standard plan – collect – process – analyse – disseminate stages of the information/intelligence cycle. Analysing and disseminating the information will benefit customers, managers, suppliers, workers, partners, shareholders, etc. While all stages have their relative importance, the one that really has an impact is the analysis part of the process. Data-mining techniques (Klösgen & Zytkow, 2002) deserve close attention in this respect. Fayyad *et al*. (1996a, b) state that data mining (DM) is a particular step in the process of knowledge discovery in databases (KDD). Fayad *et al*. (1996a, b) define the KDD process as a set of various activities for making sense of data. The KDD process includes a number of steps such as:

- developing an understanding of the application domain and identifying the goal of the KDD process,
- creating the target dataset (data preparation, data selection),
- data cleaning and pre-processing,
- data reduction and projection,
- matching the goal of the KDD process with the data-mining task (summarisation, classification, regression, clustering),

- choosing the data-mining algorithm to perform the tasks,
- the effective data mining,
- interpreting the patterns and evaluation of results, and
- consolidating/reporting the discovered knowledge (See Chapter 3).

Concerning the nature of the steps undertaken, the KDD process is similar to BI information/intelligence cycle.

1.2 Motivation for the Study

Regardless of what the intelligence function within an organisation is called (KM, BI, CVI, or KDD), there is a need for methods and tools to fulfil this function. The BI unit has to be able to answer top management questions in an accurate and timely fashion. The managers are confronted with business problems such as *assessing comparatively the economic performance of countries* in which they want to invest, *assessing the financial performance of competitor companies*, *predictions of control variables* of their internal processes. The managers' job is to formulate the questions related to the business problems, whereas BI units have to find the answers. Sometimes, the managers are interested in expanding their business in new countries. One good example is the choice that Western European countries face when they want to invest in Central-Eastern Europe in the former communist countries. Therefore, the managers have to assess the countries' economic performance and find answers to questions such as "Which is the EU candidate or newly-accepted country that offers the best investment opportunities?", or "Which country should we invest in?". For financial performance benchmarking purposes there can be questions such as "Who are currently the five best performers in our area of business and what are they good at?", "Who are the most efficient competitors?" "How would you position our liquidity compared to our competitors?", "How about our profitability?". Another problem faced by managers is to make short-term predictions for various variables related to their internal processes. One question related to this issue might be: "What is the forecasting interval in which this process variable will have value in the near future?" The answers to these questions are not straightforward. In the process of answering these questions BI units transform data to information and knowledge following the KDD process. There are two concerns here: firstly, how to acquire the data that would later be transformed into knowledge, and secondly, how to effectively (i.e. what methods should be used) transform the data into useful knowledge.

One way to collect the necessary data is from publicly available sources. Among public sources, the Internet is of particular importance because of its electronic support, widespread availability, fast growing pace and ability to automate data retrieval. While not the only means of reliance for decision support, the information publicly available is worth investigating. There is great interest among business players in new ways of creating knowledge out of the huge amount of data that is now publicly available. However, there are two problems with regard to this data: information (data) overload and data usefulness. The second problem (data usefulness) is closely related to the second concern of how to efficiently obtain useful knowledge.

There are methods that address all the business problems mentioned above. But, traditional statistical methods used to collect, clean, store, transform, and analyse the data, while still in place and useful, need to be challenged. This challenge is provided by so-called Computational-Intelligence (CI) methods such as *machine learning*, *neural networks*, *evolutionary computation* and *fuzzy logic*.

The KDD process and its engine, DM, represent the umbrella under which the CI methods operate. Each business problem (real-world application) can be matched by many data-mining tasks depending on how we approach the problem. We match our real-world applications with the DM tasks as follows: countries'/companies' economic/financial performance benchmarking is matched with both DM clustering and classification tasks and prediction of process control variables is matched with the DM regression task. The CI methods are used to address the business problems through the aforementioned DM tasks. There are numerous CI methods available in the scientific literature with which we could perform the different DM tasks mentioned above (DM clustering, classification and regression tasks). However, in this thesis we restrict the number of CI methods used to perform the DM tasks as it would be unfeasible to test all possible solutions (methods). This is in line with Hevner *et al.*'s (2004) sixth guideline for design science research (see Section 2.3).

1.2.1 Countries'/Companies' Economic/Financial Performance Benchmarking

From decision makers to creditors and investors, for all business players one common problem is to obtain accurate and timely information about the economic/financial performance of an entity (country/company).

Decision makers are interested in what the strengths and the weaknesses of their entity are, and how the decision-making process can be influenced so that poor performance or, worse, bankruptcy, is avoided. At the same time, the relative position of their entity against the other entities is of great interest. The traditional way of doing things in the areas of performance benchmarking has became obsolete, in part because of the huge amount of information to be analysed, cheaper than ever IT equipment and the re-orientation of management toward non-conventional data analysis methods. For benchmarking purposes, companies are still using basic data collection methods, raw data calculations and spreadsheets. Using ordinary spreadsheet programs, one can easily compare two to six companies at a time according to one ratio at a time. However, if one wants to obtain an overview of the competitors on the market, or wants to take into account several ratios at the same time, spreadsheet programs are difficult to use (Eklund *et al.*, 2003).

Creditors such as banks or other credit institutions have many choices (in terms of the number of countries/companies) when they decide to invest their money. They are interested in the long-term payment ability of the borrowers or how solvent the borrowers are. In the process of credit risk assessment (e.g. Schaeffer, 2000), banks rely on internal ratings or, sometimes, they use the ratings provided by specialised agencies. The problem with internal ratings is, of course, the subjective aspect of the prediction, which makes it difficult to make consistent estimates (Atiya, 2001). A comprehensive study of internal ratings in large US banks and how the rating process is conceptualised, designed, operated and used in risk management can be found in Treacy & Carey (1998).

In general, when *investors* lend money to governments/companies it is in the form of bonds (that is a freely tradable loan issued by the borrowing company) (Tan *et al.*, 2002). The buyers of the bonds (investors) have to make an assessment of the creditworthiness of the issuing country/company, to gain an overall picture of what are the least risky countries/companies to invest in. Most bond buyers do not have the resources to perform this type of difficult and time-consuming research. Usually, this analysis is performed by so-called rating agencies (e.g. Standard and Poor's, Moody's) which assign a performance rank to each country from a specific geo-political area or to each company in a particular industry sector. In his website www.default*risk*.com, Hupton presents a comprehensive list of 75 rating agencies from around the world. The goal of these rating agencies is to provide the users with timely information on entities' performance. However, there are some drawbacks when making decisions based on these ratings. For example, rating agencies adjust their ratings only when it is unlikely that the ratings will be reversed shortly afterwards (Löffler, *in press*). In other words, the ratings tend to be reactive rather than predictive (Atiya, 2001). The rating process is very complex and opaque to the users. Some countries/companies that can be of interest for one investor are not rated or are rated differently by different agencies.

The business players rely on their BI units to help them address such business problems. To perform *economic/financial performance benchmarking* the BI units need to know the available tools and to choose the best ones, or maybe combine them (e.g. create a hybrid system which is better than any single method). In an attempt to determine the degree of use of advanced methods in performance benchmarking Eklund *et al.* (2004) found out that business people use mostly newspapers (76.32% daily) and the Internet (47.37% daily) to obtain financial information about competitors. The satisfaction with current methods was as follows: 50% were satisfied (15.79% dissatisfied) with the content provided through current methods, while the satisfaction with accuracy was lower, but still high (Eklund *et al.* 2004, p. 7). In other words there is still room for improvements in using CI methods in performance benchmarking.

As a conclusion, *the BI units need to construct accurate economic/financial performance benchmarking models using CI methods that will position* *countries/companies with respect to their economic/financial performance and which will benefit all business players.*

1.2.2 Prediction of Process Control Variables

A BI unit can also face problems regarding company's production processes. ERP (Enterprise Resource Planning) systems produce a huge amount of data that can be analysed and used to predict future outcomes and improve internal processes. Business players are interested in finding good models that can detect the correlations and autocorrelations among data and make short-term predictions of process control variables. This problem is a time series prediction problem. A time series is a vector of observations gathered successively at uniformly distributed time intervals. Time series prediction involves using a model to predict future observations before they are measured based on past values. Once again CI methods, particularly artificial neural networks (ANNs), represent an important alternative against traditional time series prediction methods such as statistical and econometric models and human judgmental methods. Constructing reliable time series models is challenging because of short data series, high noise levels, non-stationarities, and non-linear effects (Moody, 1995). ANNs are so-called "black-box" models in which there is no a priori information available. In contrast, the "white-box" model is a system where all the information necessary is available. Usually, the a priori information comes in the form of knowing the type of functions relating to different variables (Wikipedia, 2005). Moody (1995) proposes ANN as a tool for macroeconomic forecasting and concludes that "relative to conventional linear time series and regression methods, superior performance can be obtained using state-of-the-art neural networks models". Hand (2002, pg. 639) is concerned with the non-linear nature of the process data: "A program that searches for a good linear regression fit will throw up the best such fit but will not reveal the fact that the data are non-linearly related". Traditional statistical and econometric models have requirements with regard to underlying data and error distributions. For example, ARIMA[1] models assume that the standard deviation of the variable being modelled is constant over time. Also, the time factor is crucial: the analyses and model building have to be done fast. There is no point in predicting the outcome of a process variable for the next day several days after. From this perspective data-mining techniques, especially ANNs, can help: they are fast and can handle a huge amount of information at a time (Klösgen & Zytkow, 2002).

The use of ANN and other data-mining techniques for predicting process variables is preferred when one is not only concerned with model building but, also seeks unexpected information, interesting patterns, and anomalies in the dataset. In addition, "black-box" models (ANNs) are preferred when a priori information is not available.

[1] Autoregressive Integrative Moving Average

Consequently, *BI units need to explore CI methods, particularly ANNs models for process variables predictions*.

1.3 Aim of the Study and Research Questions

The first goal of the dissertation is to investigate the benefits of introducing CI methods to address business problems such as countries'/companies' economic/financial performance benchmarking. Our goal is to build *hybrid* economic/financial classification models as an alternative to the rating agencies' performance ranking models. The final goal of any classification model is to find a function or method with which unseen cases (observations) can be labelled. If the class variable (the output or the variable that gives the economic/financial performance class for each record in the dataset) is known, we use the *supervised* learning method. If we do not have the output, the learning is *unsupervised* or *clustering*. There can be a *mixture* of the two learning mechanisms as well: one can construct the class variable with an unsupervised method and then apply the supervised learning algorithm to build the classifier. We call such a mixture a *hybrid* classifier. These *hybrid* classification models have not only descriptive characteristics, but also prescriptive ones: we are now able to position in the clusters newly-observed cases without having to perform again any clustering experiment.

To fulfil our goal we explore, combine, improve and compare different methods. We explore and compare the performance of SOM clustering, C-Means, fuzzy C-Means algorithm (FCM) and a new clustering algorithm that we propose called Weighting FCM (WFCM). Next, we use the result of the clustering to build the class variable, and then, we build performance classification models (*hybrid* classifiers). Classification models have been around for nearly 40 years. There are two main groups of classification models: those that belong to the *knowledge-driven approach* and those that belong to the *data-driven approach* (Kaymak and van den Berg, 2004). Knowledge-driven approaches use prior domain knowledge to improve function approximation, while data-driven approaches are based on limited domain knowledge relying on the data at hand. Our study explores four data-driven approaches to classification problems: statistical approaches (multinomial logistic regression – MLR), induction approaches (Quinlan's C4.5, C5.0 decision tree algorithms – DT), neural approaches (multilayer perceptrons – MLPs and retraining-based ANN, which is a new way of training an ANN based on its past training experience and weights reduction – RT-based ANN), evolutionary approaches (the genetic algorithm used here as an alternative way of learning the connection weights of an ANN – GA-based ANN). As we mentioned in the previous section, in this dissertation we concentrate on a carefully selected subset of methods in performing the DM clustering and classification tasks. We are interested in providing guidelines for how one can use and compare the selected CI methods in performing the DM tasks.

We apply our models to analyse the Central-Eastern European countries in terms of their economic performance and the telecom and pulp-and-paper companies as to their financial performance.

The second goal of the dissertation is to explore the use of ANNs for process variables predictions. Our main aim here is to build a general ANN model that can support all sorts of processes within organisations. Industrial processes have certain characteristics which make them difficult to model: there is a huge amount of inter-correlated data available and there are time delays between the process inputs and outputs. As in the case of other industry processes, in the case of glass manufacturing, changing an input variable may result in an output change that starts only a couple of hours later and goes on for up to several days (EUNITE Competition, 2003). At the same time we have to explore new ways to un-correlate and reduce the input space and determine the optimal/sub-optimal ANN architecture.

For this purpose we propose a forecasting mechanism consisting of a retraining-based ANN model. We validate our forecasting model by applying it, as an experimental study, to predict some output variables of the glass manufacturing process at Schott, a German glass manufacturing company.

In order to fulfil our goals and in accordance with the seventh guideline for design science (Hevner *et al*., 2004 – see Section 2.2) we formulate two main research questions: one that is intended for management-oriented audiences and the other for technology-oriented ones.

1. *How could CI methods be used to construct business models with which business problems such as benchmarking countries'/companies' economic/financial performance and predicting the control variables of internal processes could be addressed?*

2. *What technical problems need to be considered when constructing these business models?*

In order to answer the main research questions we divide them by posing and answering a number of sub-questions.

We pose three main management-oriented research sub-questions related to the three business problems addressed.

- How could CI methods best support business players in choosing the countries in which they would like to invest to extend their businesses?
- How could CI methods best support business players in performing financial performance benchmarking against their competitors?
- How could CI methods best support business players in improving their internal production processes?

Some of the technical research sub-questions (related to the CI methods used) are:

For the clustering task:

- Find, for a particular clustering task, the most adequate clustering method in terms of pattern allocation and explanatory power.
- Find an objective method to *automatically* characterise the financial performance clusters.

For the classification task

- Define a standard procedure to compare different approaches for classification.
- Find, for a particular classification task, the most adequate hybrid classification model in terms of accuracy rate (the rate of correctly classifying the observations) and class predictions.
- Validate the performance of the *hybrid* classifiers.

For the regression task:

- How does the retraining procedure improve the ANN performance when performing regression tasks?

General:

- Detect outliers in data and find the best pre-processing method for a particular case.
- Elaborate an empirical procedure for determining the ANN architecture.

1.4 Related Work and Relevance of the Study

There are many research papers that have applied CI methods in economic/financial performance benchmarking, process variables prediction and, in general, to solve business problems (Deboeck, 1998; O'Leary, 1998; Wong & Selvi, 1998; Li, 1994; Widrow *et al*., 1994). Moreover, in scientific literature one can find separately, for each of all three business problems mentioned above, research related to the one presented in this dissertation (see Chapter 4).

We position our research using the conceptual model of the multiagent knowledge building system[2] presented in Section 3.3. The model integrates data and text mining methods and is based on a society of software agents, each of which carries out its own functions and uses information provided by other agents connected to it. In this dissertation we explore the CI methods for performing the *quantitative* data-mining tasks of the knowledge building system.

[2] The work within this dissertation has been conducted in the Data-Mining and Knowledge Discovery Laboratory at Turku Centre for Computer Science and is a piece of the entire research project of constructing a multiagent knowledge building system, project which has been supervised by Prof. Barbro Back and has been conducted with the help of the following PhD students: Adrian Costea, Tomas Eklund, Piia Hirkman, Jonas Karlsson, Antonina Kloptchenko, Aapo Länsiluoto, and Dorina Marghescu.

The roots of our interest in using clustering techniques (e.g. SOM) for performance benchmarking come from a number of studies such as Martín-del-Brio & Serrano Cinca (1993), Serrano Cinca (1998a, 1998b), and Back *et al*. (1998). Serrano Cinca (1998a, 1998b) used SOM to address business problems such as corporate failure prediction, bond rating, financial performance assessment based on published accounting information, and the comparison of the financial and economic indicators of various countries. Back *et al*. (1998) used SOM to analyse and compare pulp-and-paper companies based on their annual financial statements. Karlsson (2002) used SOM to analyse world-wide telecom companies as to their financial performance. Eklund (2004) proposed SOM as a financial benchmarking tool for analysing world-wide pulp-and-paper companies.

Many authors (Witten & Franck, 2000; Costa, 2000; De Andres, 2001) have suggested the use of *hybrid* approaches when building classifiers. Quoting Witten & Frank (2000, p.39): "The success of clustering is measured subjectively in terms of how useful the result appears to be to a human user. It may be followed by a second step of classification learning where rules are learned that give an intelligible description of how new instances should be placed into the clusters." However, so far, few studies have implemented this approach efficiently. Williams & Huang (1997) have introduced a hybrid approach to identify customer groups that exert a significant impact on the insurance portfolio (insurance risk analysis) and customers who practise fraudulent behaviour (fraud detection). The *hybrid* approach of William & Huang (1997) consists of, firstly, an undisclosed clustering technique partitioning the data. Then, decision trees (Quinlan's C4.5 algorithm) are used to build a symbolic description of the clusters, and, finally, the selection of interesting "nuggets" from the rules inferred. In this dissertation, we compare different techniques for each part of the *hybrid* classifier: clustering and classification parts. We argue that *hybrid* systems that combine two or more CI methods are worth investigating. The choice of a clustering/classification technique for the *hybrid* classifier is problem- and context-dependent, as is suggested in previous research into comparing different clustering and classification techniques (De Andres, 2001). However, we find, for each experiment undertaken the most adequate hybrid classification model in terms of classification accuracy and class predictions. As an enhancement to other similar studies we introduce a standard method to compare the different approaches to the classification task (Section 5.2).

The relevance of this dissertation regarding the use of ANN to build regression models consists of introducing an alternative way of training an ANN based on its past training experience and weights reduction which improves the ANN forecasting capability. In addition, we present the steps necessary in designing and implementing the ANN as a forecasting tool.

Clustering/classification and regression models will reveal the weaknesses and strengths of the companies/countries involved in the analyses, which could benefit all business players if used.

At the company level, BI units are responsible for constructing these models. BI specialists will benefit from research that provides comparisons across a wide panoply of economic/financial performance classification and regression models, guiding them in correctly selecting an adequate model.

1.5 Overview of the Dissertation

In the subsequent paragraphs we present the structure of the dissertation.

Chapter 1 is an introductory chapter that, firstly, positions the research in the broader context of BI gathering, secondly presents the rationale and motivation of the research, thirdly describes the aim of the study and research questions, fourthly presents the related research and the relevance of the study, and, finally, gives an outline of the dissertation and presents the research contributions and a summary of publications' content.

Chapter 2 presents our research methodology. Firstly, it presents some well known research frameworks within the field of Information Systems and Social Science. Secondly, it positions the research by adopting a pluralistic research strategy emphasizing constructivism and following a number of specific guidelines for constructive research approaches.

Chapter 3 describes the knowledge discovery process in general and the data-mining process in particular and shows how both quantitative and qualitative data-mining techniques together with agent technology, can be integrated to construct Knowledge Building Systems. Next, it stresses the importance of quantitative data-mining methods and enumerates some of the most important areas of applicability for the former ones.

Chapter 4 presents some key business problems (stated in Chapter 3) that can be addressed through quantitative data-mining: economic/financial performance benchmarking and the prediction of process control variables. It also presents related research that addressed these business problems.

Chapter 5 is the most important in terms of research contribution. It presents a series of computational intelligence approaches to address the problems of economic/financial performance benchmarking and process control variables prediction. It compares the advantages and disadvantages of statistics, decision trees, neural networks, fuzzy logic and genetic algorithms when applied to assessing comparatively the economic/financial performance of countries/companies. A series of improvements in the algorithmic part of the discovery process is presented: a modified version of the FCM algorithm, new ways of validating the SOMs, new ways of training the ANNs. At the same time, we evaluate the prediction power of an ANN. Different technical problems related to the implementation of different CI methods are addressed. The need for *hybrid* approaches to solving data-mining classification task is also discussed.

Chapter 6 applies the methods described in the previous chapter using a number of experiments: the economic performance benchmarking of Central-Eastern European countries; the financial performance benchmarking of the most important companies from two large industry sectors – the pulp-and-paper and telecommunication sectors; the prediction of process control variables for the glass manufacturing process at Schott, a glass manufacturer from Germany.

The last Chapter summarises the managerial implications and the contributions of our research in performing data-mining tasks. It ends by presenting the limitations of and future directions for the study.

1.6 Contributions and Publications

The first main contribution of the research is that it explores the benefit of introducing hybrid approaches to solving some of the key problems within the areas of applicability of quantitative data mining such as countries'/companies' economic/financial performance benchmarking. The second main contribution of the research is to explore the use of ANNs for another area of applicability of quantitative data-mining: prediction of process control variables. We argue that by using our models, interested business parties can gain strategic advantages over their competitors.

The clustering/classification and forecasting models are a combination of statistics and different CI methods such as induction techniques, fuzzy logic, neural networks and genetic algorithms. The contributions of this research in using CI methods to each of the three quantitative data-mining tasks are as follows:

The use of SOM as a tool for performing DM clustering is enhanced in two ways. Firstly, the SOM is compared with other clustering methods as a tool for performance benchmarking such as C-Means clustering and fuzzy C-Means clustering. Secondly, the results of the SOM are used further in the analysis, as the input for the classification models. Moreover, we answer some technical questions related to the practical implementation of the SOM as a financial analysis tool: the validation of map dimensionality and of the quantisation error. We also introduce a new clustering algorithm – Weighting FCM – and show that it can perform better than normal FCM and SOM. We show that this new algorithm gives a better explanatory power for the clusters. We can *automatically* characterise each cluster and, also, find those observations that need to be treated carefully because of their specifics.

Our contribution to the research in using CI methods for performing the classification task is threefold. Firstly, *hybrid* classifiers have been explored and compared. This idea is not new, but little attention has been paid to it in the related literature. In Costa (2000) the author presents a number of methods that can be used for classification problems, regardless of the domain, and he states that one concern is related to the identification of possible ways of building ***hybrid*** solutions for classification. We find, for a particular problem, *the most adequate hybrid classification model* in terms of accuracy rate and class predictions and validate the performance of hybrid classifiers. Secondly, we are concerned with understanding the advantages and disadvantages of each approach when used in isolation. Thirdly, we investigate the implications of three different factors (pre-processing method, data distribution and training mechanism) on the classification performance of ANNs, we elaborate an empirical procedure for determining the ANN architecture, and find the best crossover operator in terms of GA-based ANN classification performances.

With regard to the research in using CI methods for performing the regression task, we enhance the applicability of ANN as a time series prediction tool, specifically to solving process variables prediction problems. We address some technical problems related to development of ANN architecture and ANN prediction error and we propose an alternative way of training an ANN based on its past training experience and weights reduction.

The research work and results have been published during the last two years in six scientific publications.

Publication 1, (Kloptchenko A, Eklund T, **Costea A**, Back B) *A Conceptual Model for a Multiagent Knowledge Building System* proposes a conceptual model of a knowledge-building system for decision support based on a society of software agents, and data- and text-mining methods. The novelty of the publication consists of the integration of several quantitative and qualitative data-mining techniques, namely self-organizing maps for clustering quantitative information, decision trees and/or multinomial logistic regression for classifying new cases into previously obtained clusters, prototype-matching for semantic clustering qualitative information, and various techniques for text summarisation. It is a joint publication that was initiated by Dr. Kloptchenko, but it was carried out as a joint effort by all the authors. My main contribution to the publication was to enhance the initial proposed architecture of the conceptual model. The paper has been published by Kluwer Academic Publishers in the blind peer-reviewed conference proceedings: *Proceedings of 2003 5th International Conference on Enterprise Information Systems*.

Publication 2, (**Costea A**, Eklund T) *A Two-Level Approach to Making Class Predictions*, proposes a new two-level methodology for assessing countries'/companies' economic/financial performance. Two experiments are undertaken: assessing Central-Eastern European countries' economic performance and world-wide pulp-and-paper companies' financial performance. The methodology is based on two major techniques of grouping data: SOM clustering and predictive classification models. First, we use cluster analysis in terms of self-organizing maps to find possible clusters in data in terms of economic/financial performance. We then interpret the maps and define outcome values (classes) for each row of data. Lastly, we build classifiers using two different predictive models (multinomial logistic regression – MLR and decision trees – DT) and compare the accuracy of these models. Our findings indicate that the results of the two classification techniques are similar in terms of accuracy rate and class predictions. Furthermore, we focus our efforts on understanding the decision process corresponding to the two predictive models. Moreover, we claim that our methodology, if correctly implemented, extends the applicability of the self-organizing map for clustering of financial data, and thereby, for financial analysis. I was the main author of the publication. This paper has been published by IEEE in blind peer-reviewed conference proceedings: *Proceedings of IEEE 36th Annual Hawaii International Conference on System Sciences*.

Publication 3, (**Costea A**, Eklund T) *Combining Clustering and Classification Techniques for Financial Performance Analysis*, is an enhanced version of *Publication* 2 in the sense that a new experiment is undertaken and, besides MLR and DT, ANNs are proposed as financial classification models. The goal of this publication is to analyse the financial performance of world-wide telecommunications companies by building different performance classification models. To characterise the companies' financial performance, we use different financial measures calculated from the companies' financial statements. The class variable, which for each entrance in our dataset tells us to which class any case belongs, is constructed by applying SOM clustering. We address the issue of SOM map validation using two validation techniques. Then, we build different classification models: multinomial logistic regression, decision-tree induction, and a multilayer perceptron neural network. During the experiment, we found that logistic regression and decision-tree induction performed similarly in terms of accuracy rates, while the multilayer perceptron did not perform as well. Finally, we propose that, with the correct choice of techniques, our two-level approach provides additional explanatory power over single-stage clustering in financial performance analysis. I was the main author. The paper was blind peer-reviewed and published in *Proceedings of 2004 IIIS 8th World Multi-Conference on Systemics, Cybernetics and Informatics*.

In ***Publication 4***, (Alcaraz AF, **Costea A**) *A Weighting FCM Algorithm for Clusterisation of Companies as to their Financial Performances*, we apply fuzzy logic to group telecommunications companies into different clusters based on their financial performances. The objective is to build an easy-to-use financial assessment tool that can assist decision makers in their investment planning and be applied regardless of the economic sector to be analysed. We characterise each cluster in terms of profitability, liquidity, solvency and efficiency. We implement a modified fuzzy C-Means (FCM) algorithm and compare the results with those of normal FCM and previously reported SOM clustering. The results show an improvement in pattern allocation with respect to normal FCM and SOM. The interpretation of the clusters is done automatically by representing each ratio as a linguistic variable. I contributed equally to the joint effort of constructing the Weighting FCM and writing the publication. The paper was blind peer-reviewed and published in *Proceedings of 2004 IEEE 4th International Conference on*

Intelligent Systems Design and Applications. A shorter version of this publication is Alcaraz & Costea (2004a).

Publication 5, (**Costea A**, Nastac I) *Assessing the Predictive Performance of ANN-Based Classifiers based on Different Data Pre-Processing Methods, Distributions, and Training Mechanisms*, analyses the implications of three different factors (pre-processing method, data distribution and training mechanism) on the classification performance of artificial neural networks (ANNs). We use three pre-processing approaches: "no pre-processing", "division by the maximum absolute values", and "normalisation". We study the implications of input data distributions using five datasets with different distributions: the real data, uniform, normal, logistic and Laplace distributions. We test two training mechanisms: one belonging to the gradient-descent techniques, improved by a retraining procedure (RT), and the other is a genetic algorithm (GA), which is based on the principles of natural evolution. The results show statistically significant influences of all individual and combined factors on both training and testing performances. A major difference from other related studies is the fact that for both training mechanisms we train the network using as a starting solution that obtained when constructing the network architecture. In other words, we use a hybrid approach by refining a previously obtained solution. We found that when the starting solution has relatively low accuracy rates (80-90%), GA clearly outperformed the retraining procedure, while the difference was smaller to zero when the starting solution had relatively high accuracy rates (95-98%). As has been reported in other studies, we found little to no evidence of crossover operator influence on the GA performance. The publication is a joint publication in which I am the main author. The publication has been submitted to a blind peer-reviewed international journal, the *International Journal of Intelligent Systems in Accounting, Finance and Management*. An earlier version of this paper was published as TUCS Technical Report (Costea & Nastac, 2005).

Publication 6, (Nastac I, **Costea A**) *A Retraining Neural Network Technique for Glass Manufacturing Data Forecasting*, puts forward a retraining neural network-based forecasting mechanism that can be applied to complex prediction problems, such as the estimation of relevant process variables for glass manufacturing. The main purpose is to obtain a good accuracy of the predicted data by using an optimal feed-forward neural architecture and well-suited delay vectors. The artificial neural network's (ANN) ability to extract significant information provides a valuable framework for the representation of relationships present in the structure of the data. The evaluation of the output error after the retraining of an ANN shows that the retraining technique can substantially improve the achieved results. The empirical part of the publication was conducted for the EUNITE competition in 2003. Both authors worked for several months building the ANN prediction model, training the neural networks and finally, writing the publication. The paper has been published in *Proceedings of 2004 IEEE International Joint Conference on Neural Networks*, and was blind peer-reviewed before acceptance.

Figure 1-2 relates the publications with the main data-mining tasks and the business problems addressed in the dissertation.

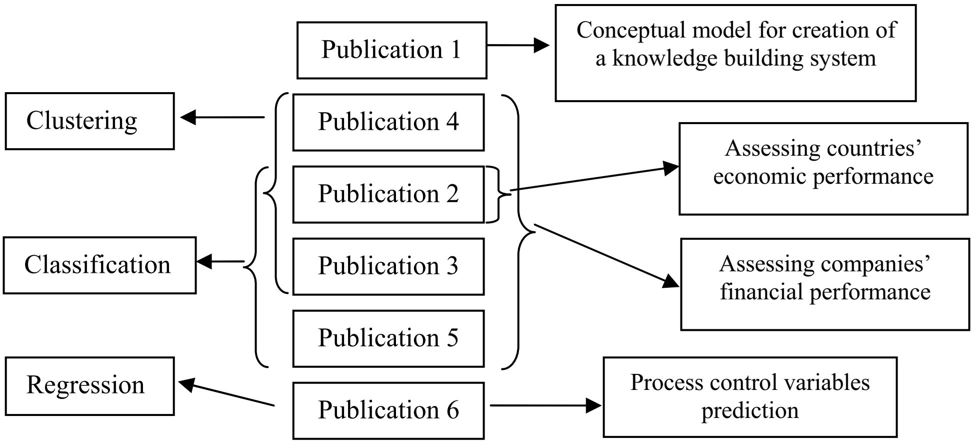

Publication 1
Conceptual model for creation of a knowledge building system
Clustering
Publication 4
Publication 2
Assessing countries' economic performance
Classification
Publication 3
Assessing companies' financial performance
Publication 5
Regression
Publication 6
Process control variables prediction

Figure 1-2 The interrelationship between data-mining tasks, publications and business related problems

Chapter 2 Research Methodologies

In this study, the term “methodology” implies two things, depending upon the *context* in which it is used. Firstly, it has a more concrete meaning related to a particular technique or combination of techniques *used* to model some kind of relationships (e.g. two-level methodology for assessing countries’/companies’ economic/financial performance). Secondly, “methodology” has a more abstract sense and refers not only to *what* was used to perform the research, but also refers to *how* this research was conducted (e.g. surveys, model building, case studies, etc.). In the later sense “methodology” is synonymous with a “super-methodology” that comprises all “sub-methodologies” used to conduct the research. This section uses the latter meaning of “methodology”.

The cyclical and interdisciplinary nature of the data-mining and knowledge discovery research process makes us believe that our research cannot and should not be based on one single research framework. In the next sections, we describe different research methodologies and research frameworks in the field of IS (Sections 2.1, 2.2) and show how our research relies on a combination of these research methodologies (Section 2.3).

2.1 Research Frameworks

In the following sections we present different research frameworks for the development of Information Systems (IS).

2.1.1 Burrel & Morgan’s / Iivari’s Research Framework

Burrell & Morgan (1979) presents a perspective for a research framework in the social sciences. Among the natural sciences, which include research in the physical, biological and behavioural domains, social science plays an important role (Figure 2-1).

The subjective-objective dimension				
The subjectivist approach to social science				The objectivist approach to social science
Nominalism	←	ontology	→	Realism
Anti-positivism	←	epistemology	→	Positivism
Voluntarism	←	human nature	→	Determinism
Ideographic	←	methodology	→	Nomothetic

Figure 2-1 A scheme for analysing assumptions about the nature of social science (Source: Burrell and Morgan, 1979)

The research framework presented in Iivari *et al.* (1998) extends that from Burrell & Morgan (1979) and applies it in an IS development context. In Figure 2-2 we present in a concise form the framework proposed by Iivari *et al.* (1998).

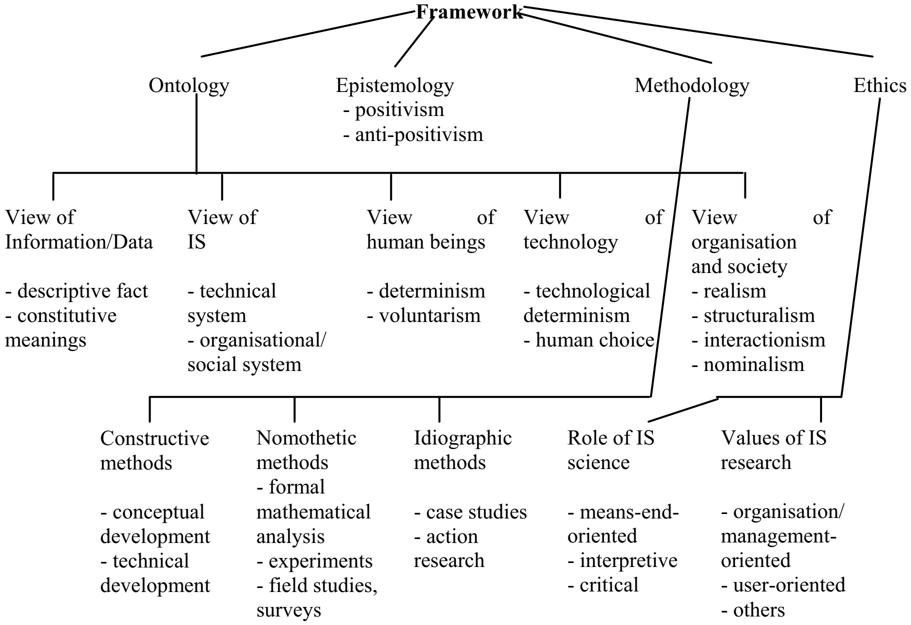


Figure 2-2 The framework for paradigmatic analysis
(Source: Iivari, 1991)

However, the two frameworks differ in two ways. Firstly, Iivari *et al.* (1998) assume that dimensions of the paradigmatic analysis are *ontology*, *epistemology*, *research methodology*, and *ethics* while, for Burrell & Morgan, the paradigmatic assumptions about the nature of the social world and the way in which it may be investigated are: *ontology*, *epistemology*, *human nature*, and *methodology*. Secondly, Iivari *et al.* (1998) assume that all these dimensions are not mutually exclusive dichotomies because “an ISDA – Information System Development Approach – may simultaneously incorporate assumptions from more than one paradigm” as opposed to Burrell & Morgan (1979), where everything is black or white. According to dictionaries (e.g. Merriam-Webster online dictionary) the term “paradigm” signifies the *generally accepted perspective* of a particular discipline at a given time. In contrast with Burrell & Morgan (1979), Iivari *et al.* (1998) include the “human nature” dimension of paradigmatic assumptions in the ontology dimension and add “ethics” as a new paradigmatic dimension. Iivari’s framework fits that of Fitzgerald & Howcroft (1998). The difference is that the former proposes an “ethics” dimension (what are the values that ought to guide IS research?), while the latter contrasts rigour and relevance in how the research should be validated (axiological level). In essence these two dimensions are equivalent, but with “ethics” we look more at the role of IS science (what this role should be) and at IS research value, while axiology is more concerned with the ways of producing research that is valuable.

According to Iivari *et al.* (1998), the paradigmatic assumptions that govern IS development research are defined as follows. The first paradigmatic assumption, *ontology*, is concerned with what is the nature of *information and data*, *information systems*, *human beings* as they are IS developers and users, *technology* and *human organisations* and *society* at large. There are two opposite ontological views as to the nature of reality: realism and idealism (nominalism). *Realists* see data as describing certain facts, IS as a technically implemented system, humans as being governed by deterministic laws and organisations as being relatively stable structures, while *idealists* use data to "construct" rather than "reflect" reality. Idealists emphasise the social nature of information systems. Iivari *et al.* (1998) borrow the distinction between *determinism* and *voluntarism* in the view of human beings from Burrell & Morgan (1979). Determinists regard a human being or his activities as being completely determined by the situation or environment where they perform, while voluntarists see the human being as completely autonomous and free-willed. For idealists the organisations or what is the external world for an individual is nothing more than names, concepts and labels that are used to structure reality (Burrell & Morgan, 1979).

The second paradigmatic assumption, *epistemology*, is concerned with what is human knowledge and how it can be acquired. In other words, epistemology is concerned with what should be the outcome of IS research. For some scientists (*positivists*) the outcome of IS research should consist of "highly generalisable methods and approaches assuming that IS research is governed by law-like regularities", while the others (*anti-positivists*) produce "constructs or metaphorical templates which could support IS developers with potentially useful insights that must be carefully evaluated anew in each project and situation". Positivist epistemology is based on traditional approaches that test hypothesised regularities. There are two opposite approaches as far as justification is concerned: *verificationism* and *falsificationism*. "Verificationists" think that hypothesised regularities can be verified by an adequate experimental research programme (Burrell & Morgan, 1979), while "falsificationists" (Popper, 1963) argue that scientists should concentrate on disproving claims, since one single counter/negative example is enough to disprove theories, while far greater positive claims cannot prove it to be true. For anti-positivists scientific knowledge evolves from the human interpretation and understanding of facts and can be achieved only by the individuals that are directly involved in the activities to be studied.

The third paradigmatic assumption, *research methodology*, is concerned with the preferred research methods for development of IS applications and with the modes of evidence-giving by which these research methods are justified. We have to stress, again, that, here, "research methodology" has an abstract sense and refers to the approach that is undertaken in the research rather than, for example, particular techniques used to model some kind of relationships. As Iivari *et al.* (1998) point out: "The term *research methodology* in this context refers to the procedures (research methods) used to acquire knowledge about ISDAs (Information System Development Approaches) and ISDMs (Information System Development

Methodologies), methods, and tools. The knowledge referred to in the context of ISDAs consists of the canons and principles needed to elaborate and refine the ISDA. This is analytically separate from the canons and principles which designers and users follow when building an IS application". Burrell & Morgan (1979) divide the research methods into two classes: nomothetic and idiographic. Iivari *et al.* (1998) add constructive methods as the third research methodology class, arguing that Information Systems and Computer Science are applied disciplines and require methods that should be concerned with the engineering of artefacts. These artefacts may be either conceptual artefacts (models, frameworks, procedures) or more practical artefacts (e.g. pieces of software). Nomothetic methods (formal-mathematical analysis, surveys, experimental methods, and laboratory and field experiments) are concerned with the scientific rigour of the research and use hypothesis testing and other quantitative techniques to analyse the data. Idiographic methods are based on first-hand knowledge of the subject under investigation and generate insights "revealed in impressionistic accounts found in diaries, biographies and journalistic records" (Burrell & Morgan1979, p. 6).

The fourth paradigmatic assumption, *ethics of research*, is concerned with the responsibilities that the researcher should acknowledge for the consequences of his/her research. There are two main concerns related to this paradigmatic assumption: the roles of IS research as an "applied" science, and the values produced by IS research. There are three potential roles for IS science (cf. Chua, 1986 and Oliga, 1988): means-end-oriented, interpretive and critical. Research from the first category aims at describing means for achieving different goals. The legitimacy of the goals is not questioned. Interpretivists assume that the world is inter-subjective and that science can represent the world with concepts and social constructs (Kvassov, 2002). Inter-subjectivity is the process of knowing others' minds. For interpretivists knowledge and meaning are acts of interpretation. The goal of interpretive scientist is to "enrich people's understanding of their action", "how social order is produced and reproduced" (cf. Chua 1986). Critical scientist assumes that research goals can be subjected to critical analysis just as well as means (Iivari *et al.*, 1998). The values of IS research should be looked at from two perspectives: who benefit from the research, and whether IS research should be considered value-free research or not. IS users, IS professionals, top management can benefit from IS research. Positivists claim that research can and should be value-free[3], as opposed to antipositivists, who deny this possibility.

As we stated earlier Burrell & Morgan (1979) simplified the research framework by dividing the research approaches in two distinct classes. Iivari *et al.* (1998) show that this separation is too simplistic, especially, when applied to IS research:

[3] Pearson (Pearson, 1900) claimed that the essence of science is the accumulation and classification of "facts". In 1930's the apologists of the logical positivist movement made an even more explicit demarcation of statements into two categories: positive and normative. Positive statements are statements of fact, while normative statements are statements of opinion. Popper (1963) was opposed to logical positivists and argued that there are no pure statements of value-free or "positive" facts and that all facts carry out value.

even though “idiographic methods appear more closely associated with the idealist ontological position”, “some positivist case studies also appear to fall in this category”. The authors cited Lee (1991) who integrated positivist and interpretative approaches. However, in order to make our life easier in making the decision regarding the research approach to be undertaken, it is advisable to follow the scheme presented in Figure 2-1. Realist ontology should be accompanied by a positivist epistemology and nomothetic research methods. This constitutes the objectivistic approach of Burrell & Morgan’s scheme, as opposite to the subjectivist one that comprises: nominalist ontology, antipositivist epistemology and ideographic research methods.

The subjective-objective approach (Figure 2-1) to research in the field of IS is not the only dichotomist view that exist in the literature. Orlikowski & Baroudi (1991) propose two antagonistic approaches to IS research: *interpretive* versus *positivist*. Kaplan & Duchon (1988) contrast qualitative and quantitative approaches to IS research.

2.1.2 March & Smith’s Research Framework

March & Smith(1995) present a two-dimensional framework for research in information technology (IT). This framework (Figure 2-3) can be applied as a research framework in the field of IS since the authors make no specific distinction between IT and IS as concepts. For the authors IT is typically instantiated as IT systems, which are “complex organisations of hardware, software, procedures, data, and people, developed to address tasks faced by individuals and groups, typically within some organisational setting”. While not falling within the scope of this section to differentiate the IT and IS concepts, it is worth mentioning that a distinction, however, exists in the literature: Checkland & Holwell (1998) state that IS research has to be looked at in the *context* of IT.

		Research Activities			
		Design science		Natural Science	
		Build	Evaluate	Theorise	Justify
Research Outputs	Constructs				
	Models				
	Methods				
	Instantiations				

Figure 2-3 A research framework in natural and design sciences
(Source: March & Smith, 1995)

March & Smith take as reference Simon’s (1981) work and state that IT research is about artificial as opposed to natural phenomena and that artificial phenomena can be both created and studied. There are two kinds of scientific interest in IT, *descriptive* and *prescriptive*. Descriptive research aims at understanding the nature of IT. Prescriptive research aims at improving IT performance (March & Smith, 1995). Therefore two distinct pieces of science, design (prescriptive) and natural (descriptive) sciences can contribute to IT research. Natural science is concerned with understanding reality – explaining how and why things are – and has two main research activities: discovery (generating theories, laws, etc.) and justification (activities that test theory's claims). Justification is the concern of the apostles of the two opposed ways of validation, "verificationists" and "falsificationists" described earlier. Design science is based on two research activities: *build* and *evaluate*. Design science is concerned with building artefacts and evaluating their practical performance.

There are four research outputs of the design science research: *constructs*, *models*, *methods* and *instantiations*. The constructs are the semantic elements that conceptualise problems within a domain and their solutions. Models constitute a set of statements that describe the relationships between constructs. Models need methods to be implemented. Instantiations are the final artefacts, limited in their scope and developed on the basis of constructs, models and methods. In IT research, instantiations can precede the complete definition of constructs, models and methods, by having designers rely on their intuition and experience.

The March & Smith (1995) framework supports the interactions between the two species of scientific activity: design and natural sciences that can be reciprocal. Firstly, natural scientists create theories that can be exploited by design scientists when constructing the models. However, natural science is not always able to explaining how and why an artefact works. Design science outputs (artefacts) can give rise to phenomena that can be the targets of natural science research (March & Smith, 1995).

2.1.3 Järvinen's Research Framework

Another IS research framework (Figure 2-4) is proposed by Järvinen (2001). The author divides research strategies into six classes: *mathematical approaches*, *conceptual-analytical approaches*, *theory-testing* and *theory-creating approaches*, and *innovation building* and *evaluation approaches*.

Firstly, Järvinen differentiates mathematical methods from other methods because they use symbols (e.g. algebraic units) that do not have a direct correspondence to objects in reality.

Next, the author differentiates between methods concerning reality using different types of research questions. There are two main classes with regard to the type of research question: one class contains research questions concerning what is a (part of) reality – *basic research* – and the other includes research questions that stress the building and evaluation process of innovations – *applied research*. The first class is again divided into two subclasses: *conceptual-analytical approaches* (which include methods for *theoretical development*) and *empirical research approaches*.

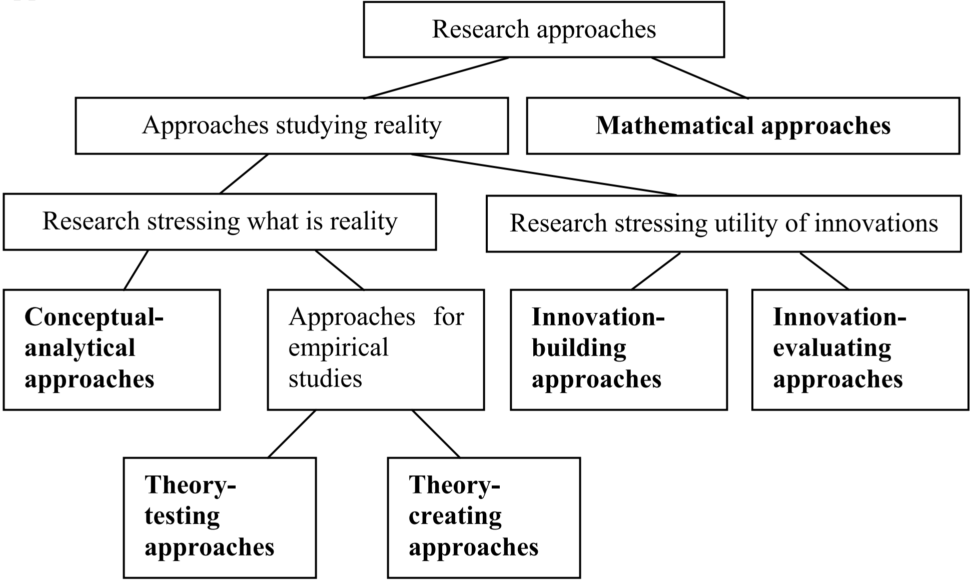

Research approaches
Approaches studying reality
Mathematical approaches
Research stressing what is reality
Research stressing utility of innovations
Conceptual-analytical approaches
Approaches for empirical studies
Innovation-building approaches
Innovation-evaluating approaches
Theory-testing approaches
Theory-creating approaches

Figure 2-4 Järvinen's taxonomy of research methods
(Source: Järvinen, 2001)

Theories can be *descriptive* and *normative*. Descriptive theories show what kind of general regularities describe the phenomenon under study, while normative theories set a standard and show what kind the phenomenon ought to be. Sometimes normative theory is known as *prescriptive* theory, which is concerned with "prescribing" what the phenomenon should be so that optimistic goals are met and pessimistic consequences can be prevented. Theory building can be done *deductively* and *inductively*. Deductively, a theory is derived from axioms or assumptions, while, inductively, a theory is derived from empirical generalisations or by interpreting old results in a new way. Descriptive and normative theories can be derived both deductively and inductively. These two types of theories are related to March & Smith's descriptive and prescriptive research. The former is concerned with understanding the nature of IT, while the latter aims at improving IT performance.

Empirical research approaches comprise *theory-testing approaches* which include methods such as laboratory experiments, surveys, field studies, field tests and a particular form of a case study (proposed in Lee, 1989) and *theory-creating approaches* which include, essentially, qualitative and explorative research methods such as normal case studies, grounded theory. As Järvinen (2001, p. 64) says: "In theory-creating approaches there are some general features. The raw data of studies is often text. Even images, voice recordings and videos are transcribed into text. The new theory is *compressed* from the raw data…"

The last two classes (*innovation-building* and *innovation-evaluation*) resemble March & Smith's build and evaluate activities of design science and Iivari's constructive methodology. *Innovation-building* is concerned with building artefacts that perform different tasks. Sometimes even the enhancement, improvement, extension, adjustment, transformation of an existing artefact is considered innovation-building. In the *evaluation* phase of the innovation some criteria are used and some measurements performed.

The differences between Järvinen (Figure 2-4) and the other frameworks presented (Iivari *et al.*, 1998 – Figure 2-2 and March & Smith, 1995 – Figure 2-3) can be summarised as follows:

1. Järvinen separates the mathematical approaches from the other approaches that do not have a direct link with reality.
2. Järvinen argues that March & Smith's decision to accept only natural science to describe the world is too restrictive. He adds social science to describe the human side of life.
3. Järvinen and Iivari present their research framework in a tree-like form, while March & Smith's framework has a tabular form, which is more confusing.
4. Besides March & Smith's fidelity with real world phenomena, completeness, level of detail, internal consistency, efficiency, generality, ease of use and impact on the environment and users as criteria for evaluating models, methods and instantiations (innovations) Järvinen adds form and content, richness of knowledge and in addition to economic, technical and physical impacts, also impact on social, political and historical contexts (cf. Kling, 1987 cited in Järvinen, 2001).

2.1.4 Constructivism

As we mentioned earlier, Iivari added a new methodological level to his paradigmatic framework: constructivism. Constructive methods are concerned with engineering artefacts. The author distinguishes between two types of artefact construction: *conceptual development* (development of models, frameworks, and procedures) and *technical development*, which means the implementation of conceptual artefacts, for example, through programming languages. Kasanen *et al.* (1993) discuss the applicability of a constructive approach in management accounting research. The authors argue that a constructive approach means problem solving through the construction of organisational procedures and models. Constructions are entities that produce solutions to explicit problems. The usability of the constructions can be demonstrated through implementation of the solution. In management accounting research the constructions are called managerial constructions and address problems that come up in an organisation's life. An artificial language (Morse alphabet, computer programming languages) or, in the case of managerial accounting, a new budgeting system, a new performance benchmarking model, can be considered as examples of constructions. Constructions should not be mixed up with "constructs" from March & Smith (1995) design research. Design science instantiations can be viewed as a type of construction. Some authors (Järvinen, 2001; Lainema, 2003) consider, and we incline to agree, that design science and constructivism are one and the same approach. At the same time, Kasanen's constructions are similar to Simon's or Iivari's artefacts and with Järvinen's innovations.

2.2 Pluralistic Research Strategy Emphasizing Constructivism

Some authors argue that information systems is an essentially pluralistic field (Fitzgerald & Howcroft, 1998; Iivari, 1991; Nissen *et al*., 1991). Banville & Landry (1989) suggest that the field of information systems can best "be understood and analysed only with the help of pluralistic models". Järvinen (2001) claims that "a single research perspective for studying information systems phenomena is unnecessarily restrictive". Hassard (1991) proposed a successful multi-paradigm approach in organisational analysis.

Pluralistic research strategy is one of the four research strategies that a researcher can pursue when he/she is confronted with two dichotomist research approaches. Fitzgerald & Howcroft (1998) describe these four strategies in the case of a "hard" – "soft" framework dichotomy as follows:

When an *isolationist* strategy is adopted, the researcher operates strictly according to a particular paradigm and ignores other alternatives. This is in accordance with Burrell & Morgan's framework (Figure 2-1), where the two opposite paradigmatic assumptions are mutually exclusive and exhaustive. The drawback of this approach is that it misses the insights that can be obtained by applying both paradigms complementarily.

The second research strategy is to make each research approach strive to achieve *supremacy*. There is little interest in pursuing this strategy since both approaches have strengths and weaknesses. Also, because the IS field is short-lived, the positivist ("hard") approach, which favours quantitative methods, has taken supremacy. However, the interpretive approach, which favours qualitative methods, has been increasingly adopted.

Another research strategy consists of *integration* of the competing approaches. Lee (1991) proposed an integration of interpretive and positivist approaches. Some authors argue that this integration can be problematic (Newman & Robey, 1992; Walsham, 1995 – both cited from Fitzgerald & Howcroft, 1998). The integration into a single approach can make "each approach scarifying its particular strengths".

The fourth research strategy that a researcher can adopt is *pluralistic* strategy. A pluralistic strategy means that the research approaches are not mutually exclusive. In other words, it means combining opposite approaches of different abstraction levels (e.g. combining realist ontology with subjectivist epistemology – data are seen as describing certain facts and the scientific knowledge is obtained through human interpretation and understanding of these facts) and even opposite approaches of the same abstraction level (e.g. at the methodological level combining qualitative and quantitative methods). Iivari (1991) suggests that epistemological monism can coexist with methodological pluralism. At different abstraction levels (paradigmatic assumptions) there are the following competing dichotomies (Fitzgerald and Howcroft, 1998):

- interpretivist vs. positivist – paradigm level,
- relativist (idealist) vs. realist – ontological level,
- subjectivist vs. objectivist and emic/insider/subjective vs. etic/outsider/objective – epistemological level,
- qualitative vs. quantitative, exploratory vs. confirmatory, field vs. laboratory, and idiographic vs. nomothetic – methodological level,
- relevance vs. rigour – axiological level.

2.3 The Approach Taken in This Work

In our research we follow the pluralistic strategy (suggested by Iivari *et al.*, 1998), emphasizing objectivity and measurement (e.g. positivist approach at Fitzgerald & Howcroft's paradigmatic level). The pluralistic nature of our research comes from the combination of different approaches at the methodological level.

We position our research with respect to Iivari *et al.* (1998) and Fitzgerald & Howcroft (1998) as follows:

- we adopt a realist approach at the ontological level,
- we adopt a positivist/objectivist approach at the epistemological level,
- we use a combination of approaches at the methodological level,
- we use rigour at the axiological level (we use quantitative methods for testing our research questions with emphasis on internal validity) as opposed to relevance (here the focus is on external validity and relevance to practice).

At the methodological level the dichotomies are treated as follows:

1. Concerning the qualitative-quantitative research dichotomy we present the conceptual framework of constructing a knowledge building system that will combine different data-mining techniques (qualitative and quantitative) to perform different data-mining tasks. However, in this dissertation we focus on quantitative data-mining methods.
2. We combine exploratory approaches (discover patterns in data and describe the particular characteristics of the subject under consideration) with confirmatory approaches (test hypotheses and theory verification).
3. We construct our models in a laboratory setting rather than by using field experiments. However, the data collected for analysis are accurate and reflect the real performance of the countries/companies/processes under study. The

nature of the data means that our experiments resemble field rather than laboratory experiments.

4. With regard to the last methodological dichotomy (idiographic vs. nomothetic) we follow the constructive research methodology proposed by Iivari *et al.* (1998) as an alternative to the nomothetic and idiographic methods (Figure 2-2).

Constructivism is closely related to the research in this dissertation since here we help decision makers to make decisions by *constructing* different performance benchmarking and forecasting models. Kasanen *et al.* (1993) give an example of three dissertations which used constructive research. One of these dissertations, Wallenius (1975), developed and compared new methods to support multi-criteria decision making. The methods were compared in a laboratory setting using hypothetical decision makers and a Belgian steel company was used as a case study for implementations. We take a similar approach in our research, following Järvinen's innovation building and evaluation approaches.

Liu (2000) argues that the constructive approach is a meta-methodology that comprises multiple sub-methodologies, each one involving a group of techniques. The constructive approach is "more concrete than a general philosophy for analysing the world, but broader and more flexible than a specific methodology that is usually limited to a specific type of research" (Liu 2000, p. 79).

Hevner *et al*. (2004), an extension of March & Smith (1995), proposed seven guidelines to help researchers, reviewers, editors and readers to understand how to perform effective design science (constructive) research. They argue that these guidelines should be addressed in some manner if the design-science research is to be complete.

The *first* guideline suggests that the design-science must produce a viable artefact (in the form of a construct, model, method, or instantiation). We follow this guideline by constructing *hybrid* models for assessing economic and financial performance and ANN-based models for predicting process variables.

The *second* guideline states that the objective of design-science research should be to develop solutions for *relevant* business problems. We address business problems such as countries'/companies' economic/financial performance benchmarking, and process variable prediction with the means provided by the CI methods. In other words we use *new* methods for solving *old*, but still important, problems.

The *third* guideline is concerned with design evaluation. We evaluate our models by using several criteria: quantitative, such as quantisation error, accuracy rate or mean square error, or qualitative, such as fidelity with real world phenomena, form and content, and richness of knowledge (cf. Järvinen, 2001) in the form of class predictions. Hevner *et al*. (2004) support this view: "IT artefacts can be evaluated in terms of functionality, completeness, consistency, accuracy, performance, reliability, usability, fit with the organisation, and other relevant quality attributes".

The *fourth* guideline suggests that the design-science research must provide clear and verifiable contributions in the area of design artefacts, design foundations and/or design methodologies. Our contributions to using CI methods for addressing business problems are described in Section 1.6. We contributed to the area of artefact design by presenting for each model the design steps that should be followed. Our small contribution to design methodologies consists of positioning our research with regard to the many IS research frameworks available in the literature.

By using *rigorous* methods for constructing and evaluating our models we addressed the *fifth* guideline concerned with research rigour.

The *sixth* guideline (do design as a search process) is of particular importance for our dissertation. Design science research, particularly constructing clustering/classification and forecasting models, is inherently iterative. Simon (1996) describes the design process as a generate/test loop. We developed our models iteratively, compare them to each other and apply the best for a particular case. It is unfeasible to test all possible solutions for a given problem. We have to restrict the models used, relying on *satisfactory* solutions for the specific problem. In our research we followed the same idea comparing a certain number of satisfactory models. We chose the best model for each experiment undertaken.

The *seventh* and last guideline is concerned with communicating the research. The research results have to be presented effectively both to technology-oriented as well as management-oriented audiences. We follow this guideline by posing two types of research questions: BI research questions, related to the business problems addressed intended for managerial audiences, and technical research questions, related to CI methods used for technical designers.

Essentially, all the *Publications* that support this dissertation follow the constructive research approach. The summary of the publications' content is presented in Section 1.6. Both descriptive and prescriptive approaches are present in the publications. Firstly, a description of the problem addressed and a conceptual model for knowledge building are given (*Publication* 1). Then prescriptive models – both classification and forecasting models are prescriptive – are built to assess the future economic/financial positions (*Publications* 2, 3, 4, 5) and process variables values (*Publication* 6), respectively. The new algorithm (Weighting-FCM) proposed in *Publication* 4 can be considered as a construction since it is a transformation of an existing algorithm. The empirical procedure for determining the proper ANN architecture and the new way of training ANN based on its passed experience and weights' reduction (*Publications* 5, 6) can also be considered part of the design process.

Chapter 3 Data Mining and the Knowledge Discovery Process

In this Chapter we describe the knowledge discovery process, outline the data-mining tasks and the algorithms used to perform these tasks and present a conceptual model for knowledge creation.

3.1 Data, Information, Knowledge

Before we describe the knowledge discovery (in databases) – KDD – process, we discuss the different concepts that are closely related to this process such as: *data*, *information*, and *knowledge*. Even though they are not interchangeable these three terms are related. For organisations it is crucial to clarify what data, information and knowledge mean, which of them is needed, which of them the organisations already possess, how they differ and how to get from one to the other.

In a general context, ***data*** is a set of discrete, objective facts about events (Davenport & Prusak, 1998). In an organisational context data are seen as a collection of transaction records that has no significance beyond its existence. Data can be considered as a driver for information and knowledge, a means through which information and knowledge can be stored and transferred. Nowadays, there is a shift in data management responsibility: from a centralised information systems department to individuals' desktop PCs. In other words, the availability of data within the organisation has increased along with the technology that supports distributed systems. Even though organisations need and are sometimes heavily dependent on data, it does not mean that more data are necessarily better data. As Davenport & Prusak (1998) suggest, the argument that one should gather more data so that the solutions for the organisation problems will rise automatically is false from two perspectives: first, too much data can *hide* the data that matter and, second, data provide no judgment or interpretation about what has happened.

Information is data that have relevance and purpose (Davenport & Prusak, 1998) or a flow of meaningful messages (Nonaka & Takeuchi, 1995). The information is commonly seen as a message that "gives shape to" data. It has a sender and a receiver, but judgment of the information value – if it really informs the receiver or not – rests with the receiver. According to Davenport & Prusak (1998) there are several ways of transforming data into information: *contextualisation* – the purpose for what the data were gathered is known; *categorisation* – the units of the analysis or key components of the data are known; *calculation* – transformation of the data using mathematics or statistics; *correction* – the data are cleared of errors; *condensation* – the data are summarised in a concise form. The information can be transmitted using *soft* or *hard* networks. Among hard networks we mention electronic mail-boxes, wires, online instant messengers, satellite, post offices, etc. Soft networks are informal meetings, coffee-breaks, etc. Both information and explicit knowledge can be transmitted via soft networks. In the literature there is still confusion about the difference between information and knowledge. In their paper Kogut & Zander (1992) present information as a form of knowledge, stating that information is "knowledge which can be transmitted without loss of integrity". Stenmark (2002) cites seven papers which used different definitions of data, information, and knowledge (Stenmark 2002, Table 1, p. 2).

Davenport & Prusak (1998) propose a working definition of ***knowledge***: "knowledge is a fluid mix of framed experience, values, contextual information, and expert insight that provides a framework for evaluating and incorporating new experiences and information". Knowledge can be experience, concepts, values, beliefs that increase an individual's capability to take effective action (Lai & Chu, 2000). Knowledge is obtained from individuals, communities of thoughts or well-established organisational routines and rules. Knowledge can be *tacit* or *explicit* (Nonaka & Takeuchi, 1995). Tacit knowledge resides in people's mind, while explicit knowledge is knowledge that can be articulated using grammatical statements, mathematical expressions, specifications, and manuals. Some authors (e.g. Stenmark, 2002) argue that explicit knowledge is, in fact, information. In this study we adopt the emerging (practical) sense of "knowledge" concept: at the organisational level knowledge is the information that is organised in meaningful patterns that can help managers make crucial decisions. Knowledge derives from information as information derives from data. The transformation of information through knowledge is done according to Davenport & Prusak (1998) through human-like activities such as *comparison* – how does information about this situation compare to other situations that are known; *consequence* – what are the implications of the information for decisions and actions; *connections* – how does this piece of knowledge relate to others; and *conversation* – what do other knowledgeable people think about this information. Quigley & Debons (1999) relate information with who?, when?, what?, and where?, and knowledge with why?, and how?. In our thinking all human-like activities through which information can be translated into knowledge can be performed partially using computational intelligence techniques. Comparisons and connections are highlighted using different clustering techniques; consequences of some actions can be traced using classification models. A society of intelligent software agents can resemble human beings' conversations by sharing and exchanging information about common goals. We agree with the fact that the total substitution of humans by intelligent systems is neither possible nor efficient. At the same time, we think that intelligent systems can provide interested parties with useful and timely information that can be easily transformed into knowledge by the receiver, something that even very experienced people cannot provide. We look at our models as a complementary source to support the decision-making process. Knowledge can also move down the value chain and become information and data. Too much knowledge is hard to disseminate. As the ancient Greek playwright Aeschylus said: "Who knows useful things, not many things, is wise".

Knowledge is a very important asset for organisations, especially because other resources (technology, capital, land, and labour) are no longer sources of sustainable competitive advantage (Davenport & Prusak 1998, p.16). The most important value that knowledge brings to a company is that it can make that it can give the company a competitive advantage over rivals since it is incorporated into people's minds and, unlike material assets, it increases with use.

3.2 Data Overload and Data Usefulness

Nowadays, companies are bombarded with masses of data about their market environment. Such publicly available data are crucial for their competitiveness. The managers face two problems with regard to these data: *data overload* and *data usefulness*. Table 3-1 shows that the Internet is the second largest among data (information) generation mechanisms.

Table 3-1 New data produced in 2002 in terabytes

Medium (electronic flows)	**2002 Terabytes[4]**	***Internet***	**2002 Terabytes**
Radio	3,488	Surface Web[5]	167
Television	68,955	Deep Web[6]	91,850
Telephone	17,300,000	Email (originals)	440,606
Internet	***532,897***	Instant messaging	274
TOTAL	**17,905,340**	**TOTAL**	***532,897***

(Source: Lyman and Varian, 2003)

In 2000 the estimated volume of data on the public Web reached 20 to 50 terabytes. By 2002 it had tripled (167 terabytes). Lyman & Varian (2003) randomly selected 9,800 websites in order to estimate the size of an average webpage and the content of an average website. The sum of all 9806 website size was 33.1 GB. According to the NetCraft Survey[7] as of August 2003, these 9800 website represent 0.02 per cent of the 42.8 million web servers, which gives a total size of the surface web of 167 terabytes. If we follow the same methodology and take the number of sites estimated by Netcraft Survey for May 2005 (64 millions), we can calculate the size of surface web in May 2005 (221 terabytes). Among the storage media (paper, film, magnetic, optical) 92.5% of the data has been stored in 2002 in magnetic form.

Data usefulness is closely related to the process of transforming data into knowledge. The better the data, the better the knowledge obtained from the data.

[4] 1 terabyte = 10^{12} bytes

[5] Fixed Web pages

[6] As quantified in a landmark study by BrightPlanet in 2000, the "deep Web" (the database driven websites that create web pages on demand) is perhaps 400 to 550 times larger than the information on the "surface" (Lyman & Varian, 2003).

[7] http://news.netcraft.com/archives/web_server_survey.html

KDD addresses both these problems (information overload and data usefulness) by looking at the "new generation of computational theories and tools that can assist humans in extracting useful information (knowledge) from the rapidly growing volumes of digital data" (Fayyad *et al*., 1996b).

KDD lies at the confluence of many different disciplines and research fields such us *statistics*, *information theory*, *databases*, *artificial intelligence*, *machine learning*, *pattern recognition*, *fuzzy sets*, *visualisation*, and *high-performance computing*. Besides these fields, there are certain other long-term contributors to the KDD growing research field that have received less mention: *sciences*, *logic*, and *philosophy of science* (Klösgen & Zytkow 2002, p. 22). The link between sciences (quantitative theories) and KDD adds to the usefulness of the empirical demonstrations and generalisations that can be extracted from the data. In science (e.g. chemistry, physics) basic laws and theories can emerge from a concrete experiment that can be applied to a broad range of situations (Klösgen & Zytkow 2002, p. 23). In KDD the search for patterns in data can be followed by transforming the discovered regularities into theories that cover many datasets. The framework of logic is the base for many research disciplines such as mathematics, the theory of databases, artificial intelligence, and, therefore, is linked indirectly with the KDD process. For example, in KDD we may generate some classification rules and treat a minimal number of them as axioms and the others as derived from the axioms, thus resembling a deductive system. However, KDD is undermining the application of deductive systems by accepting a limited accuracy for the axioms. Inductive logic programming is also present in the emerging KDD field (e.g. data can be expressed as Prolog literals, while knowledge takes the form of Prolog rules). The influence of the philosophy of science on KDD is mainly indirect through the introduction of key field concepts and research frameworks that any well-established research field should have.

3.3 The KDD Process

KDD is the *nontrivial process* of identifying *valid*, *novel*, *potentially useful*, and ultimately *understandable patterns* in data (Fayyad *et al*., 1996c). In other words, KDD is the process of transforming data into knowledge.

In a KDD definition, *data* are represented by a set of facts (entries in a database), while *pattern* refers to a subset of the data that share similar characteristics or to some rule that covers a number of observations. The term *process* of the definition implies that KDD consists of many steps, which involve data preparation, pattern discovery and knowledge evaluation and refinement. All these steps are performed iteratively. The term *non-trivial* is related to the data-mining step of the KDD process in the sense that the methods used to analyse the data are not trivial (e.g. computing averages), but advanced (CI methods). Fayad *et al*. (1996b) consider the patterns to be knowledge if they "exceed some interestingness threshold" and are determined "by whatever functions and thresholds the user chooses". In other words, knowledge is user-oriented and domain-specific.

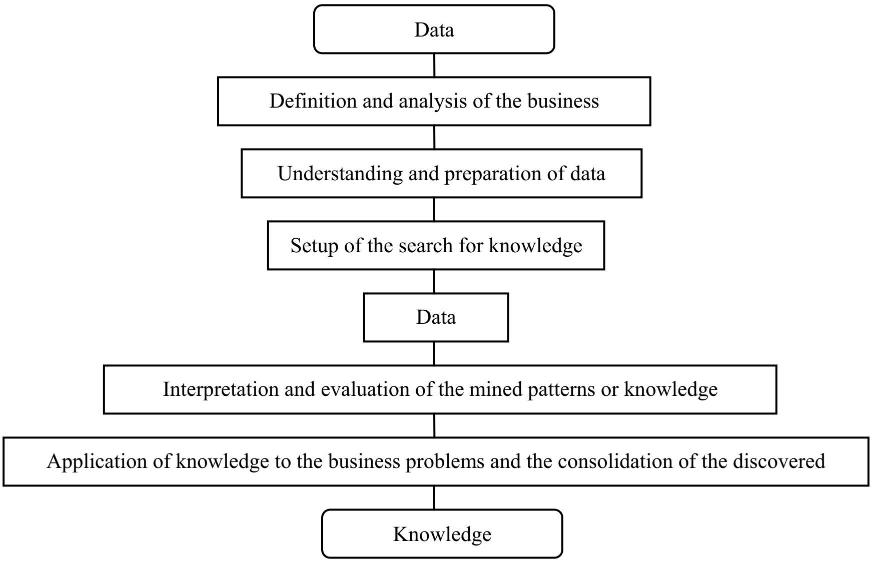

Data
Definition and analysis of the business
Understanding and preparation of data
Setup of the search for knowledge
Data
Interpretation and evaluation of the mined patterns or knowledge
Application of knowledge to the business problems and the consolidation of the discovered
Knowledge

Figure 3-1 An overview of the steps of the KDD process
(Source: Klösgen & Zytkow, 2002)

The discovered patterns should be *valid,* which means that they should be valid for new data with some degree of certainty (accuracy). Patterns should be *novel* (unknown), *potentially useful* (lead to some benefits) and *understandable* for the users after (if necessary) some post-processing.

Fayyad et al. (1996b) propose a description of the KDD process that consists of nine steps. Shearer (2000) describes CRISP-DM (CRoss Industry Standard Process for Data Mining), a non-proprietary, documented and freely available data-mining model. CRISP-DM consists of the following six phases: business understanding, data understanding, data preparation, modelling, evaluation, and deployment (Shearer, 2000, p. 14). However, the KDD process (Figure 3-1) described in Klösgen & Zytkow (2002) suit better our approach to knowledge creation. It consists of the following steps (Klösgen & Zytkow 2002, p. 10).

1. *Definition and analysis of the business problem* that is targeted to be solved by means of the KDD process. Among business problems that can be addressed via knowledge discovery in large databases there are: marketing applications (predicting and analysing customer behaviour), assessing comparatively countries' economic performance, companies' financial performance benchmarking (gathering and analysing information about competitors), prediction of a portfolio's return on investment, prediction of process variables in production areas (prediction of control variables of a glass manufacturing process), market basket analysis (optimal shelf space allocation, store layout, product location), analysis of exceptions (e.g. for a sales representative the analysis of products and regions that have levels of sales far above or below the average), etc. This step matches Fayad *et al.*'s (1996b) first step of the

KDD process: “developing an understanding of the application domain and the relevant prior knowledge and identifying the goal of the KDD process from the customer’s viewpoint”.

2. *Understanding and preparation of data* imply selection of the target dataset on which the discovery process is to be performed, data cleaning and pre-processing, and data reduction and projection. This step of the KDD process is the most time-consuming: according to Romeu (2001) up to sixty per cent of total project time is dedicated to data preparation, which is, mainly, manual work that is difficult to automate. When selecting the variables (attributes) we should focus on their relevance to the problem at hand. The data from different tables should be pulled together, because “the preponderance of discovery tools apply to single tables” (Klösgen & Zytkow, 2002). Usually, when we construct data-mining models we use two datasets: one for constructing the model (training data set) and another for testing (test dataset). The data-cleaning task is concerned with finding odd and missing values and replacing them with legitimate values. There are several data pre-processing methods that have to be tested to find the proper one for a particular dataset. If the dataset is too large for performing a reasonable mining task, it can be reduced (feature selection, elimination of incomplete observations) or transformed (principal component analysis). In our multiagent-based knowledge creating system (Section 3.4), the data preparation step of the KDD process would be performed automatically by a Data Collection Agent. This step comprises the second (creating the dataset), third (data cleaning and pre-processing), and fourth (data reduction and projection) steps of Fayyad *et al.*’s (1996b) KDD process.

3. *The setup of the search for knowledge* unites steps number five (matching the goal of the KDD process to a particular data-mining task) and six (choosing the data-mining methods for searching for patterns) of Fayyad *et al.*’s (1996b) KDD process. The goal of the KDD process is described in step 1. We will shortly describe the data-mining tasks such as clustering, classification, regression, summarisation, dependency modelling, and change and deviation detection later in Section 3.3.1. Depending on the data at hand and on the business problem that we attempt to solve (goal of the KDD process) we can use a combination of data-mining tasks (e.g. applying clustering to obtain the class variable and, then, classification to model the relationship between the class variable and the dependent variables). The second part of this step is to decide which data-mining method(s) and algorithm(s) are better for performing the search for patterns. At this stage we also have to state more precisely what will be the overall criteria based on which the KDD process will be evaluated (some users are more interested in understanding the model than its predictive capabilities). In this dissertation we look at algorithms and methods from different fields such as statistics, machine learning, neural networks, evolutionary programming, and fuzzy logic to find models/patterns that will address different business problems.

4. *The data-mining (DM)* step is the most important step in the KDD process. It corresponds to step seven of Fayyad's KDD process. The term DM has its roots in statistically oriented data analysis research communities. Actually, the correct term for this KDD step should be Knowledge Mining since we mine for knowledge and not for data (as we mine for gold and other precious metals and not for dirt or rock). However, in this dissertation we use the well-established term DM. DM is defined as "a step in the KDD process consisting of applying data analysis and discovery algorithms that, under acceptable computational efficiency limitations produce a particular enumeration of patterns over the data" (Fayyad *et al.*, 1996a). The user plays an important role at this stage and can help the data-mining method by correctly performing the previous steps.

5. *Interpretation and evaluation of the mined patterns or knowledge refinement* involve the visualisation and interpretation of the extracted patterns/models or of the data covered by the rules extracted. For example, in the case of performance benchmarking through clustering, this step will consist of looking at the economic/financial performance clusters individually and at the characteristics (variables) of each cluster. At this stage predictions can be performed. For example, in assessing countries'/companies' economic/financial performance, the classification models obtained can be applied for newly observed data and the information can be documented and reported to interested parties. This step matches the eighth step of Fayyad's "interpreting mined patterns".

6. *Application of knowledge to the business problems and the consolidation of the discovered knowledge* involve incorporating the knowledge into the organisation's general information system. At this stage, the managers act on the discovered knowledge and use it in the decision-making process. The knowledge obtained can reveal weaknesses and suggest the best course of action that an entity should take so that its performance might improve significantly. This step resembles the ninth step in Fayyad *et al.* (1996b), "acting on the discovered knowledge".

The data-mining step is the core of the KDD process and can be seen as the engine of the "knowledge-creating" machine. Here is the point where the KDD process differs from other analytical tools (query and reporting tools, statistical analysis packages, online analytical processing – OLAP – and visualisation tools). The goal of the KDD process and DM is to *discover* new patterns in data, while most analytical tools are based on *verification* where "the system is limited to verifying user's hypotheses" (Fayyad *et al.*, 1996a). The problem with the verification-based approach is that it "relies on the intuition of the analyst to pose the original question and refine the analysis based on the results of potentially complex queries against a database" (Moxon, 1996). Among the factors that limit the reliability of verification-based analytical tools are the ability of the analyst to pose appropriate questions and to manage the complexity of the attribute space. DM supports the discovery-based approach since "one defining data-mining characteristic is that research hypotheses and relationships between data variables are obtained as a result of (instead of as a condition for) the analyses activities" (Romeu, 2001). *The discovery* goal of the KDD process can be further divided into *prediction,* where the system finds patterns or models for the purpose of future predictions and *description,* where the discovered patterns are presented in a human-understandable way to the user. In this dissertation we combine the two different goals of the KDD process: we are interested in finding both patterns (models) that describe the economic/financial situation of entities as well as models for economic/financial (class) predictions.

In order to fulfil its role DM could perform a number of tasks such as clustering, classification, regression, dependency modelling, summarisation, and change and deviation detection. The link between these tasks and the real-world applications or business problems (the final goal of KDD is to address these problems) is not straightforward, because real-world applications rarely have a simple single solution. Many different tasks may match a particular application, depending on how one approaches the problem (Smyth, 2002). For example, one real-world application would be to assess companies' financial performance from a particular sector. Treating our problem as a supervised learning task implies that we already have financial performance classes for all the observations used to train the classifier. Actually there are no labelled data available, thus, the class variable has to be created at the beginning, by treating our problem as an unsupervised task. Only after the class variable has been constructed, can a classifier be trained. Smyth (2002) pinpoints various advice worth consideration when linking real-world applications with the data-mining task. The author states that it is advisable to start with only one task to address a real-world application and, only if necessary, add more complex ones. He also suggests removing the irrelevant details of the original formulation of the problem so that it resembles more closely a standard textbook task description. In order to select the proper task for a given problem, the data miner should have a complete understanding of both the business problem addressed and the task linked to it. Finally, Smyth (2002) states that it is better to approximate the solution to the right problem than it is to solve the wrong problem exactly.

3.3.1 DM Tasks and Algorithms

There are many different tasks that can be performed by means of the KDD process and, even though they are not denominated in the same way by all authors, the following tasks are common to many of them (Fayyad *et al.*, 1996a; Romeu, 2001, Klösgen & Zytkow, 2002):

1. *Clustering*. The term "clustering" is used in many research communities to describe methods for grouping unlabelled data (Jain *et al.*, 1999). Traditional clustering methods intend to identify patterns in data and create partitions with different structures. These partitions are called clusters, and elements within each cluster should share similar characteristics. The partitions can be mutually

exclusive (disjoint) or may contain observations that belong in some degree to several clusters (overlapping). The standard application of clustering in business has been consumer behaviour analysis where clusters are constructed with consumers that have similar purchasing characteristics. In this thesis our focus is on clustering for performance benchmarking. In Sections 4.1.2 and 4.1.4 we present the related research in applying clustering techniques to address economic/financial performance benchmarking. Clustering is also known as unsupervised classification.

2. *Classification*. According to Bock (2002) there are three approaches related to classification:
    a. *Classification as an ordering system for objects* (e.g. classification of books in a library, the ordering of chemical elements in the periodic system, classification of products and merchandise for international standardisation). This approach is beyond the scope of this dissertation.
    b. *Classification as a class assignment or supervised learning (learning with a teacher)*. This approach corresponds to the common view of the classification task: a learning function that maps a data item (observation) into one of several predefined classes (Hand *et al.*, 2001). In this case, classification models (classifiers) are built with which new observations can be assigned different classes. For example in medicine a disease can be recognised based on patient symptoms, in performance benchmarking countries/companies can be classified according to their economic/financial performance, in marketing good consumers can be identified by their purchasing characteristics, etc.
    c. *Classification as class constructing or clustering or unsupervised learning (learning without a teacher)*. Clustering is discussed above – point 1.

Clustering and supervised learning can be combined when class variables are not available to obtain *hybrid* classifiers. Throughout the research we addressed business problems by both simplifying them to a single data-mining task and also by matching them with different data-mining tasks when necessary.

3. *Regression* is the process of learning a function that maps a data item to a real-value prediction variable and the discovery of functional relationships between variables (Fayyad *et al.*, 1996a). Classification can be considered as a particular case of regression analysis where the outcome is a discrete value (class). In regression we try to find a function that links an output (or many) to a number of inputs. These functions range from very simple ones (linear, one input) to very complex (non-linear, many inputs) leading to three different regression models: standard linear model, generalised linear model, and generalised additive model. The standard linear model links the outputs to the inputs with a function that is a linear combination of the inputs. The generalised linear model is applied predominantly to perform classification tasks since the outcome values are constrained to a sensible range. For example

the logit function (see Section 5.2.1) derives expected values between zero and one. The generalised additive models can accommodate the non-linear effects of the original inputs. The standard classic approach to model fitting in regression is called maximum likelihood estimation (MLE). MLEs are estimates that maximise the likelihood function, which is the joint probability density of the data (Rao & Potts, 2002). In addition to standard statistical regression techniques, ANNs proved to be very useful. The main advantages of neural approaches over traditional ones to regression are: ANNs are free of any distributional assumptions, are universal approximators, can handle the inter-correlated data, and they provide a mapping function from the input to the outputs without any a priori knowledge about the function form (function approximation capability) (Hornik *et al.*, 1989; Basheer & Hajmeer, 2000).

4. *Dependency modelling* – is concerned with finding models that describe significant dependencies between variables. At the structural level the dependency model specifies which variables are dependent on each other, while at the quantitative level the model specifies the strengths of the dependencies (Fayyad *et al.*, 1996b). Probabilistic and causal networks (Spirtes, 2002) are two techniques that are increasingly applied to performing this data-mining task.

5. *Summarisation* consists of methods for finding a compact description of a subset of data. Among these methods there are: calculation of standard deviation and means for the observations, derivation of summary rules, multivariate visualisation techniques, and discovery of functional dependencies between variables (Fayyad *et al.*, 1996b).

6. *Change and deviation detection* involves finding the differences between current data and previously measured or normative values. Change detection deals with analysing change (one entity observed at two points of time) or trend (a sequence of equidistant points of time) over the dataset. Deviation analysis starts with identifying the deviating sub-groups (sub-groups where the target variable differs significantly from its expected value in relation to the input values from that particular sub-group) and rely on hypothesis testing to test whether the sub-group is interesting or not. Generally, the rejected null hypothesis assumes an uninteresting, non-deviating sub-group. Klösgen & Anand (2002) call this data-mining task *sub-group discovery.*

The algorithms used to perform data-mining tasks described above are numerous and they come from different research fields (statistics, machine learning, artificial intelligence, fuzzy logic, etc.). Romeu (2001) groups data-mining algorithms in three categories: mathematically based, statistically based and “mixed” algorithms.

Mathematically based (*deterministic*) algorithms include mathematical programming (linear, non-linear, integer), network methods (link[8] and affinity[9] analysis), and memory-based reasoning approaches (nearest-neighbour classifiers).

Statistically based (*stochastic*) algorithms include traditional statistics regression, discrimination techniques (linear discriminants, quadratic discriminants, logistic discriminants or logistic regression), statistical time series analysis, factor analysis[10], etc.

The difference between mathematical and statistical algorithms lies in the approach that they are based upon: mathematical models are deterministic (random phenomena are not involved and these models produce the same output for a given starting condition), while statistical ones are stochastic (based on random trials). Although some of them were recently employed in solving data-mining tasks (Bradley *et al*., 1999), it is beyond the scope of this dissertation to study the mathematically based approaches.

"Mixed" algorithms borrow heavily from both, the algorithmic and the stochastic components (Romeu, 2001). Romeu includes here: clustering methods, induction techniques such as decision trees, neural networks, and genetic algorithms. We introduced these techniques in Chapter 1 as CI methods. In this dissertation we explore and combine statistically based and CI methods to address some business problems. We match the business problems with different data-mining tasks. In Chapter 5 we present the CI methods used in this study, starting with clustering methods such as SOM, C-Means, FCM and a newly developed Weighting FCM algorithm (which perform the clustering task). Then, we present some classification methods such as multinomial logistic regression, Quinlan's algorithm for decision-tree induction, artificial neural networks for supervised learning, and genetic algorithms for learning the weights of an ANN (which performs the DM classification task). We match the business problem of countries'/companies' economic/financial performance benchmarking with both DM clustering and classification tasks. Next, Chapter 5 presents an ANN forecasting method that uses a retraining procedure for learning the weights. With this method we perform the regression task associated with the prediction of process variables business problem.

Whatever the algorithm we use to perform the data-mining tasks, we need criteria to evaluate its performance to be able to rigorously compare it with other approaches. For our models we used quantitative criteria such as quantisation error,

[8] The data are represented as a network, where the nodes are pieces of information and the links provide the order (sequence) in which they appear. In this way patterns of behaviour in the system entities are constructed. Visualisation plays an important role in visualizing these networks.

[9] Market basket analysis is the classical example in affinity analysis which seeks to identify the associated products that consumers buy (consumers' purchasing behaviour)

[10] Factor analysis as well as Principal Component Analysis transforms the input space into a smaller, uncorrelated space. The drawback is that the reduced model explains less problem variability (Romeu, 2001).

accuracy rate or mean square error, or qualitative ones such as fidelity with real-world phenomena, form and content, and richness of knowledge in the form of class predictions.

3.4 A Conceptual Model for a Multi-Agent Knowledge-Building System

In this Section we introduce a conceptual model for a knowledge-building system based on a society of software agents (Figure 3-2). Software agents are computational programs or entities situated in a computing environment and that assist users with computer-based tasks. They act to accomplish specialised tasks on behalf of users and act towards reaching certain user-specified or automatically generated goals with a certain degree of autonomy and flexibility (Jennings and Wooldridge, 1998). The idea of using software agents for decision support is not new. Wang *et al*. (2002) proposed a society of software agents for monitoring and detecting financial risk. Liu (1998, 2000) explores the application of software agent technology in an environmental scanning support system and in building the information infrastructure for strategic management support systems. Two different software agent systems (EdgarScan developed by PriceWaterhouseCoopers and FRAANK proposed in Nelson *et al*., 2000) have been developed independently to retrieve financial information from the EDGAR SEC[11] database. However, these agent systems are limited in scope: they retrieve information rather than process it or use basic information-processing methods such as ratio calculations. The knowledge-building system proposed in Figure 3-2 supports the entire collect-process-analyse-disseminate information cycle by using separate software agents for each activity.

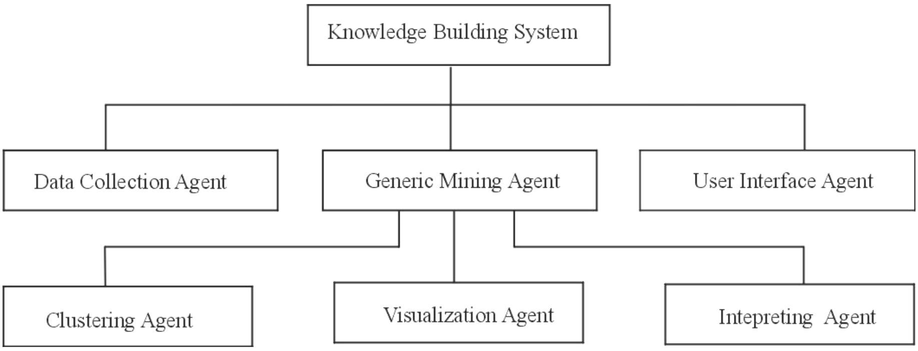


Figure 3-2 Architecture of the Knowledge-Building System.
(Source: *Publication* 1)

The system integrates an agent that collects the data from various sources (*Data Collection Agent*), one that mines the data using qualitative or quantitative methods

[11] EDGAR SEC = Electronic Data Gathering, Analysis, and Retrieval system for managing the Securities and Exchange Commission's financial fillings.

(*Generic Mining Agent*) and another that communicates the findings in an easy-to-understand fashion (*User Interface Agent*). The system could be used to perform different data-mining tasks such as clustering, classification, and regression, which match, among others, the following business problems: countries'/companies' economic/financial performance benchmarking and prediction of process control variables.

Depending on what mining techniques and data are used, there are two main instances of the Generic Mining Agent: the *Data-Mining Agent* and the *Text-Mining Agent*.

The *Data-Mining Agent* (Figure 3-3) processes the numerical data. For example, if our goal is to assess financial performance of, say, telecommunications companies the Data-Mining Agent should provide the Knowledge-Building System with the cluster that a company belongs to using the *Data-Clustering Agent*. Further, a classifier can be constructed by first visualizing the clustering results (with the *Data-Visualisation Agent*) and then by putting the *Data-Classification Agent* to work. The Data-Classification Agent should apply different techniques such as decision trees, multinomial logistic regression and/or a supervised neural network for classifying the data and should use the model that achieves the highest accuracy in training and the best prediction performance. Then, the *Data-Interpretation Agent*'s job would be to explain the findings.

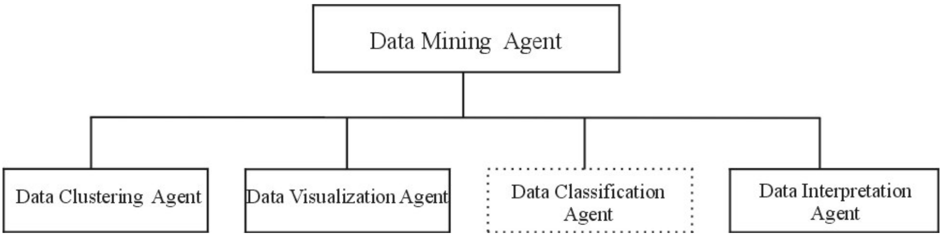


Figure 3-3 Data-Mining Instance of the Generic Mining Agent
(Source: *Publication* 1)

The *Text-Mining Agent* has the same functionality as the Data-Mining Agent but processes text data. The Text-Mining Agent would include the same sub-agent as the DataMining Agent, but it might use some specific sub-agents. One such agent would be the *Summarisation Agent,* which would extract the most relevant sentences from large amounts of text data so that different text-analysis methods such as the prototype-matching method (Visa *et al*., 2002; Back *et al*., 2001) could be efficiently applied.

The limitations of such a complex knowledge-building system can be analysed from two perspectives: the limitations specific to each individual agent and those specific to the integration/communication among agents. For example, the Data-Collection Agent is limited in its activity because of the lack of standardised financial reporting. Currently, there is no way for collection agents to automatically retrieve financial data from diverse web sites without specifically coding the agent for a specific page (Debreceny & Gray 2001). The introduction of a new financial reporting language (XBRL – eXtensible Business Reporting Language) will ensure the efficiency of the Data-Collection Agent. Other limitations are concerned with the techniques used by the agents. For example, SOM results are difficult to validate. Wang (2001) addresses this issue by proposing a number of techniques for verifying clustering results. We present in *Publication* 3 some ways for validating the SOM results. Text-mining techniques have some disadvantages caused by the complexity, synonymy and polysemy of the text data.

The main integration limitation consists of finding a standard way of collecting, analysing, and communicating numeric and text data. The User Interface Agent will have to combine the results obtained by both Data- and Text-Mining Agents and show how these results complement each other. A standard way to present the results of both data- and text-mining methods is difficult to implement.

In this dissertation we are concerned with quantitative (numeric) data mining, and not text mining. In other words, we use quantitative data in our experiments and, consequently, methods that are more specific to quantitative data mining. Typically, the knowledge-building system would include the whole KDD process.

Chapter 4 Areas of Applicability of Quantitative Data Mining

In this chapter we discuss three areas of the applicability of data mining, namely: countries' economic performance benchmarking, companies' financial performance benchmarking, and the prediction of process control variables and present related work that has been carried out addressing the above business problems.

As we mentioned in Chapter 3, Section 3.3, the knowledge discovery process can have two goals: description and prescription. In this dissertation we use, separately or in combination, both descriptive and prescriptive techniques. The different business problems are addressed with methods from the two different categories. For example, for countries'/companies' economic/financial benchmarking we use descriptive clustering methods such as SOM or fuzzy clustering in combination with prescriptive classification methods such as logistic regression, decision trees and artificial neural networks. However, this division between descriptive (clustering) and prescriptive (classification and regression) methods is not strict. Some authors (e.g. Tan *et al.*, 2002) consider that clustering techniques have prescriptive properties as well. We agree with them in the sense that one can use distances to the centres of the clusters to place (predict) the classes for newly observed cases, without constructing a classification model. However, as new instances are placed in the clusters, the characteristics of the clusters might change, thus inducing errors when calculating the distances to the clusters' centres and further diminishing the prescriptive properties of the clustering.

4.1 Benchmarking

There are many definitions of *benchmarking*, but briefly it includes activities such as comparing, learning and adopting best practices that can increase performance. Benchmarking is simply about making comparisons with other organisations and then learning the lessons that those comparisons throw up (The European Benchmarking Code of Conduct, 2005). Benchmarking refers to the process of comparing entities' performance based on some common performance measures. Usually, the entities are companies or organisations. In this study, we consider two types of entities: countries and companies. We benchmark countries as to their economic performance and companies as to their financial performance.

There are many advantages of using benchmarking. According to APQC (2005) some advantages can be: to improve profits and effectiveness, accelerate and manage change, set stretch goals, achieve breakthroughs and innovations, create a sense of urgency, overcome complacency or arrogance, understand world-class performance, and make better-informed decisions. Of the Fortune 500 best-ranked companies more than 70 per cent use some form of benchmarking on a regular basis (Greengard, 1995). Researchers have paid great attention to benchmarking, too: Dattakumar & Jagadeesh (2003) report 350 publications related to benchmarking as of June 2002. However, the complexity of the benchmarking process makes it hard to implement by smaller entities. In a survey on the APQC website, most of the respondents (69%) answered that they spent only a small amount of time (0-20%) on benchmarking activities.

Depending on the scope of the benchmarking there are four types of benchmarking process: *internal*, *competitor*, *functional* and *generic* benchmarking (Bendell *et al.*, 1998; Camp, 1989). At the same time, depending on the goal of the benchmarking, there are three types of benchmarking: *performance benchmarking*, *process benchmarking* and *strategic benchmarking* (Bhutta & Huq, 1999).

Internal benchmarking is the process by which comparisons are made between parts of the same entity. This is one of the easiest benchmarking investigations and is suitable for international firms that have similar operating units in different locations. However, the outcome of such benchmarking is unlikely to yield results that are world best practices (Bendell *et al.* 1998, pp. 82-84).

Competitor benchmarking is the process of comparing the performances of direct competitors. The comparability of measures (e.g. size) used in the benchmarking process deserves high consideration (Camp 1989, p. 63). Here the entities compete in the same area (e.g. different countries compete for EU accession or to attract foreign capital, different companies compete on the same market), making comparability achievable.

Functional benchmarking includes comparisons between indirectly competing entities. In the case of companies' performance benchmarking, there is a great potential for identifying functional competitors or industry leader firms to benchmark even if in dissimilar industries. This would involve, in the case of logistics, identifying those firms that are recognised as having superior logistics functions wherever they exist (Camp 1989, pp. 63-64).

Generic benchmarking involves comparisons of generic processes across heterogeneous entities. This is the most pure form of benchmarking, is the most difficult concept to gain acceptance and use but probably that with the highest long-term payoff (Camp 1989, p. 65).

Performance benchmarking is concerned with comparing performance measures to determine how well one entity is performing as compared to others.

Process benchmarking involves comparing the methods and processes used across different entities.

In *strategic benchmarking* the comparisons are made in terms of strategies (e.g. policies, direction, long vs. short term investment, etc).

The pairs' suitability between goal-oriented and scope-oriented types of benchmarking is shown in Table 4-1.

Table 4-1 Suitability of goal and scope-oriented types of benchmarking

	Internal benchmarking	Competitor benchmarking	Functional benchmarking	Generic benchmarking
Performance benchmarking	Medium	**High**	Medium	Low
Process benchmarking	Medium	Low	High	High
Strategic benchmarking	Low	High	Low	Low

(Source: Bhutta & Huq, 1999, originally adapted from McNair & Liebfried, 1992)

As Table 4-1 shows some scope-goal benchmarking pairs have more relevance than others. For example, it is irrelevant to compare the entity's strategy with itself, whereas comparing the performance and strategies of different competitors in the same area should be relevant. The type of benchmarking process that constitutes the business application in this dissertation is at the intersection of performance and competitor benchmarking (the bold word in Table 4-1).

We narrow even more our benchmarking applications by only looking at *economic* performance measures in the case of countries and at *financial* performance measures in the case of companies. We call such a benchmarking *economic/financial performance competitor benchmarking*. In economic performance competitor benchmarking we analyse the differences between countries from one geopolitical area with respect to macro-economic variables. In financial performance competitor benchmarking we are interested in finding the gaps in the performance of different companies from the same industry with regard to four financial performance measures: profitability, liquidity, solvency, efficiency.

4.1.1 Economic Performance Competitor Benchmarking

One real-world application addressed in this dissertation is to assess comparatively the economic performance of EU candidate and non-EU countries against the newly accepted ones. Using CI methods we try to position, in time, already accepted countries and those aspiring to join. This type of analysis can benefit the countries involved, EU in its monitoring process, business players such as international companies that want to expand their business and individual investors.

The choice of the indicators for analysing comparatively countries' economic performance is not trivial. Depending on the goal of the analysis, there are two sets of indicators that may be employed: macro-economic and micro-economic indicators. We chose for our study the macro-economic indicators since they synthesise the economies of the countries involved. There are two reasons why we preferred macro-economic instead of micro-economic indicators to compare the economic performance of the countries involved. Firstly, macro-economic data are easier to obtain and there is a standard way of calculating these indicators, which makes the data comparable. Secondly, one may argue that the use of micro-economic indicators (which are extracted from companies' financial statements) does not represent the real performance of the country where companies perform their current activities. The working capital of a foreign multi-national company that operates in one country does not necessarily reflect the economic strength of that specific country.

In Chapter 6, Section 6.1 we present in detail the economic dataset used to assess comparatively countries' economic performance and the rationale behind the choice of variables.

4.1.2 Related Research in Countries' Economic Performance Benchmarking

Comparisons of countries based on their economic and social performances have been conducted before. However, in the majority of cases only descriptive techniques (e.g. SOM) have been used to find differences and similarities in economic performance of different countries.

The Global Competitiveness Report 2004-2005 (Porter *et al.*, 2004) ranks 104 world-wide economies according to an index called the *Growth Competitiveness Index (GCI)*. This index is a composed index and includes three different contributors to the economic growth: the quality of the macroeconomic environment, the state of the country's public institutions, and the country's technological readiness. The economies are separated into two groups: countries for which technological innovation is critical for growth (*core* economies – e.g. Sweden, USA), and countries for which the adoption of technologies developed abroad is critical for growth (*non-core* economies – e.g. Czech Republic, Romania). The GCI is calculated differently for the two groups: in the case of core economies, the technological sub-index is given more weight (1/2) than for the non-core economies (1/3). The other two sub-indexes (macroeconomic quality sub-index and institutional stability sub-index) have the following weights: 1/4 and 1/4 for core economies and 1/3 and 1/3 for the non-core economies respectively. In order to compare the countries' productivity another index is constructed: *Business Competitiveness Index (BCI),* which measures the wealth of companies within each country. The data necessary to calculate the indexes include hard and survey data. The hard data come from each country' national statistics office, while the survey data are collected as the result of an annual Expert Opinion Survey. In 2004 there were 8700 business respondents to the survey. For 3 years in a row Finland was ranked the first country according to the GCI, followed by USA, Sweden and Denmark in 2004. With respect to BCI the first performer in 2004 was USA, followed by Finland, Germany and Sweden.

Some related research studies that applied SOM to assess comparatively countries' performance are: Kaski & Kohonen (1996), Ong & Abidi (1999), and Serrano Cinca (1998a). Basically, in the above studies, SOM maps were constructed based on some key economic or social indicators.

Kaski & Kohonen (1996) apply SOM to analyse comparatively world countries with respect to welfare and poverty. The authors used 39 indicators (taken from the World Development Report) that describe factors such as health, education, consumption, and social services. With the aid of these indicators structures of welfare and poverty were revealed by the SOM. In all, six distinct regions were identified on the map based on the projections of individual variables. The best performers were OECD countries. The Eastern European countries formed the second compact region close to the OECD countries. Region 3 included countries from South America and was considered the second best performer region. The Asian (region 4) and African countries (region 5 and 6) were the worst performers. The SOM "welfare map" was found to correlate strongly with the GDP per capita variable, which was not used in constructing the map.

Ong & Abidi (1999) apply SOM to a 1991 World Bank dataset that contains 85 social indicators in 202 countries finding clusters of similar performance. Here, the different performance regions were constructed objectively by applying different clustering techniques (C-Means, C nearest neighbour) on the trained SOM.

Serrano Cinca (1998a) performs two SOM experiments to compare the EU member countries' performance. In the first experiment, the author compares 15 EU member countries using 4 macro-economic variables constructed along the guidelines proposed in the Maastricht Treaty: rate of inflation, national debt, interest rate, and deficit. By analysing the synaptic weights (SOM weights) the author finds, for each neuron in the SOM, the most important variable. Then, different regions are formed with countries that have high interest rates, low deficit, high deficit, etc. The second experiment uses micro-economic data (data taken from companies' balance sheets) to compare the performances of EU countries. The data were taken from the BACH database (BACH, 2005), which contains homogenised aggregate financial data for 10 EU countries and is maintained by the European Commission. In total 16 ratios were used to explain countries' performance and they measured the financial results, relative costs and financial structures of the companies. Again, the synaptic weights were used to find the relative importance of each ratio in every neuron and, as in the first experiment, regions of high financial charges, low provisions, etc. were revealed. The two experiments show similar results in terms of countries' positions, thus validating each other.

Drobics *et al*. (2000) used SOM to analyse USA macroeconomic data recorded between 1963 and 1985. The goal of the paper was to find regions of interest in the data and to label them. The authors developed fuzzy rules from a clustering obtained using the SOM algorithm. The confidence for each rule was above 75 per cent.

Kasabov *et al*. (2000) proposed a hybrid intelligent decision-support system (HIDSS) for helping decision making in a complex, dynamic environment. Three computational models are mixed in the HIDSS: (1) the Repository of Intelligent Connectionist-Based Information Systems (RICBIS) is a conglomerate of computational intelligence techniques such as Fuzzy Neural Networks, SOM, MLPs, (2) Evolving Fuzzy Neural Networks (EFuNN) that can learn more quickly in an incremental, adaptive way through one-pass propagation of any new data examples, and (3) the Evolving Self-Organizing Map (ESOM) uses a learning rule similar to SOM, but its network structure is evolved dynamically from input data. The aim of the paper is twofold: to develop a computational model for analysing and anticipating signals of sudden changes of volatility in financial markets, and to study phenomena pertaining to economic performance in the European Monetary Union (EMU) area.

In our experiments, we go one step further and model the relationships between the class variable (which shows the economic positions of countries) and input ratios by constructing economic classification models. In addition to their descriptive properties these models have prescriptive ones that allow us to predict countries' future economic performance.

4.1.3 Financial Performance Competitor Benchmarking

The second and most important business application addressed in this dissertation consists of analysing comparatively the financial performance of companies from certain industries.

We use financial ratios to assess their financial performance. Foster (1986, p. 96) presents the reasons for using financial ratios instead of absolute value indicators:

- to control the effect of size differences across firms or over time;
- to make the data more suitable to statistical methods;
- to probe a theory in which a ratio is the variable of interest;
- to exploit an observed empirical regularity between a financial ratio and the estimation or prediction of a variable of interest.

The first reason has a direct implication for our study in the sense that we use financial ratios to make the financial performance of different-sized companies from our experiments comparable. Usually, financial ratios are created by relating the absolute values of financial items to common bases such as total assets and sales (Lev 1974, p. 34).

In Lehtinen's (1996) study of the reliability and validity of financial ratios in international comparisons, the author proposes the financial ratios based on which one could assess comparatively the financial performance of different companies.

The *validity* of a ratio is the extent to which the ratio measures what it is intended to measure. For example, Operating Margin ratio is intended to measure profitability. However, the capital, which according to the definition of profitability should be taken into account in profitability ratios, is not included in the Operating Margin ratio making the validity of this ratio relatively low. The Return on Equity ratio includes both revenue and capital values making the ratio's validity higher (Lehtinen 1996, p. 9). *Reliability* refers to the extent to which the ratios values can be manipulated through accounting policies. A financial ratio that is easily manipulated by the accounting policy is not very reliable (Lehtinen 1996, p. 9). International accounting differences occur as a consequence of different accounting practices related to the depreciation method used, how the inventory is valued (LIFO vs. FIFO), the capitalisation or not of leases, research and development costs, and goodwill, revaluation of fixed assets, provision and reserves[12], deferred taxation. A more detailed explanation of these accounting differences can be found in Lehtinen (1996, pp. 53-60). A more recent book on international accounting comparison is Nobes & Parker (2002), which provides in Chapter 2 (pp. 17-32) a detailed discussion of the causes of international accounting differences and in Chapter 3 (pp. 34-51) discusses major international differences in financial reporting.

In Chapter 6, Sections 6.2, 6.2.1 and 6.2.2 we present in detail the pulp-and-paper and telecom datasets used to assess companies' financial performance.

4.1.4 Related Research in Companies' Financial Performance Benchmarking

The research literature assessing comparatively companies' financial performance is relatively rich. We include here companies' financial benchmarking, companies' failure prediction, companies' credit/bond rating, analysis of companies' financial statement, and analysis of companies' financial text data.

As in the case of assessing countries' performance, SOM was used extensively in assessing comparatively companies' financial performance. There are two pioneer works of applying the SOM to companies' financial performance assessment. One is Martín-del-Brío & Serrano Cinca (1993) followed by Serrano Cinca (1996, 1998a, 1998b). Martín-del-Brío & Serrano Cinca (1993) propose Self Organizing Maps (SOM) as a tool for financial analysis. The sample dataset contained 66 Spanish banks, of which 29 went bankrupt. Martín-del-Brío & Serrano Cinca (1993) used 9 financial ratios, among which there were 3 liquidity ratios: current assets/total assets, (current assets – cash and banks)/total assets, current assets/loans, 3 profitability ratios: net income/total assets, net income/total equity

[12] Cf. Lehtinen (1996, pp. 59-60) the term "provisions" is used in countries where corporate taxation and official financial statements are linked to each other (e.g. Germany), while the term "reserves" is used in countries where corporation taxation is not based on official financial statement figures (e.g. USA, UK, the Netherlands)

capital, net income/loans, and 3 other ratios: reserves/loans, cost of sales/sales, and cash flows/loans. A solvency map was constructed, and different regions of low liquidity, high liquidity, low profitability, high cost of sales, etc. were highlighted on the map. Serrano Cinca (1996) extends the applicability of SOM to bankruptcy prediction. The data contain five financial ratios taken from Moody's Industrial Manual from 1975 to 1985 for a total of 129 firms, of which 65 are bankrupt and the rest are solvent. After a preliminary statistical analysis the last ratio (sales/total assets) was eliminated because of its poor ability to discriminate between solvent and bankrupt firms. Again, a solvency map is constructed and, using a procedure to automatically extract the clusters different regions of low liquidity, high debt, low market values, high profitability, etc. are revealed. Serrano Cinca (1998a, 1998b) extended the scope of the Decision Support System proposed in the earlier studies by addressing, in addition to corporate failure prediction, problems such as: bond rating, the strategy followed by the company in relation to the sector in which it operates based on its published accounting information, and comparison of the financial and economic indicators of various countries. Our study is more related to these later studies in the sense that we perform a variety of SOM-related experiments as well.

The other major SOM financial application is Back *et al.* (1998), which is an extended version of Back *et al.* (1996a). Back *et al.* (1998) analysed and compared more than 120 pulp-and-paper companies between 1985 and 1989 based on their annual financial statements. The authors used 9 ratios, of which 4 are profitability ratios (operating margin, profit after financial items/total sales, return on total assets, return on equity), 1 is an indebtedness ratio (total liabilities/total sales), 1 denotes the capital structure (solidity), 1 is a liquidity ratios (current ratio), and 2 are cash flow ratios (funds from operations/total sales, investments/total sales). The maps were constructed separately for each year and feature planes were used to interpret them. An analysis over time of the companies was possible by studying the position each company had in every map. As a result the authors claim that there are benefits in using SOMs to manage large and complex financial data in terms of identifying and visualizing the clusters.

Eklund *et al*. (2003) investigate the suitability of SOM for financial benchmarking of world-wide pulp-and-paper companies. The dataset consists of 7 financial ratios calculated for 77 companies for six years (1995-2000). In our study we used the same dataset when performing the pulp-and-paper industry experiment (Section 6.2.1). Eklund *et al*. (2003) construct a single map for all the years and find clusters of similar financial performance by studying the feature plane for each ratio. Next, the authors used SOM visualisation capabilities to show how the countries' averages, the five largest companies, the best performers and the poorest performers evolved over time according to their position in the newly constructed financial performance clusters. Karlsson *et al.* (2001) used SOM to analyse and compare companies from the telecommunications sector. The dataset consists of 7 financial ratios calculated for 88 companies for five years (1995-1999). In this study we use the same dataset updated with complete data for 2000, 2001 and with available data for 2002 when performing our telecom sector experiment (Section 6.2.2). Karlsson *et al.* (2001) used a similar approach to Eklund *et al.* (2003) and built a single map. The authors identify six financial performance clusters and show the movements over time of the largest companies, countries' averages and Nordic companies. Both, Eklund *et al.* (2003) and Karlsson *et al.* (2001) used quantitative financial data from the companies' annual financial statements. The ratios were chosen based on Lehtinen's study (Lehtinen, 1996) of the validity and reliability of ratios in an international comparison. Kloptchenko (2003) used the prototype matching method (Visa *et al.*, 2002; Toivonen *et al.*, 2001; Back *et al.*, 2001) to analyse qualitative (text) data from telecom companies' quarterly reports. Kloptchenko *et al.* (2004) combined data and text-mining methods to analyse quantitative and qualitative data from financial reports, in order to see if the textual part of the reports can offer support for what the figures indicate and provide possible future hints. The dataset used was that from Karlsson *et al.* (2001).

The use of fuzzy clustering – especially the Fuzzy C-Means (FCM) algorithm – in assessing comparatively companies' financial performance is relatively scarce. The fuzzy logic approach can also deal with multidimensional data and model non-linear relationships among variables. It has been applied to companies' financial analysis, for example, to evaluate early warning indicators of financial crises (Lindholm & Liu, 2003).

One of the pioneer works in applying *discriminant analysis* (DA) to assessing comparatively companies' financial performance is Altman (1968). Altman calculated discriminant scores based on financial statement ratios such as working capital/total assets; retained earnings/total assets; earnings before interest and taxes/total assets; market capitalisation/total debt; sales/total assets. Ohlson (1980) is one of the first studies to apply *logistic regression* (LR) to predicting the likelihood of companies' bankruptcy. Since it is less restrictive than other statistical techniques (e.g. DA) LR has been used intensively in financial analysis (see Section 5.2.1). De Andres (2001, p. 163) provides a comprehensive list of papers that used LR for models of companies' financial distress.

Induction techniques such as Quinlan's C4.5/C5.0 decision-tree algorithm were also used in assessing companies' financial performance. Shirata (2001) used a C4.5 decision-tree algorithm together with other techniques to tackle two problems concerning Japanese firms: prediction of bankruptcy and prediction of going concern status. For the first problem, the authors chose 898 firms that went bankrupt with a total amount of debt more than ¥10 million. For the going concern problem 300 companies were selected out of a total of 107,034 that have a stated capital of more than ¥30 million. The financial ratios used were: retained earnings/total assets, average interest rate on borrowings, growth rate of total assets, and turnover period of accounts payable. As a conclusion of the study, the author underlines that decisions concerning fund raising can create grave hazards to business and, therefore, in order to be successful, managers have to adapt to the changing business environments.

Supervised learning ANNs were extensively used in financial applications, the emphasis being on bankruptcy prediction. A comprehensive study of ANNs for failure prediction can be found in O'Leary (1998). The author investigates fifteen related papers for a number of characteristics: what data were used, what types of ANN models, what software, what kind of network architecture, etc. Koskivaara (2004a) summarises the ANN literature relevant to auditing problems. She concludes that the main auditing application areas of ANNs are as follows: material error, going concern, financial distress, control risk assessment, management fraud, and audit fee, which are all, in our opinion, linked with the financial performance assessment problem. Coakley & Brown (2000) classified ANN applications in finance by the parametric model used, the output type of the model and the research questions.

4.2 Process Variables Prediction

In this Section we present our third business problem that can be solved by means of data-mining techniques: prediction of process control variables.

The basic problem in predicting process control variables is to find a model that maps some real-numbered inputs to some real-numbered outputs so that the prediction performance of the model is maximised. For example, Figure 4-1 depicts the relationship between 29 input and 5 output variables of a glass manufacturing process. The output variables are control variables (e.g. temperatures) that measure the quality of the glass at different points in the melting tank. The glass quality depends in a complicated manner on many input variables, some of which are measurable (e.g. energy entries, raw material components, environmental temperatures) and some non-measurable (e.g. corrosion effects). Usually, the state of the melt (the quality of the molten glass) is determined with the help of physically based mathematical models such as continuum simulations (Lankers & Strackeljan, 2004). However, these models have two main disadvantages: they require very large computation times, and they work within idealised boundary conditions and cannot take into account many of the disturbances affecting a real melting tank in a production environment (Lankers & Strackeljan, 2004).

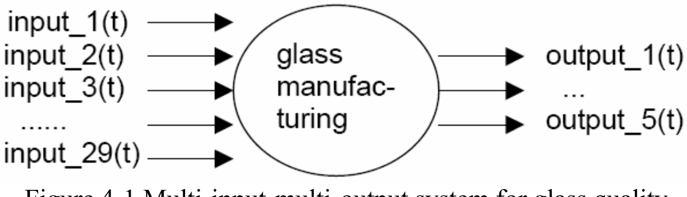


Figure 4-1 Multi-input-multi-output system for glass quality
(Source: EUNITE Competition, 2003)

There are certain common characteristics of process engineering applications, which make them difficult to model. First of all, the process data contain full noise and uncertainty as a result of the difficulty of measurement and data collection.

Another characteristic is the way the output variables depend on the input variables. For example, in the case of glass manufacturing, changing an input variable may result in an output change starting only a couple of hours later and going on for up to several days (EUNITE Competition, 2003). Koskivaara (2004b) found a similar behaviour to financial statement accounts: a value of an account *x* at moment *t* depends on some of its past values: $x(t\text{-}1)$, $x(t\text{-}2)$, etc. The third difficulty comes from the non-measurable influences that can affect the process. In this case, the model has to be able to adapt to process changes. Fourthly, data in engineering applications are characterised by the high dimensionality of inputs that requires compression techniques (e.g. Principal Component Analysis – PCA – Bishop (1995)).

We address the process variables prediction problem through ANN models. An important advantage of ANNs over traditional statistical and econometric models is the fact that ANNs are free from any data distribution assumptions (Hornik *et al*., 1989). At the same time, once a reliable ANN model has been constructed for a particular problem, it can be applied with small changes to other similar problems. Moody (1995) describes problems that the expert faces when constructing time series prediction models such as noise, non-stationary and non-linearity. The noise distributions are typically heavy-tailed and include outliers (Moody 1995, p. 2). A time series $x(t),\, t = 1,\ldots,N$ is non-stationary if the joint probability distribution of the sub-set $x(t), x(t+1),\ldots, x(t+m-1)\ \forall m$ is dependent on the time index *t*. Non-linearity can be handled with ANNs by finding the proper number of hidden layers and hidden nodes that will best disseminate the input space.

In Chapter 6, Section 6.3 we present in details our experiments in modelling the glass-manufacturing process at Schott, a German-based company.

4.3 Related Research in Prediction of Process Control Variables

There are countless research studies that apply data-mining techniques to engineering applications. For a comprehensive review of ANN for industrial applications see Meireles *et al*. (2003).

The data-mining tasks used for engineering applications are mainly classification and regression. The classification task matches diagnostic applications in which the various engineering machines are classified into categories associated with different faults. Loskiewicz-Buczak & Uhrig (1992) developed a neural network diagnosis model for a steel sheet-manufacturing plant. The model was used to classify the steel-manufacturing machines based on fault categories. Alhoniemi (2000) analyses the process data of a continuous vapour phase digester (at the UPM Kymmene pulp-and-paper company) using SOMs. The author states that by interpreting the SOM visualisations the reasons for the digester faults can be determined. Moczulski (2002) implemented a diagnostic expert system for rotating machinery.

The regression task matches engineering applications such as city water consumption prediction (An *et al.*, 1997), prediction of the remaining useful life of a piece of equipment (Hansen *et al.*, 1996), etc.

A particular set of engineering applications is represented by process control applications. Some exemplary applications are: sound and vibration control applications (Widrow & Stearns, 1985), vehicular trajectory control applications (Nguyen & Widrow, 1989; Pallet and Ahmad, 1992), fighter flight control applications (Schwartz, 1992; Schwartz & Treece, 1992), train control (Khasnabis *et al.*, 1994), cement production control (Mrozek & Plonka, 1998), manufacturing cell control (Rovithakis *et al.*, 1999), turbo generator control (Venayagamoorthy & Harley, 1999), gold stud bump-manufacturing control and optimisation (Liau & Chen, 2005), steel making control (Staib & Staib, 1992), and glass-manufacturing process control (Müller *et al.*, 2004). Next, we describe in more detail the last two applications, as they best resemble our glass-manufacturing control application.

Staib & Staib (1992) proposed the Intelligent Arc Furnace controller to help improving the steel-melting process at North Star Steel, Iowa. The scrap steel is melted in an electric arc furnace. Three graphite electrodes (one foot in diameter and 20 feet long) are inserted into the furnace to heat the scrap steel. The electrodes are supplied with energy from a three-phase power line of massive electric-power capacity. The depth of the electrodes in the furnace is controlled by three independent servos. The positioning of the electrodes into the furnace is crucial for effective operation. Staib & Staib (1992) proposed three neural networks, which used the furnace state and regulator values interchangeably as the network inputs and outputs. The first network emulates the furnace regulator. It uses previous and current regulator and furnace state values – reg(t-1), reg(t), state(t-1), state(t) – to determine the correct output for the regulator – reg(t+1). The second network emulates the furnace state and uses reg(t-1), reg(t), state(t-1), state(t) plus reg(t+1) to determine the next furnace state – state(t+1). The error (the difference between the actual furnace next state and the estimated one) is used to adjust the network weights. Finally, the third network (Figure 4-2) is a combination of the previous two: the regulator values obtained with the first network – reg(t+1) – are used as inputs for the second network, which determines the next furnace state – state(t+1). Here the error is used to adjust the weights of the first network. As a consequence of introducing the neural network controller the consumption of electric power decreased by five to eight per cent, the wear and tear on the furnace and the electrodes was reduced by about 20 per cent and the daily throughput of steel increased by 10 per cent (Widrow *et al.* 1994, p. 101).

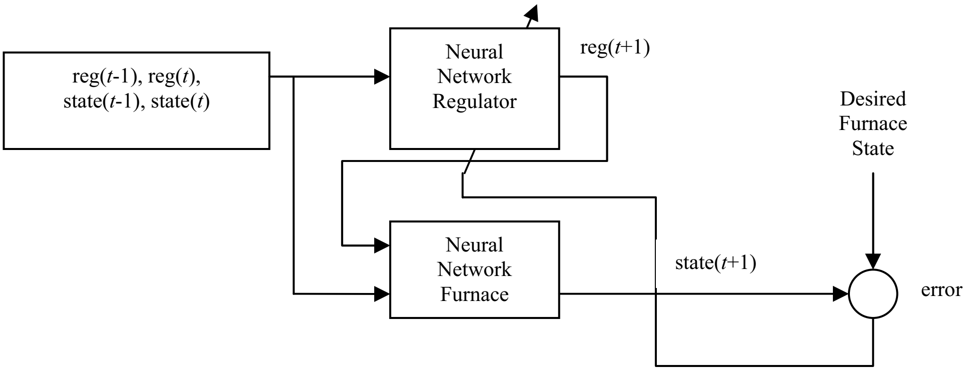


Figure 4-2 Furnace/Regulator network structure. Reg(*t*) – regulator outputs for time *t*, state(*t*) – furnace state conditions for time *t*.
(Source: Widrow *et al.* 1994, p. 100)

Müller *et al.* (2004) proposed a complex system called Expert System ES III for the control of the entire glass-melting technology, starting from batch charging to conditioning. The system incorporates techniques from different fields such as Model-Based Predictive Control, Neural Networks, Fuzzy Logic and Advanced Optimisation. The system was constructed at Glass Service, Inc. a Czech-based consulting company in the field of glass melting, conditioning and forming. Glass Service, Inc. through its expert system ES III provides modelling and consulting services for the glass-manufacturing process such as assessment of container strength, support of mould design, speed increase, light weighting, mould cooling design, troubleshooting glass failure, analysis of cyclic thermal behaviour, understanding the process, yield improvement, eliminating sampling, reduced time to market (Glass Services, Inc., 2005). The data gathering needed for modelling the melting process is improved by visual observation and image analysis of the processes in the molten glass. Some possible benefits of implementing the above expert system are: furnace operation consistency 24 hours a day, furnace sensor fault detection, furnace stability leading to fewer defects, more stable crown and bottom glass temperatures, less risk of corrosion because of better controlled and maintained temperatures, increased glass homogeneity, fuel savings by continuous optimised heat input distribution, and improved yield by less glass defects coming from melting and forming (Müller *et al.*, 2004). Compared with manual control, ES III gave energy savings of up to three per cent and improved the yield up to eight per cent, which means the payback time could be much less than half a year (Müller *et al.*, 2004). The ES III helped in more than 300 furnace optimisations.

As we mentioned in the introductory chapter the contribution of our research is more on choosing and improving the CI methods used to address business problems (countries' economic performance benchmarking, companies' financial performance benchmarking, and prediction of process variables) rather than improving the business models that underlie each business application.

Chapter 5 CI Methods for Quantitative Data Mining

In this Chapter, we present the statistical[13] and the CI methods employed in performing DM clustering, classification and regression tasks. Through these DM tasks we address the three business applications described in the previous chapter: countries'/companies' economic/financial performance benchmarking and prediction of process control variables.

5.1 Different Approaches to DM Clustering Task

In this section we introduce different techniques used to carry out the DM clustering task. We compare different clustering techniques such as SOM, C-Means, FCM, and introduce a new clustering algorithm called Weighting FCM. We also tackle some existing technical problems with the aforementioned techniques: validate the performance of the SOM algorithm, and optimise SOM topology.

5.1.1 The SOM

The SOM (Self-Organising Map) algorithm is a well-known unsupervised-learning algorithm developed by Kohonen in the early 80's and is based on a two-layer neural network (Kohonen, 1997). The algorithm creates a two-dimensional map from *n*-dimensional input data (Figure 5-1). After training, each neuron (unit) of the map contains input vectors with similar characteristics, e.g. companies with similar financial performance.

The SOM has a rectangular or hexagonal topology. Each neuron *i* contains a weight vector $m_i = [m_{i1}, \ldots, m_{in}]$ where *n* is the number of inputs. Before training, the weight vectors are initialised. The default initialisation is a random initialisation. However, the weight vectors can be initialised linearly[14] as well. At each step of the training algorithm an input observation *x* is randomly selected and distances from *x* to all weight vectors are calculated. The best matching unit (neuron) – denoted by m_c – is the one whose weight vector is closest to *x* (Equation 5-1).

$$\|x - m_c\| = \min_i \{\|x - m_i\|\} \tag{5-1}$$

[13] We follow the classification of the DM algorithm proposed by Romeu (2001) presented in section 3.3.1, which groups DM algorithms in three categories: mathematically based, statistically-based and "mixed" algorithms. In our experiments we do not use mathematically based algorithms. The statistically based algorithms used are: C-Means and Multinomial Logistic Regression. All the other algorithms proposed in this dissertation (CI methods) are so-called "mixed" algorithms.

[14] In a linear initialisation the weights vectors (reference vectors) are initialised in an orderly fashion along a two-dimensional subspace spanned by the two principal eigenvectors of the input data vectors (Kohonen *et al.* 1995, p.19).

Usually, the squared Euclidean distance is used to calculate the distances. After the closest neuron for input vector *x* has been identified, the weight vector m_i is updated with the formula:

$$m_i(t+1) = \begin{cases} m_i(t) + \alpha(t)[x - m_i(t)], & i \in N_c \\ m_i(t), & i \notin N_c \end{cases} \tag{5-2}$$

where *x* is the randomly selected observation at moment *t*, N_c is a set of neurons in the vicinity of the winner *c* and $0 < \alpha(t) < 1$ is the learning rate, which is a monotonically decreasing function of *t* (Kohonen 1997, pp. 86-88; Alhoniemi *et al.* 1999, pp. 4-5). The learning $\alpha(t)$ can be a linear function (the default):

$$\alpha(t) = \alpha(0)(1 - t / rlen) \tag{5-3}$$

or a inverse-type function:

$$\alpha(t) = \alpha(0)C/(C + t) \tag{5-4}$$

where $\alpha(0)$ is the initial learning rate, $C = 100 / rlen$ and *rlen* is number of steps in training (Kohonen *et al.* 1995, p. 15). In any case, $\alpha(t)$ decreases to 0. The set N_c can be defined using the radius length *N* (the radius of the circle, which represents the vicinity of the winner *c*). *N* can be defined as a function of time:

$$N(t) = 1 + [N(0) - 1] \cdot \left(1 - {}^{t}\!/_{rlen}\right) \tag{5-5}$$

where $N(0)$ is the initial radius length. *N*(*t*) decreases linearly to 1.

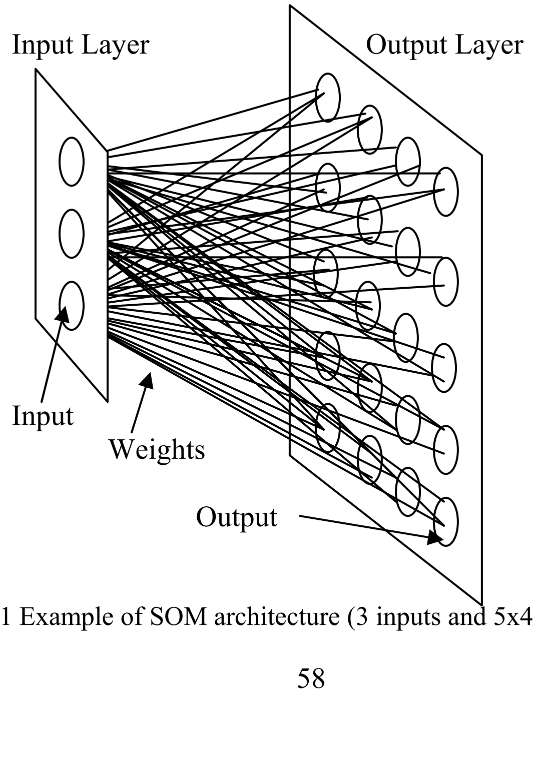


Figure 5-1 Example of SOM architecture (3 inputs and 5x4 rectangular map)

The algorithm stops when a predefined number of training steps has been reached (the default stopping criterion) or if the improvement in the overall *average quantisation error* is very small. The overall average quantisation error is given by formula:

$$qe = \frac{1}{n}\sum_{i=1}^{n}\left\| x_i - m_c^{(i)} \right\| \tag{5-6}$$

where $m_c^{(i)}$ is the closest weight vector for the input vector x_i.

In general, the learning process can proceed batchwise (Kohonen, 1998) or sequentially: in batch learning a number of observations are fed into the model simultaneously, while in sequential learning the observations are presented sequentially, one by one, to the algorithm. In our experiments, we used the sequential learning algorithm.

Kohonen (1997) suggests two training phases: a rough phase where *N*(0) is about half of the map diameter size and $\alpha(0) = 0.5$ and a fine-tuning phase where *N*(0) is very small (e.g. *N*(0) = 2) and $\alpha(0)$ is, also, very small (e.g. $\alpha(0) = 0.05$). Other rules of thumb (Kohonen, 1997, pp. 86-88) in choosing SOM parameters include: horizontal map dimensionality (*X*) should be approximately 1.3 times vertical dimensionality (*Y*). The training length (*rlen*) for the second training phase (rough phase) is 500 times the number of neurons. The training length for the first phase is ten per cent of the training length for the second phase. However, Kohonen (1997, p. 88) admits that the selection of training parameters is not crucial when training small maps (few hundreds of nodes).

The result of SOM training is a matrix that contains the codebook vectors (weight vectors). The SOM can be visualised using the *U-matrix* method proposed by Ultsch (1993). The unified distance matrix or U-matrix method computes all distances between neighbouring weights vectors. The borders between neurons are then constructed on the basis of these distances: dark borders correspond to large distances between two neurons involved, while light borders correspond to small distances. In this way we can visually group the neurons ("raw" clusters) that are close to each other to form supra-clusters or "real" clusters (Figure 5-2 (a)). The "raw" clusters can automatically be grouped using another clustering technique such as Ward's method. The main idea of the Ward method is to group at each step two "raw" clusters (or weight vectors) that will lead to the least "information loss". Ward defines the information loss in terms of squared Euclidean distance. For more details concerning the Ward method see Ward (1963).

In addition to the U-matrix map, a *component plane* or *feature plane* can be constructed for each individual input variable. In the feature planes light/"warm" colours for the neurons correspond to high values, while dark/"cold" colours correspond to low values (Figure 5-2 (b)). The component plane representation can be considered a "sliced" version of the SOM, where each plane shows the distribution of one weight vector component (Alhoniemi *et al*. 1999, p. 6). Also, *operating points* and *trajectories* (Alhoniemi *et al*. 1999, p. 6 and Figure 5-2 (a) red line) are used to find how different points (observations) move around on the map (e.g. how the companies evolved over time with respect to their financial performances).

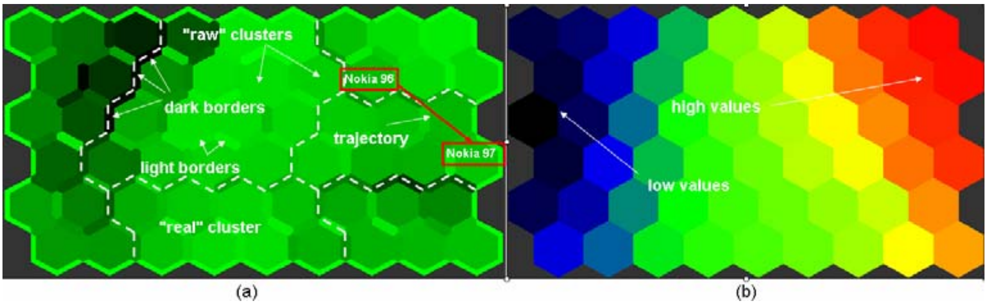


Figure 5-2 (a) The U-matrix representation with Nenet v1.1a software program and (b) Return on Total assets (ROTA) component plane

There are a number of commercial and public domain SOM software tools. *SOM_PAK* is the first SOM software program developed at the Helsinki University of Technology by a team under the supervision of the SOM algorithm inventor, Professor Teuvo Kohonen. The program is public-domain software for non-commercial use and it is written in C programming language (Kohonen *et al*., 1995). It runs under UNIX and MS-DOS platforms. *SOM Toolbox* is another public domain software package that can be incorporated as a normal toolbox in the Matlab environment. The SOM Toolbox was developed by the HUT team and is available at http://www.cis.hut.fi/projects/somtoolbox/. A complete explanation of the SOM Toolbox can be found in Vesanto *et al*. (2000). Two commercial SOM software tools are: *Nenet v1.1a,* available as a limited demonstration at http://koti.mbnet.fi/~phodju/nenet/Nenet/Download.html and *Viscovery® SOMine*, a product of Eudaptics GmbH in Austria (www.eudaptics.com).

SOM for DM clustering task – Research Issues

Many researchers have focused on applying SOM to performing the DM clustering task in general, and economic/financial performance benchmarking in particular. Oja *et al*. (2003) cites 5384 scientific papers – published between 1981 and 2002 – that use the SOM algorithms, have benefited from them, or contain analyses of them. However, relatively few of them (73) have applied SOM to business-related issues (Oja *et al*., 2003). Eklund *et al*. (2003) finds out that "the self-organizing map is quite capable of producing easy to interpret results, which can be used to provide a better overall picture of a company's financial performance". Back *et al.* (1998) found that "by using self organizing maps, we overcome the problems associated with finding the appropriate underlying distribution and the functional form of the underlying data in the structuring task that is often encountered, for example, when using cluster analysis". Deboek (1998) outlines 12 financial, 4 economic and 5 marketing applications of the SOM. Basically, in all these applications a SOM was used to cluster the data.

There are two main differences between our study and those referred to in terms of using the SOM as a performance-benchmarking tool. One difference is that here we do not solely apply the SOM as a tool for economic/financial performance benchmarking but, at the same time, we compare it with some other clustering techniques such as C-Means clustering (Costea *et al.*, 2001) or fuzzy C-Means clustering (*Publication* 4). The other difference comes from the limitation that techniques such as the SOM have: in essence they constitute descriptive data analysis techniques and aim at summarising the data by transforming it into a two-dimensional space and preserving the dissimilarities between observations. Employing the SOM does not imply that the use of other well-known techniques is renounced; rather, it is very productive to complement it with other tools (Serrano Cinca, 1998a). Consequently, in this study, we go one step further and use the output of the SOM (or the output of the other clustering techniques) as the input for the classification models. Moreover, another distinction with the other studies is that, in our research, we answer some technical questions related to the practical implementation of the SOM as a performance-benchmarking tool. As was mentioned in Section 1.3, we have addressed two technical SOM problems: the validation of map topology and quantisation error. At the same time, a method to *automate* the process of constructing the clusters is presented (see Section 6.2.2).

5.1.2 C-Means

C-Means is a partitive statistical clustering technique first proposed in MacQueen (1967). The goal of the C-Means algorithm is to minimise the sum of the variances within clusters. The objective function is defined as:

$$J = \sum_{j=1}^{C} \sum_{i \in S_j} \left\| x_i - c_j \right\| \tag{5-7}$$

where $\|\cdot\|$ is some measure of similarity (e.g. Euclidean distance), C is the number of clusters, x_i is observation i and c_j is centre of cluster j.

The basic algorithm has the following steps:

1. Define the number of clusters C.
2. Select C observations as the centres for the C predefined clusters.
3. Assign each observation from the dataset to the closest cluster (e.g. in terms of Euclidean distance).
4. Recalculate the centres of the clusters (centroids – c_j) either after each observation assignment or after all assignments.
5. Repeat steps 3 and 4 until the number of maximum iterations has been reached or the centroids no longer change.

There are different variations of the C-Means algorithm, depending on how the clusters' centres are initialised or what distance measure is used (Han & Kamber, 2000; Theodoridis & Koutroumbas, 2003, pp. 529-532). The strengths of he C-Means algorithms come from their relative efficiency and ability to yield local optimum results. However, the C-Means algorithm has certain disadvantages: the need to specify the number of the clusters in advance, it is sensitive to initialisation of the centres, applicable only to ratio/interval scale data, unable to handle noisy data and outliers, unsuitable for discovering clusters with non-convex shapes (Han & Kamber, 2000).

5.1.3 Fuzzy Clustering

In contrast to the above methods, fuzzy logic (Zadeh, 1965) deals with the uncertainty that comes from imprecise information and vagueness. This uncertainty is caused by the linguistic imprecision intrinsic in many problems and is called *lexical uncertainty*. While conventional stochastic uncertainty deals with the uncertainty of whether a certain event will occur or not, *lexical* uncertainty deals with the uncertainty of the definition of the event itself (Lindström 1998, p. 3). Conventional Boolean logic is substituted by *degrees* or *grades of truth*, which allow for intermediate values between true and false. For example, in classical set theory an element x of a set X is a member of subset A ($A \subset X$) if the value for x of the characteristic function χ_A is 1. In other words, $\chi_A(x) = 1$, with $\chi_A : X \rightarrow \{0, 1\}$. Similarly, a fuzzy subset A of the set X is defined by the membership function $\mu_A : X \rightarrow [0, 1]$. The membership function μ_A assigns a degree of truth (membership degree) in the closed interval [0, 1] for every element in the fuzzy (sub)set A. Degrees of 0 and 1 represent non-membership and full membership respectively to that set, while values in between represent intermediate degrees of set membership (Carlsson & Fuller 2002, p. 1). In this framework, fuzzy clustering methods assign different membership degrees to the elements in the dataset, indicating to what degree the observation belongs to every cluster.

A traditional method in fuzzy clustering is the *fuzzy C-Means clustering method* (FCM) (Bezdek, 1981). A further developed version is the *Weighting FCM* described later in this Section.

Fuzzy C-Means (FCM)

In fuzzy clustering every observation is assigned a vector representing its membership degree in every cluster, which indicates that observations may contain, with different strengths, the characteristics of more than one cluster. The objective of FCM (Bezdek, 1981) is to minimise the following objective function:

$$J(U, v) = \sum_{k=1}^{n} \sum_{i=1}^{c} (u_{ik})^m (d_{ik})^2 \tag{5-8}$$

where c is the number of clusters, n is the number of observations,

$$U = \{u_{ik}, i = 1,...,n; k = 1,..., c \mid u_{ik} \in [0, 1]\} \tag{5-9}$$

is a fuzzy C-partition of the dataset *X* and $u_{ik} \in [0, 1]$ is the membership degree of observation x_k in cluster *i* with $\sum_{k=1}^{c} u_{ik} = 1$,

$$d_{ik} = \|x_k - v_i\| = \sqrt{\sum_{j=1}^{p} (x_{kj} - v_{ij})^2} \tag{5-10}$$

is the Euclidean distance between observation x_k and the cluster centre v_i with *p* – number of input variables.

The weighting exponent $m \in [1, \infty)$ controls the extent of membership-sharing between the clusters. When $m \to 1$, the Fuzzy C-Means converges to the C-Means and when $m \to \infty$, then $u_{ik} \to 1/c$ and the centres tend towards the centroid of the dataset (the centres tend to be equal). The FCM algorithm has the following steps:

1. Fix c, $2 \le c \le n$, and *m*, $1 \le m < \infty$. Initialise $U^{(0)}$. Then, for *s*th iteration, $s = 0, 1, 2, \ldots$
2. Calculate the *c* cluster centres $v_i^{(s)}$ based on $U^{(s)}$
3. Calculate $U^{(s+1)}$ based on $v_i^{(s)}$
4. Compare $U^{(s+1)}$ to $U^{(s)}$: if $\left\| U^{(s+1)} - U^{(s)} \right\| \le \varepsilon$ stop; otherwise return to Step 2.

Basically the FCM algorithm is similar to the C-Means algorithm: the number of clusters is predefined, the algorithm stops when there is no change in the membership degree matrix. It also inherits C-Means weaknesses such us sensitivity to noise and outliers. Several robust methods to deal with noise and outliers in the context of FCM are presented in Leski (2003).

Weighting FCM

The FCM algorithm gives the membership degree of every observation for every cluster. The usual criterion for assigning the data to their clusters is to choose the cluster where the observation has the highest membership value. While that may work for a great number of elements, some other data vectors may be misallocated. This is the case when the two highest membership degrees are very close to each other, for example, one observation has a membership degree of 0.45 for the first cluster and 0.46 for the third. It is difficult to say in which cluster should we include it and it is possible that, after analysing the vector components, we realise it does not correspond to the average characteristics of the cluster chosen. We call this data vector an "uncertain" observation. Therefore, it would be useful to introduce into the algorithm some kind of information about the characteristics of every cluster so that the "uncertain" observations can be better allocated depending on which of these features they fulfil more.

Generation of linguistic variables
We can express the characteristics of every cluster by using linguistic variables for each input variable. For example, in the case of companies' financial performance analysis, the quantitative ratio Operating Margin is represented using a linguistic variable - qualitative Operating Margin. A linguistic variable can be regarded as a variable whose values are fuzzy numbers. For example, the above Operating Margin linguistic variable takes as values different linguistic terms: very low (VL), low (L), average (A), high (H), and very high (VH) operating margin. Each linguistic term is represented by a fuzzy number whose membership function is defined according to the value domain of the Operating Margin. It is common to represent linguistic variables by linguistic terms positioned symmetrically (Lindström, 1998). However, this symmetrical representation of the linguistic variable holds only in the case where the empirical distribution of the quantitative variable is symmetric. Since the data are not always symmetric, we can apply normal FCM (which is free of any distributional assumptions) to each variable to make a fuzzy partition of the variable corresponding to the five linguistic terms (see Figure 5-3).

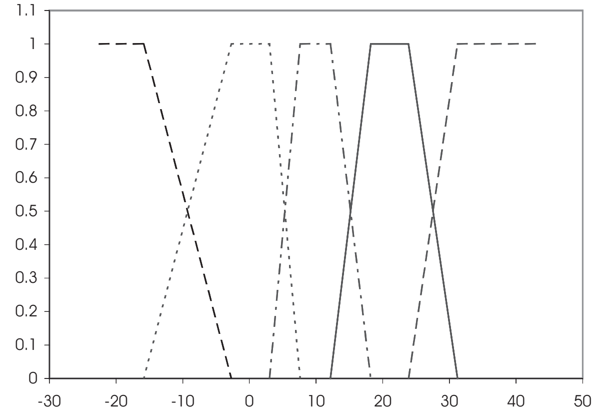


Figure 5-3 Example of fuzzy partition of a linguistic variable
(Source: *Publication* 4)

We can proceed likewise and obtain fuzzy partitions for all individual input variables. Then, using the linguistic variables, we obtain a *linguistic matrix* that contains a linguistic term for each value of the input variables.

Calculation of the weights for FCM
Once we have the linguistic matrix, we can obtain an importance coefficient (weight) for every variable in every cluster and introduce it into the clustering algorithm. The objective is to better allocate "uncertain" observations, taking into consideration the linguistic characterisations of the variables in every cluster. The

weights are calculated on the basis of the “certain” observations. Therefore, we first have to differentiate between “certain” and “uncertain” observations.

For that we apply normal FCM and obtain the matrix U with the membership degrees of every observation in every cluster. “Uncertain” observations are those for which the difference between the two maximum membership degrees is less than twice the equal membership level (1/c) for every cluster, which seems a reasonable assumption to clearly define linguistic structures in the clusters. Next, we remove the “uncertain” observations from the clusters. Then, we characterise each cluster using the observations from the *linguistic matrix* that correspond to the remaining “certain” observations. In every cluster and for every variable we can obtain how many times every linguistic term appears and also the percentage with respect to the total number of observations in the cluster. Clearly, an input variable will be important for the cluster if it has a high percentage of occurrences concentrated in few linguistic terms. On the other hand, if one input variable has a number of occurrences evenly distributed among the linguistic terms, it will not be a good definer of the cluster. As a measure of how evenly or unevenly the percentages of the occurrences are distributed we use the standardised variation coefficient (SVC_{ij}):

$SVC_{ij} = {VC_{ij}} \Big/ {\sum_{j=1}^{p} VC_{ij}}$, where variation coefficient $VC_{ij} = {\sigma(perc_{ij})} \Big/ {\overline{perc_{ij}}}$ and

p is the number of variables, $perc_{ij}$ the vector of percentages of the input variable j in cluster i, $\sigma(perc_{ij})$ and $\overline{perc_{ij}}$ are the standard deviation and the mean respectively of the vector $perc_{ij}$. A high variation coefficient for the percentages indicates that the ratio clearly defines the cluster.

Introduction of the weights in the calculation of the new distances

The previous weights are introduced in the Euclidean distance term of the FCM algorithm in the following form:

$$d_{ik} = \left[\sum_{j=1}^{p} (x_{kj} - v_{ij})^2 SVC_{ij} \right]^{1/2} \qquad (5\text{-}11)$$

The “uncertain” observations are gradually allocated to the clusters based on the new distances. The Weighting FCM algorithm has the following steps:

1. Fix c and m. Initialise $U = U^{(1)}$. Apply normal FCM to the whole dataset and determine the “certain” (I) and “uncertain” (I') sets of observations. Determine SVC_{ij} based on “certain” observations. We will denote the final U obtained at this step by $U^{(l)}$. Next (steps 2-5 iteratively), allocate the

"uncertain" observations to the "certain" clusters. Iteration *s* (*s* = 1, 2, …) in allocating the "uncertain" elements consists of following steps:

2. In iteration *s*, calculate the centres of the clusters with the membership degrees $u_{ik}^{(s)}$ corresponding to the "certain" observations of the current iteration and $u_{ik}^{(s-1)}$ corresponding to the previous iteration. When *s* = 1, $u_{ik}^{1} = U^{(l)}$ and $u_{ik}^{0} = 0, \forall\ i = 1,...,c; \forall\ k = 1,...,n$.
3. Calculate $u_{ik}^{(s+1)}$ of the "uncertain" observations with the centres obtained in Step 2, and the previous degrees $u_{ik}^{(s)}$, $k \in I'$ where I' is the set of "uncertain" data.
4. Identify the new "certain" observations from I' (based on $u_{ik}^{(s+1)}$ from the previous step) and allocate them to the corresponding clusters. Update I with the new "certain" observations from I'. The remaining "uncertain" observations will become I' in the next iteration.
5. If at least one "uncertain" observation was allocated, go to Step 2. If not, stop.

5.1.4 Related Research in Comparing Different CI Methods for Performing the DM Clustering Task

In our study we compare different methods (e.g. statistical methods such as C-Means and CI methods such as SOM and FCM) to determine which one performs the DM clustering task better. In this section we present other studies that compare different clustering techniques.

One study that compared the SOM with different hierarchical clustering techniques is Mangiameli *et al.* (1996). The authors compared the SOM with the following hierarchical clustering methods: *single linkage*, *complete linkage*, *average linkage*, *centroid method*, *Ward's method*, *two-stage density* and *Kth nearest neighbour*. The authors tested these methods on 252 datasets. The datasets were characterised by different factors such as level of dispersion (low, medium, high), outlier concentration (10%, 60%), irrelevant information (1 irrelevant variable, 2 irrelevant variables), and cluster density (10%, 60%). The SOM was found to be the best method in terms of cluster accuracy: for 191 datasets (75.8%) It was the best performer regardless of the values for the different factors. Also, the SOM results were found not to be sensitive on the initial learning rate $\alpha(0)$ (see Equation 5-3).

Vesanto & Alhoniemi (2000) compared basic SOM clustering with different partitive (C-Means) and agglomerative (single linkage, average linkage, complete linkage) clustering methods. At the same time, the authors introduced a two-stage SOM clustering (similar with our SOM clustering approach) which consists of, firstly, applying the basic SOM to obtain a large number of prototypes ("raw" clusters) and, secondly, clustering these prototypes to obtain a reduced number of data clusters ("real" clusters). The partitive and agglomerative clustering methods were used to perform the second phase of the two-stage clustering. In other words, these methods were used to group the prototypes obtained by SOM into "real" clusters. The comparisons were made using two artificial and one real-world datasets. The comparisons between the basic SOM and other clustering methods were based on the computational cost. SOM clearly outperformed the agglomerative methods (e.g. average linkage needed 13 hours to directly cluster the dataset III, whereas SOM needed only 9.5 minutes). The clustering accuracy (in terms of conditional entropies) was used to compare the direct partitioning of data with the two-stage partitioning. The results show that partitioning based on the prototypes of the SOM is much more evenly distributed (approximately an equal number of observations is obtained in each cluster). At the same time, the two-stage clustering results were comparable with the results obtained directly from the data.

Another study that compares different methods for clustering is Kiang (2001). The author implemented two versions of the extended SOM networks similar to that proposed in Vesanto & Alhoniemi (2000): one uses minimum variance and the other a minimum distance criterion for clustering the SOM prototypes (Kiang 2001, p. 176). The author performed two experiments. In the first experiment the author used a classic problem in group technology, which is a production flow analysis problem and involves 43 parts and 16 machines. He compared the extended versions of SOM with five other clustering methods proposed in five previous studies that analysed the same dataset. The comparison criterion used was *grouping efficacy* proposed in Kumar & Chandrasekharan (1990). The extended SOM performed better than the previous reported results. In the second experiment the author compared SOM with C-Means, Ward's method and a non-parametric method supported by SAS MODECLUS procedure in terms of clustering accuracy (rate of correctness %). Again, the extended versions of the SOM outperformed the other clustering methods for both iris and wine recognition datasets used in this experiment.

No report was found in literature that compared FCM and SOM for the DM clustering task.

5.2 Different Approaches to the DM Classification Task

In this section we introduce different approaches to solving the DM classification task. As we mentioned before (Section 1.3), we build hybrid classifiers by following a two-phase methodology: firstly, one of the clustering techniques described in Section 5.1 is applied and we obtain several clusters that contain similar data-vectors, and then, we construct a classification model in order to place new data within the clusters obtained in the first phase. We are interested in finding the *most adequate hybrid classification model* for a given business problem.

Pendharkar (2002) differentiates the approaches to the DM classification task on the basis of the algorithms used. The author differentiates between statistical, induction and neural approaches. In this Section we present the approaches that we used in our experiments for performing the DM classification task: one statistical approach (multinomial logistic regression – MLR), one induction approach (decision-tree induction in the form of C4.5/C5.0/See5.0 algorithm – DT), and three neural approaches (the first one is a standard approach where we use the ANN obtained when we determine the ANN architecture as our ANN classification model, the second and third approaches use a retraining procedure – RT – and the genetic algorithm – GA – respectively to further improve the accuracy of the ANN obtained when determining the ANN architecture).

Once we have the class variable for our datasets, we can use the above statistical, induction and neural predictive modelling techniques to build the hybrid classifiers. In general, when a predictive modelling technique is applied to classification, there are certain steps that have to be followed:

1. *Check the requirements regarding the dataset: sample size, missing data.* The requirements for the sample size depend on the technique used to build the classifier. When we encountered missing data, we replace them with averages or using simple regression models. We had very few missing data in our datasets.
2. *Compute the classification model* using an available software program (e.g. SPSS, See5) or, a self-constructed program (e.g. a Matlab script). Here the parameters of the models are carefully scrutinised using different empirical validations.
3. *Assess the model fit (accuracy)*. Here we validate our model on the basis of the training data. There are two criteria to test the utility of the model through the means of classification accuracy: proportional by-chance criterion and maximum by-chance criterion. Both criteria require the classification accuracy to be 25% better than the *proportional by-chance accuracy rate* and *maximum by-chance accuracy rate* respectively (Hair *et al*. 1987, pp. 89-90). The proportional by-chance accuracy rate is calculated by summing the squared proportion of each group in the sample: the square proportion of correctly classified cases in class 1 + … + the square proportion of correctly classified cases in class *n*. The maximum by-chance accuracy rate is the proportion of cases in the largest group.
4. *Interpret the results*. Here the relative importance of the attributes in building the classifier is discussed. Also here, we discuss the correspondence between the class predictions and what happened in reality.
5. *Validate the model* (based on the test data). To validate our classification models we split the data into two datasets with approximately the same number of observations. We perform two new classifications: one when the first dataset is used for training and the second dataset for testing, and the other with the two datasets interchanged. The validation is performed in terms of training and

testing accuracy rates: a model is validated if there is a small difference in the training and testing accuracy rates of both models and, also, there are small differences between the training and testing accuracy rates in the case of each individual model. The above method is also known as the *hold-out* method. Another method used to validate the model is the *C-fold cross-validation* method: the dataset is divided into *C* subsets and each time, one of the *C* subsets is used as the test set and the other *C*-1 subsets form a training set. Then we compute the average accuracy rate based on the *C* trials. Again, the models are validated based on the differences between training and cross-validation accuracy rates. Besides the objective evaluation criteria such as accuracy rates (for classification) and quantisation errors (for clustering), we also check the fidelity with real-world phenomena. In other words, we check which model best answered our experiment-related questions.

For all our classification approaches we follow the above methodological steps.

5.2.1 Statistical Approaches

Statistical techniques were deployed first to tackle the classification task: univariate statistics for prediction of failures introduced by Beaver (1966), linear discriminant analysis (LDA) introduced by Fisher (1936), who first applied it to Anderson's iris dataset (Anderson, 1935), multivariate discriminant analysis (MDA) – Altman (1968), Edmister (1972), Jones (1987), and probit and logit (logistic) models – Ohlson (1980), Hamer (1983), Zavgren (1985), Rudolfer *et al.* (1999).

Compared to other statistical approaches such as Discriminant Analysis (DA) for the classification problem, the Logistic Regression (LR) approach has a very important feature: it has a nice probabilistic interpretation because of the sigmoid function employed (Atiya 2001, p. 930). DA makes two important assumptions not made by LR: the input variables need to be normally distributed, and within-group variances-covariances have to be equal. The most used DA method is the Linear DA (LDA), in which the so-called discriminant scores depend linearly on the dependent variables. LR does not assume a linear relationship between input and output variables. Because of the non-linear nature of its regression function, LR can handle non-linear effects even when exponential and polynomial terms are not explicitly added as additional independent variables (Garson, 2005). At the same time, LR does not assume normally distributed error terms. Moreover, Eisenbeis (1977) suggests that economic and financial variables, in particular, violate both DA assumptions since many measurements are of nominal or ordinal nature at best.

Binary logistic regression – BLR – refers to the case where the dependent variable has two classes, whereas *multinomial* logistic regression – MLR – refers to the multi-class case. MLR classifies cases by calculating the likelihood of each observation belonging to each class. The regression functions have a logistic form and return the likelihood (the odds) that one observation (x) belongs to a class (C):

$$odds(x \in C) = \frac{1}{1+e^{-logit}} = \frac{1}{1+e^{-(w_0+w_1v_1+...+w_pv_p)}} \tag{5-12}$$

where v_1, ...v_p are the input variables, and w_0,...,w_p are the regression coefficients (weights).

MLR computes the odds by performing the following steps:

1. For each of the first c-1 classes, the weights w_0,...,w_p are estimated from the training data by applying maximum likelihood estimation (MLE). The mathematics underlying the maximum likelihood estimation procedures is beyond the scope of the present dissertation. However, MLE seeks to maximise the log likelihood (LL), which reflects how likely it is that the observed values of the dependent may be predicted from the observed values of the independents. That is, logistic regression calculates changes in the log odds of the dependent, not changes in the dependent itself as OLS (ordinary least squares) regression does (Garson, 2005). Further reading for interested reader is Hosmer & Lemeshow (2000).
2. Calculation of the logistic transformation of *odds* (taking the logit of the *odds ratio*) based on the estimates calculated in step 1 for each of c-1 classes.

   $$logit(odds\ ratio) = ln\left(\frac{odds}{1 - odds}\right) = w_0 + w_1v_1 + ... + w_pv_p \tag{5-13}$$

   where *logit*(*odds ratio*) is also known as the difference in *log likelihood* (LL), and *ln* is the natural logarithm (the logarithm in base e = 2.71..). Then, *odds* are extracted from the *logit* with the aid of equation (5-12). The rationale behind the usage of a logistic function is that $logit(odds) \in (-\infty,+\infty)$, whereas $odds \in [0, 1]$.
3. The last likelihood (that the observation x belongs to the last class, c) is calculated by subtracting the sum of other likelihoods from 1.

In contrast to LR, probit models are based on the cumulative distribution function (CDF) of the unit-normal distribution. In other words, $odds(x \in C)$ becomes:

$$odds(x \in C) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{z} e^{-\frac{1}{2}z^2} dz \text{, where } z = w_0 + w_1v_1 + ... + w_pv_p \text{.}$$

However, LR has two main advantages over the probit models: *simplicity* and *interpretability* (Fox, 1997). Simplicity relates to the form of the sigmoid function (employed by LR), which is much simpler than the cumulative distribution function (employed by probit). Interpretability relates to the fact that the sigmoid function has an inverse that is interpretable, while the inverse of the cumulative distribution function is not interpretable (Fox, 1997). Throughout our experiments we used LR as this has the least number of assumptions and is the most successful statistical approach for classification.

Next we follow the methodological steps presented at the beginning of this section. In their guidelines in applying logistic regression Hosmer & Lemeshow (2000) suggest that the minimum number of cases per independent variable (***size***) is 10-15. The ***missing data*** are handled differently depending on the software package used to perform the logistic regression. However, we replace the missing data as we explained in the beginning of Section 5.2. Throughout our experiments we ***computed*** the multinomial logistic regression using SPSS (SPSS for Windows, 2002).

To test whether our ***model fits*** the dependency between the class variable and input vector, MLR uses the "Likelihood Ratio" test. This test shows whether the difference in likelihood values between the model that contains the independent variables ("full" model) and the model without the independent variables ("null" model) is statistically significant. Each model produces a maximised likelihood, i.e. L_1 for the "full" model and L_0 for the "null" model. MLR calculates the statistic $-2\ln L_0 - (-2\ln L_1) = 2(\ln L_1 - \ln L_0)$ where $-2\ln L$ is called *deviance* under the model. The above statistic, which tests the difference in likelihood, has an approximately chi-square distribution. The significance of the chi-square statistic (e.g. sig. < 0.05) provides evidence for the relationship between the dependent variable and the combination of independent variables. However, this significance test does not tell us whether certain independent variables are more important than others. The importance of each independent variable is tested with the same test ("Likelihood Ratio"), but, in this case, MLR calculates the difference in likelihood between the "full" model and the model that does not contain the variable in question. We will present this test later in this section. In order to measure the strengths of the relationship between the dependent variable and the combination of independent variables, MLR computes the so-called "pseudo" R-Square measures. These correlation measures attempt to measure the strength of the association between dependent and independent variables rather than explain the variance in the dependent (Garson, 2005). The correlation measures are based on the deviance quantities ($-2\ln L$) One of the correlation measures is Cox & Snell's R^2, which has the following form:

$$R^2_{Cox\ and\ Snell} = 1 - \left[\frac{-2\ln L_0}{-2\ln L_1} \right]^{2/n} \qquad (5\text{-}14)$$

where n is the sample size. Cox & Snell's R^2 is hard to interpret because its maximum is less than 1. Another correlation measure is Nagelkerke's R^2 (Nagelkerke, 1991) which is based on the Cox & Snell's coefficient. Nagelrkerke's R^2 has the following formula:

$$R^2_{Nagelkerke} = \frac{1 - \left[\frac{-2\ln L_0}{-2\ln L_1} \right]^{2/n}}{1 - \left(-2\ln L_0 \right)^{2/n}} \tag{5-15}$$

Nagelkerke R^2 divides the Cox & Snell's coefficient by its maximum in order to achieve a measure that ranges from 0 to 1 (Garson, 2005). A value close to 1 for these coefficients shows a strong relationship. Another measure used to assess the classification performance of MLR is the classification accuracy rate, which is the proportion of correctly classified cases. The accuracy rate is validated according to the proportional by-chance and maximum by-chance criteria presented at the beginning of Section 5.2.

To ***interpret the results*** we check how each independent variable contributes to explaining the likelihood variation in the output (using the "Likelihood Ratio" test). We also check whether the coefficients' estimates are significant (using the Wald test).

The "Likelihood Ratio" test evaluates the overall importance of each independent variable for explaining the likelihood variations in the dependent variable. Again, the difference in likelihood values (–2ln*L* values) between the model that contains that specific independent variable and the model that does not include it is assumed to follow a chi-square distribution. The significance of the chi-square statistic calculated for each independent variable (e.g. sig. < 0.05) gives the evidence that the independent variable in question contributes significantly to explaining differences in classification.

MLR calculates the estimates ($\hat{w}_i, i = 0,..., p$) for the coefficients of all regression equations using the MLE procedure. If there are *c* classes, the table builds *c*-1 regression equations. One class, usually the last one, is the reference class. In other words, for each independent variable, there are *c*-1 comparisons. MLR calculates the standard errors for the regression coefficients, which show the potential numerical problems that we might encounter. Standard errors larger than 2 can be caused by multicolinearity between variables (not directly handled by SPSS or other statistical packages) or dependent variable values that have no cases, etc (Hosmer & Lemeshow, 2000). Next, MLR calculates the *Wald* statistic, which tests whether the coefficients are statistically significant in each of the *c*-1 regression equations. In other words it tests the null hypothesis that the logit coefficient is zero. The Wald statistic is the ratio of the unstandardised logit coefficient to its standard error (Garson, 2005). Next, MLR shows the degree of freedom for the Wald statistic. If "sig." values are less than the 1 – confidence level (e.g. 5%) then the coefficient differs significantly from zero. The signs of the regression coefficients show the direction of the relationship between each independent variable and the class variable. Positive coefficients show that the variable in question influences positively the likelihood of attaching the specific class to the observations. Values greater than 1 for $e^{\hat{w}_i}$ show that the increase in the variable in question would lead to a greater likelihood of attaching the specific class to the observations. For example, if $e^{\hat{w}_1} = 3$ for class c_1 and variable v_1, we can interpret this value as follows: for each unit increase in v_1 the likelihood that the observations will be classified in class c_1 increases by approximately three times. Finally, MLR shows the lower and upper limits of the confidence intervals for the $e^{\hat{w}_i}$ values at the 95-per cent confidence level.

To ***validate*** the MLR classification models we use the general procedure presented at the beginning of Section 5.2.

5.2.2 Tree Induction Approaches

Another technique used to partition an input data space in predefined classes is the **decision-tree induction**. The decision-tree learners construct the trees using the so-called *divide-and-conquer* method. This method originates in the work of Hunt (Hunt, 1962; Hunt *et al.*, 1966). The method, also known as *Top-Down Induction of Decision Trees* (TDIDT) was developed and refined over more than twenty years by Ross Quinlan (Quinlan 1979, 1983, 1986, 1993a, 1993b, 1997; Kohavi & Quinlan, 2002). At each step, the TDIDT method selects an attribute that "best" discriminates the dataset (according to a certain criterion), and does this recursively for each subset until all the cases from all subsets belong to a certain class. In the first implementation of the TDIDT method, called the ID3 algorithm (Quinlan, 1979), the test for choosing the splitting attribute is based on the *gain* criterion. Let us assume we have a test X with n outcomes that partitions the set T of training cases into subsets T_1, T_2, …, T_n. At each step the algorithm calculates the *information gain* of splitting the data based on test X with the formula:

$$gain(X) = info(T) - info_X(T) \tag{5-16}$$

where $gain(X)$ is the information that is gained by partitioning T in accordance with the test X, $info(T)$ is the expected amount of information needed to specify the class of a case in T and $info_X(T)$ is the expected amount of information needed to specify the class of a case in T given that the case reaches the node with the test X (Quinlan, 1993b). $info(T)$ is given by:

$$info(T) = -\sum_{j=1}^{k} \frac{freq(C_j,T)}{|T|} \times \log_2 \frac{freq(C_j,T)}{|T|} \tag{5-17}$$

where k is the number of classes, $freq(C_j,T)$ is the frequency of cases that belong to the class C_j in the set T, and $|T|$ is the number of cases in T. The information is measured in units called *bits*. $info(T)$ given by (5-17) is known as the *entropy* function and it was selected because it is the only function that satisfies the requirements for the information values: (1) when the numbers of cases in all classes, except one, are zero, the information is zero; (2) when the number of cases in each class is the same for all classes, the information reaches a maximum; (3) must obey the multistage property (Witten & Franck 2000, p. 93).

Let us consider a set *T* which has 3 classes and the number of cases in *T* is nine with the following distribution: two cases in the first, three cases in the second, and four cases in the third class. In this particular case, the information needed to specify the class of a case in *T* is *info*([2/9, 3/9, 4/9]). This information can be calculated in two ways: in a one-stage and a two-stage way. In the one-stage way, we decide, in one step, in which one of the three classes the case belongs and we assign that class to it. In the two-stage way, first, we decide whether the case (the observation) is in the first class or one of the other two classes: *info*([2/9, 7/9]) and then, if it is not the first class, we decide which one of the other two it is: 7/9**info*([3/7, 4/7]). Consequently, *info*([2/9, 3/9, 4/9]) = *info*([2/9, 7/9]) + 7/9**info*([3/7, 4/7]). If the case is in the first class, the second decision will not be needed. The multistage property of function *info*(*T*) refers precisely to the above characteristic: *info*(*T*) must be able to calculate the information needed in the stages described above. One function *F* obeys the multistage property if:

$$F(p,q,r) = F(p,q+r) + (q+r)F\left(\frac{q}{q+r}, \frac{r}{q+r}\right), \; p+q+r=1.$$

One such function is the *entropy* function used in Equation 5-17:

$$entropy(p_1, p_2, ..., p_n) = -p_1 \log_2 p_1 - p_2 \log_2 p_2 - ... - p_n \log_2 p_n \quad (5\text{-}18)$$

where $p_1 + p_2 + ... + p_n = 1$ (Witten & Franck 2000, p. 94). The entropy function obeys the multistage property as demonstrates, in the above example, the equality: *info*([2/9, 3/9, 4/9]) = *entropy*([2/9, 3/9, 4/9]) = *entropy*([2/9, 7/9]) + 7/9**entropy*([3/7, 4/7]).

The expected amount of information needed to specify the class of a case in *T* given that the case reaches the node with the test *X* with *n* outcomes – $info_X(T)$ – is:

$$info_X(T) = \sum_{i=1}^{n} \frac{|T_i|}{|T|} \times info(T_i) \quad (5\text{-}19)$$

where $|T_i|$ is the number of cases in subset T_i, and *info*(T_i) is calculated using (5-17). To give an example of how the *gain information* is calculated we consider the playing forecast data in Table 5-1.

Let us calculate the information gain of splitting the data based on the attribute *outlook*. We calculate the parameters needed from Table 5-1: *k*=2, *freq*(*Play*, *T*) = 9, *freq*(*Don't Play*, *T*) = 5, $|T| = 14$, *n* = 3, $|T_1| = 5$, $|T_2| = 4$, $|T_3| = 5$, which leads to:

$$info(T) = -9/14 \times \log_2(9/14) - 5/14 \times \log_2(5/14) = 0.940 \; bits \, .$$

$$\begin{aligned} info_{outlook}(T) = {} & 5/14(-2/5 \times \log_2(2/5) - 3/5 \times \log_2(3/5)) \\ & + 4/14(-4/4 \times \log_2(4/4) - 0/4 \times \log_2(0/4)) \\ & + 5/14(-3/5 \times \log_2(3/5) - 2/5 \times \log_2(2/5)) = 0.694 \; bits \end{aligned}$$

The information gain is therefore:

$$gain(outlook) = 0.940 - 0.694 = 0.246 \; bits \, .$$

Table 5-1 A small playing forecast dataset

Outlook	Temp (°F)	Humidity (%)	Windy?	Class
sunny	75	70	true	Play
sunny	80	90	true	Don't Play
sunny	85	85	false	Don't Play
sunny	72	95	false	Don't Play
sunny	69	70	false	Play
overcast	72	90	true	Play
overcast	83	78	false	Play
overcast	64	65	true	Play
overcast	81	75	false	Play
rain	71	80	true	Don't Play
rain	65	70	true	Don't Play
rain	75	80	false	Play
rain	68	80	false	Play
rain	70	96	false	Play

(Source: Quinlan 1993b, p. 18)

The information gain for the other attributes can be calculated in a similar fashion. For example, the attribute *windy?* yields *gain*(*windy?*) = 0.940 – 0.890 = 0.05 *bits*. The attribute with the highest gain is chosen for splitting the set *T*. Then, the above procedure is applied recursively for all subsets T_i.

The gain information criterion has a major drawback: it tends to prefer attributes with large numbers of possible values (Witten & Frank 2000, p. 95). In other words, the gain information criterion favours the tests with many outcomes (large *n*). To overcome this problem Quilan proposed the *gain ratio* criterion (Quinlan, 1988). The gain ratio is derived by taking into account the number and size of the daughter nodes into which an attribute splits the dataset, disregarding any information about the class (Witten & Frank 2000, p. 95).

$$gain\ ratio(X) = gain(X) / split\ info(X) \tag{5-20}$$

where *split info*(*X*) represents the potential information of dividing *T* into *n* subsets. The *split info* is higher for highly branching attributes leading to smaller information gains. The formula is similar to (5-17) but, here, instead of the class to which the case belongs, the outcome of the test is considered:

$$split\ info(X) = -\sum_{i=1}^{n} \frac{|T_i|}{|T|} \times \log_2\left(\frac{|T_i|}{|T|}\right) \tag{5-21}$$

The split information generated by dividing the data into 3 subsets based on the attribute *outlook* is therefore:

$$split\ info(outlook) = -5/14 \times \log_2(5/14) - 4/14 \times \log_2(4/14)$$
$$-5/14 \times \log_2(5/14) = 1.577$$

Therefore, the gain ratio for the attribute *outlook* is:

$$gain\ ratio(outlook) = 0.246/1.577 = 0.156\ bits$$

In the case of the attribute *windy?* we obtained:

$$gain\ ratio(windy?) = 0.050/0.984 = 0.051\ bits$$

As the figures show, the gain ratio criterion reduced the advantage of the attribute *outlook* compared with the attribute *windy?* (the gain for *outlook* decreased from 0.246 to 0.156, while there was an increase in the information gain for *windy?* from 0.05 to 0.051). This is in line with the assumption that the gain ratio criterion favours attributes with a smaller number of outcomes (*outlook* has three, while *windy?* has two outcomes).

Quinlan’s C4.5 and its commercial implementations See5 (for the Windows platform) and C5.0 (for the UNIX platform) use the gain ratio criterion to split the data. Beside the splitting criterion, in decision-tree induction we have to deal with some other issues such as: the treatment of missing values and a pruning mechanism to avoid over-fitting.

C4.5 deals with the missing data as follows: if a splitting test *X* has an unknown outcome (the attribute on which the test is based has a missing value), then the unknown outcome is treated as another outcome. *gain*(*X*) is calculated based only on the known outcomes, whereas *split info*(*X*) is calculated based on all outcomes, including the unknown outcome. A case with an unknown test outcome is divided into fragments whose weights are proportional to the relative frequencies of the known outcomes (Quinlan 1993b, p. 33).

In general, there are two types of pruning methods: pre-pruning methods which involve decisions on when to stop developing sub-trees in the tree-building process, and post-pruning methods which build the complete tree and prune it afterwards. There are two types of post-pruning: *sub-tree replacement* and *sub-tree raising* (Witten & Franck 2000, p. 162). C4.5 implements both of them. The criterion to prune the (sub)tree is based on minimising the error rate associated with the (sub)tree. If the error rate of a (sub)tree is greater than the error rate of the leaf which would replace the (sub)tree, then the (sub)tree is pruned. C4.5 has two ways to predict the error rates: one is based on the training sample and the other on the hold-out sample. The first prediction method is a "heuristic based on some statistical reasoning, but the statistical underpinning is rather weak and ad hoc" (Witten & Franck 2000, p. 164). The error probabilities are assumed to have a binomial distribution and the upper limit of these probabilities can be found from the confidence limits for the binomial distribution. Quinlan (1993b, p. 41) provides an illustrative example of how these probabilities are estimated for each leaf node based on a particular confidence *c* (the default value for confidence *c* used in C4.5 is 0.25). The lower the confidence, the more drastic is the pruning.

C4.5 offers also a method to validate the error rates based on the *C-fold cross-validation* approach. The available cases are divided into *C* equal-sized blocks and, for each block, a tree is constructed from the cases in all the other blocks and tested on the cases in the "hold-out" block (Quinlan 1993b, p. 40).

Another parameter in C4.5 is *minimum-case* parameter (*m*), whose effect is to eliminate tests for which the outcomes have less than *minimum-case* instances. Also, tests are not incorporated in the decision tree unless they have at least two outcomes (Witten & Franck 2000, p. 169).

CART (classification and regression trees) is another decision-tree learner introduced by Breiman *et al*. (1984). The CART technique is also based on the divide-and-conquer methodology described above. However, C4.5 and CART differ with respect to the splitting criterion, the treatment of missing values and the pruning mechanism. CART uses the *Gini diversity index* (Kohavi & Quinlan, 2002) as the base for the splitting criterion and provides a class distribution for each case instead of a single class. The *Gini diversity index* has the following form:

$$info_{Gini}(T) = 1 - \sum_{j=1}^{k} \left[\frac{freq(C_j, T)}{|T|} \right]^2 \qquad (5\text{-}22)$$

where all the notations have the same meaning as for the C4.5 algorithm described above (see Equation 5-17). $info_{Gini}(T)$ is the expected amount of information needed to specify the class of a case in *T*. The information gain (the splitting criterion) is calculated as in the case of C4.5 (see Equation 5-16) by the following formula:

$$gain_{Gini}(X) = info_{Gini}(T) - \sum_{i=1}^{n} \frac{|T_i|}{|T|} \cdot info_{Gini}(T_i) \qquad (5\text{-}23)$$

Unlike C4.5, CART handles missing data by taking into account only the outcomes that have known values. CART finds for each chosen test another test called the *surrogate split,* which is used if some outcome of the original test is not known. The surrogate split test can have its own surrogate split. For details regarding CART (e.g. the pruning mechanism used) we refer the reader to Breiman *et al.* (1984) or Kohavi & Quinlan (2002).

Another decision-tree learner is the recursive partitioning algorithm (RPA) introduced in Frydman *et al.* (1985). In general, the RPAs have as the splitting criterion so-called purity measures, i.e. the absolute purity (in percentages) of the classes that we can achieve in each node. Usually, this purity measure is calculated for each attribute-value pair, and the best pair is chosen as a test to split the tree. Throughout our experiments we used the C4.5 algorithm, as this is the most researched and successful induction technique.

Again we follow the methodological steps in building a classification model presented at the beginning of Section 5.2. As we showed above, C4.5 handles quite nicely the ***missing data***. However, in our experiments we replace the missing data with averages or using simple regression models. To ***construct the decision-tree*** classifier we ran See5 software. The parameters were the confidence factor c = 0.25 and the minimum number of cases m = 5. We have also used fuzzy thresholds. A threshold for a variable in a dataset is a constant value between a pair of adjacent values. If one variable has n distinct values, there are n-1 thresholds. A fuzzy threshold is a fuzzy number that corresponds to the constant value represented by the threshold. For example, if one threshold $\theta = 9$ for a variable v, then its corresponding fuzzy number can have a triangular shape as in the following figure:

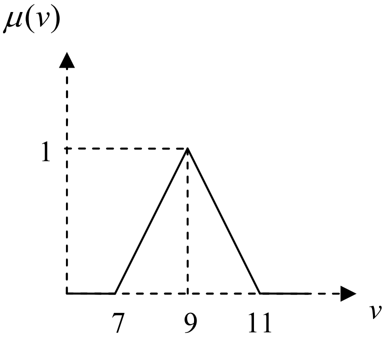


Figure 5-4 A fuzzy number v represented by a triangular membership function μ

A fuzzy number can have other shapes as well: trapezoidal, Gaussian, etc. The shape of the fuzzy number is given through the membership function (μ) described in Section 5.1.3. Further reading regarding fuzzy numbers is, for example, Berthold (1999). We used fuzzy thresholds in our C4.5 implementation because if we do not use them, then small changes in the values of the attribute could lead to changing the branch taken to test the next attribute. Using fuzzy thresholds both branches of the tree are explored and the results combined to give a predicted class. We ***assess the accuracy of the model*** by calculating the training accuracy rate and two more accuracy rates: the proportional by-chance accuracy rate and the maximum by-chance accuracy rate. Usually, decision trees are easy to ***interpret***. The higher one attribute is in the tree structure, the more important that attribute is. We ***validate*** the model by splitting the dataset into two parts exactly as we did in the case of MLR. We obtained two new decision trees. We also perform a *C-fold cross-validation* on training data. The model is validated based on the differences in accuracy rates.

5.2.3 Neural Approaches

Another approach to tackling the DM classification task is represented by **artificial neural networks** (ANNs). Among ANNs, the Multilayer Perceptron neural network (MLP), trained by the back-propagation algorithm is currently the most widely used neural network (Hagan *et al*., 1996). MLPs are feed-forward neural networks that can be trained with the standard back-propagation algorithm or other gradient-descent algorithms. Feed-forward neural networks allow the signal to be transmitted only forward. In other words, there are no loops in the network structure. An MLP is an interesting alternative to other classifiers: even when the type of distribution of the features is unknown an MLP with the optimal number of hidden nodes approaches Bayesian classifiers and hence its error rate will be close to the minimum (Richard & Lippmann, 1991). In the literature (Haykin, 1995) there are some estimates of the utilisation of different forms of neural networks: MLPs (81.2%), recurrent networks (5.4%), Kohonen's SOMs (8.3%), others (5.1%). However, this reference is rather old and new types of ANNs are gaining more and more acceptance. Usually, the ANN[15] weights are learned by gradient-descent-like algorithms such as back-propagation and its variants.

Another way of learning the connection weights of an ANN is represented by evolutionary algorithms such as genetic algorithms. Yao (1999) explores the possible benefits of combining ANNs and evolutionary algorithms (EAs). EAs refer to a class of population-based stochastic search algorithms such as evolution strategies (ESs), evolutionary programming (EP) and genetic algorithms (GAs) that are based on the principles of natural evolution (Yao 1999, p. 1424). Yao & Liu (1997) proposed a new evolutionary system - EPNet - for evolving ANNs. The authors use evolutionary programming for evolving simultaneously ANN architecture and connection weights. EPNet was applied to a number of experiments (N-parity problem, the two-spiral problem, four medical diagnosis problems, the Australian credit card assessment problem, and the Mackey-Glass time-series prediction problem) that show that EPNet can discover ANNs that would be difficult to design by human beings.

[15] In this dissertation, the term *ANN* has two meanings depending on the context: firstly, it has the generally accepted meaning for an artificial neural network, and, secondly, it means an ANN in the form of a MLP.

Fogel *et al*. (1995, 1998) used ANNs trained with evolutionary algorithms to analyse interpreted radiographic features from film screen mammograms. The results show that even small ANNs (with two hidden nodes and a small number of important features) can achieve comparable results with much more complex ones. Chellapilla & Fogel (1999) combined ANNs and evolutionary algorithms to learn appropriate and, sometimes (e.g. checkers) near-expert strategies in zero and non-zero-sum games such as the iterated prisoner's dilemma, tic-tac-toe, and checkers.

The generic classification model based on neural approaches is depicted in Figure 5-5. Usually, when constructing classification models, the first step is to separate the data into training (*TR*) and test (*TS*) sets. If the class variable is missing, as in our case, a clustering method could be applied to build this variable (Section 5.1). The second step consists of selecting the proper ANN architecture. This step is concerned with determining the proper number of hidden layers, and the appropriate number of neurons in each hidden layer. Furthermore, here we decide how the class variable should be coded. In other words, how many neurons are necessary in the output layer to represent the class variable? The last step, ANN training, consists of specific tasks depending on the training mechanism used.

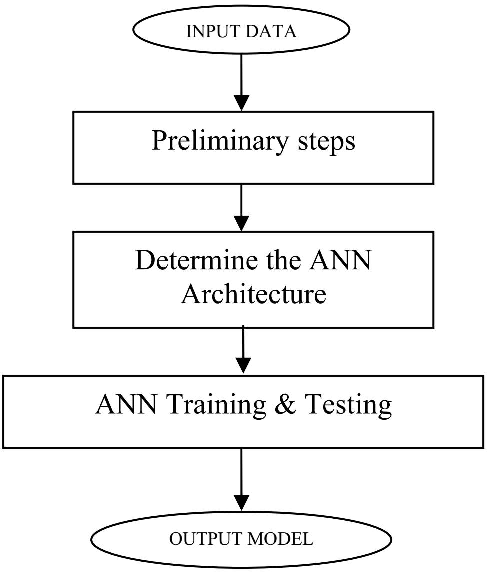


Figure 5-5 ANN generic classification model
(Source: *Publication* 5)

As we mentioned at the beginning of Section 5.2 we used three neural approaches to build ANN-based classification models. We describe these next.

First neural approach: the ANN obtained when determining the ANN architecture

Once we have determined the class variable and pre-processed the data, we can use the training data to determine the proper architecture for the ANN. Regarding the number of output neurons, we have two alternatives when applying ANNs to pattern classification. The first alternative, which is most commonly used, is to have as many output neurons as the number of classes. The second alternative is to have just one neuron in the output layer, which will take the different classes as values. The choice between the two alternatives is based on the number of cases per weights ratio restriction.

Choosing the number of hidden layers and the number of neurons in each hidden layer is not a straightforward task. The choices of these numbers depend on "input/output vector sizes, size of training and test subsets, and, more importantly, the problem of non-linearity" (Basheer & Hajmeer 2000, p. 22). It is well known that neural networks are very sensitive regarding the dimensionality of the dataset (Hagan *et al*., 1996; Basheer & Hajmeer, 2000; Demuth & Beale, 2001). Basheer & Hajmeer (2000) cite a number of papers that introduce different rules of thumb that link the number of hidden neurons (*NH*) with the number of input (*NI*) and output (*NO*) neurons or with the number of training samples (N_{TRN}). One rule of thumb, proposed in Lachtermacher & Fuller (1995) suggests that the number of hidden neurons *NH* for one output *ANN* is: $0.11N_{TRN} < NH(NI+1) < 0.30N_{TRN}$. Upadhyaya & Eryurek (1992) related the total number of weights N_w to the number of training samples: $N_w = N_{TRN}\log_2(N_{TRN})$. Masters (1994) proposed that the number of hidden neurons in the hidden layer should take values in the vicinity of the geometric mean of the number of inputs (*NI*) and of outputs (*NO*). We followed Basheer & Hajmeer's (2000, p. 23) advice that "the most popular approach to finding the optimal number of hidden nodes is by trial and error with one of the above rules". For example, in *Publication* 3 and Costea (2003) we chose the Lachtermarcher & Fuller (1995) rule and varied NH between 7 and 25, and 5 and 9 respectively, depending on the size of the training set. In *Publication* 5 we chose Masters' rule of thumb as a starting point to develop our ANN architectures. Concerning the number of hidden layers, we performed in each case a number of experiments for ANN architectures with one and two hidden layers to see what the appropriate number of hidden layers is. Depending on the dataset used, an ANN with one or two hidden layers performed better in terms of the mean square error of training. We did not take into consideration three hidden layer cases because of the restriction imposed by the number of cases per weights ratio.

We have used the sigmoid and linear activation functions for the hidden and output layers respectively, as this combination of activation functions provided the best results in our experiments. Regarding the training algorithms, they fall into two main categories: heuristic techniques (momentum, variable learning rate) and numerical optimisation techniques (conjugate gradient, Levenberg-Marquardt). Various comparative studies, on different problems, were initiated in order to establish the optimal algorithm (Demuth & Beale, 2001; Nastac & Koskivaara, 2003; Costea, 2003). In Costea (2003) we compared four training algorithms in terms of error rates and convergence speed. Our findings suggest that there is a negative correlation between error rates and the convergence speed. Therefore, in choosing the training algorithm one should seek a compromise between these two factors. As a general conclusion, it is difficult to know which training algorithm will provide the best result for a given problem. A smart choice depends on how many parameters of the ANN are involved, the dataset, the error goal, and whether the network is being used for pattern recognition (classification) or function approximation. Statistically speaking, it seems that numerical optimisation techniques present numerous advantages. Analysing the algorithms that fall into this class, we observed that the Scaled Conjugate Gradient (SCG) algorithm (Moller, 1993) performs well over a wide variety of problems. Even if SCG is not the fastest algorithm (as Levenberg-Marquardt in some situations), the great advantage is that this technique works very efficiently for networks with a large number of weights. The SCG is something of a compromise: it does not require large computational memory, and yet, it still has a good convergence and is very robust. Furthermore, we always apply the early stopping method (*validation stop*[16]) during the training process in order to avoid the over-fitting phenomenon. Moreover, it is well known that for early stopping, one must be careful not to use an algorithm that converges too rapidly (Hagan *et al.*, 1996; Demuth & Beale, 2001). The SCG is well suited for the validation stop method. There are, also, other possibilities to avoid the over-fitting phenomenon: weight decay, curvature-driven smoothing (see Bishop, 1995, pp.338-346), but their applicability is beyond the scope of the present dissertation.

In our experiments (performed using Matlab's Neural Networks toolbox) we have kept all parameters of the MLPs constant (the learning algorithm - SCG, the performance goal of the classifier, the maximum number of epochs), except the number of neurons in the hidden layers (*NH* when we had one hidden layer and NH_1, NH_2 when we had two hidden layers). In the appendix to *Publication* 5 we present a flowchart with the empirical procedure to determine the architecture for an ANN with two hidden layers.

Second neural approach: RT-based ANNs

Once we determine the ANN architecture (with the corresponding set of weights), we can stop and train the fixed architecture network normally with the training data and so obtain the ANN classification model (this is the first neural approach for classification used in *Publication* 3 and Costea (2003) and described in the previous section). However, we can go one step further and improve the accuracy of the trained network. The first training improvement mechanism is a retraining-based ANN (*Publication* 6; Nastac & Costea, 2004a), briefly described next:

- Start with a network with an initial set of weights from the previous step (Determining ANN architecture) as the reference network;
- Perform *L* runs to improve the ANN classification accuracy. After each experiment we save the best set of weights (the solution) in terms of classification accuracy. Each experiment consists of:

[16] The validation stop method implies separating the training set into two parts: an effective training set (TRe) and a validation set (VAL). The training process stops when the difference between the effective training error and the validation error exceeds a certain threshold.

- ❖ Reduction of the weights of the current best network with successive values of scaling factor γ (γ = 0.1, 0.2, …, 0.9) by multiplying the weights with γ.
  - ➢ Retrain the ANN with the new weights and obtain nine accuracy rates.
- ❖ Choose the best network from the above nine in terms of classification accuracy.
- ❖ Compare the accuracy rate of the current network with that obtained in the previous step and save the best one for the next run as the current best network.

Depending on the splitting of the training set (*TR*) into the effective training set (*TRe*) and validation set (*VAL*) we have three types of retraining mechanisms: one (RT1) where *TRe* and VAL are common to all of the *L* runs, another (RT2) where *TRe* and *VAL* are different for each run, but the same for all nine reduction weights trainings (second step of the experiment), and finally, RT3 where *TRe* and *VAL* are distinct for each training. We have four types of accuracy rates: effective training accuracy rate (ACR_{TRe}), validation accuracy rate (ACR_{VAL}), total training (effective training + validation) accuracy rate (ACR_{TR}) and test accuracy rate (ACR_{TS}). Correspondingly, we calculate four mean square errors: MSE_{TRe}, MSE_{VAL}, MSE_{TR}, and MSE_{TS}.

Third neural approach: GA-based ANNs (Evolutionary Approaches)

The second ANN training mechanism used to refine the solution is based on the principle of natural evolution. The algorithms used to perform this type of training are called evolutionary algorithms (EAs), one of which is the genetic algorithm (GA). GA is a heuristic optimisation search technique designed after the natural selection process, i.e. it follows the nature of sexual reproduction in which the genes of two parents combine to form those of their children (Anandarajan *et al.*, 2001). Unlike the traditional gradient-descent training mechanisms, GAs are provided with a population of solutions, and by initialisation, selection and reproduction mechanisms, achieve potentially good solutions. All solutions (chromosomes) compete with each other to enter the new population. They are *evaluated* according to the objective function. The best performing chromosomes are, then, *selected* based on this objective function to enter the new population. After selection, the chromosomes are randomly paired and recombined to produce new solutions with the *crossover* operator. Then, some chromosomes can *mutate* so that new information is introduced into the solution. The process is repeated iteratively until there is no increase in performance from one generation to the other.

In the case of GA-based ANN training, the GA's chromosome (solution) is the set of ANN weights after training represented as a vector. Next, we describe the GA steps performed to train the ANN.

Initialisation and fitness evaluation

The population size is a parameter of our models. It was set to *PS* = 20. Dorsey & Mayer (1995) suggest that this value is good enough for any grade of problem complexity. The first chromosome of the population is the set of weights obtained when determining the ANN architecture. The other 19 chromosomes are generated by training the ANN with the previously obtained architecture. Afterwards, the first generation of the algorithm may begin. The number of generations is related to the empirical formula suggested in Ankenbrandt (1991). Each chromosome is evaluated using the accuracy rate for the training set (ACR_{TR}).

Selection

Firstly, the elitism technique is applied in the sense that the best N_{elite} chromosomes in terms of ACR_{TR} are inserted into the new population. The rest of the chromosomes (20-N_{elite}) are selected based on the probability of selection (*roulette wheel* procedure) for each chromosome:

$$P_i = ACR_{TR-i} \Big/ \sum_{i=1}^{20} ACR_{TR-i} \qquad (5\text{-}24)$$

The higher the probability P_i for a chromosome is, the higher its chance of being drawn into the new population. We decided to employ *elitist* selection in our algorithms as a consequence of what was reported in the literature. Rudolph (1994), Miller & Thomson (1998), Shimodaira (1996), Fogel *et al.* (2004) are a few papers that prove the usefulness of using elitist selection.

Next, 80 per cent (probability of crossover: $P_c = 0.8$) of the chromosomes obtained previously are randomly selected for mating. The choice of crossover probability as well as the other GA parameters (mutation probability, population size) is more art than science. Tuson & Ross (1998) suggested that the proper choice of the crossover in the case of non-adaptive GAs depends upon the population model, the problem to be solved, its representation and the performance criterion being used. Rogero (2002) mentions that the probability of crossover is problem-dependent. The probability of crossover is not essential for the performance of our algorithm as long as it has a high value. This is because after reproduction we increase the population to include both the parents and their offspring.

Reproduction

The selected chromosomes are randomly paired and recombined to produce new solutions. There are two reproduction operators: *crossover* and *mutation*. With the first the mates are recombined and newborn solutions inherit information from both parents. With the second operator new parts of the search space are explored and, consequently, we expect that new information will be introduced into the population. In our studies we have applied four types of crossover: arithmetic, one-point, multi-point and uniform crossover.

The children chromosomes are *added* to the population. The size of the population becomes $PS' > PS$. Next, we apply the mutation operator for all the chromosomes in PS'. We used only uniform mutation.

The probability of mutation is set to $P_m = 0.01$, which means that approximately one per cent of the genes will mutate for each chromosome. A number $\alpha \in [0,1]$ is generated for each gene of each chromosome and if $\alpha \leq P_m$, the new gene is randomly generated within the variable domain. Otherwise, the gene remains the same. If at least one gene is changed, then the new chromosome is added to the population, obtaining $PS'' > PS' > PS$. As in the case of crossover probability, the proper setting of mutation probability depends on the population model, the problem to be solved, and the fitness function (Tuson & Ross, 1998). DeJong (1975) considers mutation probability to be inversely proportional to population size. Hesser & Männer (1990) include in the calculation of mutation probability both population size and chromosome length. Hoehn (1998) introduced mutation at both parental and offspring levels and implemented four GAs based on the mutation probabilities for the two levels. The author finds that introducing parental mutation is generally advantageous when compared to the standard GA with only offspring mutation. In our experiments we used both parental and offspring mutation by applying mutation to both parents and their offspring. This operation was possible since, after applying crossover operation, we *add* the new chromosomes (offspring) to the population and keep their parents.

The final step in constructing the new population is to reduce it in size to 20 chromosomes. We select from PS'' the best 20 chromosomes in terms of ACR_{TR} satisfying the condition that one chromosome can have no more than *max_lim* duplicates. We use the mutation operator to generate more chromosomes if the number of best chromosomes that satisfy the above condition is less than 20.

As a summary, excluding the crossover, the parameters of our GA models are as follows: number of generations (N_{gen}), population size (PS), number of elite chromosomes (N_{elite}), maximum number of splitting points (*max_split*) in the case of multi-point crossover, probability of crossover (P_c), probability of mutation (P_m), and maximum number of duplicates for the chromosomes (*max_lim*).

In *Publication* 3 and Costea (2003) we use the first neural approach and compare the ANN with multinomial logistic regression and decision-tree induction with respect to classification accuracy. Again, as in the case of MLR and DT, we follow the methodological steps presented at the beginning of Section 5.2. We replace the ***missing data*** as we did in the case of MLR and DT. To ***construct the ANN classifier*** we build a small Matlab script. We ***assess the model accuracy*** by calculating the training accuracy rate. To interpret the results we check how the ANN had classified the data. We ***validate*** the model by splitting the dataset into two parts exactly as we did in the case of MLR and DT. We obtained two new

ANN classifiers. We also perform a *C-fold cross-validation* on training data. Again, the ANN model is validated based on the differences in accuracy rates.

Neural Approaches for DM Classification Task – Research Issues

A crucial step in ANN training is the pre-processing of the input data. Pre-processing can be performed in two ways: one way is to apply the pre-processing technique for each individual input variable obtaining the same dimensionality of the input dataset, and the other is to apply a transformation to the whole input dataset, at once, possibly obtaining a different dataset dimensionality. The second way of pre-processing is applied when the dimensionality of the input vector is large, there are intercorrelations between variables and we want to reduce the dimensionality of the data and uncorrelate the input. For example, Vafaie & DeJong (1998) proposed a system for feature selection and/or construction that can improve the performance of the classification techniques. The authors applied their system to an eye-detection face recognition system, demonstrating substantially better classification rates than competing systems. Zupan *et al*. (1998) proposed function decomposition for feature transformation. Significantly better results were obtained in terms of accuracy rates when the input space was transformed using feature selection and/or construction. The former way of pre-processing (pre-processing of each individual variable separately) deals with two comparability issues regarding the input variables. Firstly, when we do not have any information about the importance of input variables to explain variations in the outputs, the most natural assumption is that each variable has to have the same importance in the training process. For that we could scale all variables so that they always fall within a specified range. Secondly, the dispersion of the variables should be the same for all variables, so that the impact of variable dispersion on ANN training is the same for all variables. Few research papers have studied different individual variable data pre-processing methods to help improve the ANN training. Koskivaara (2000) investigated the impact of four pre-processing techniques on the forecasting capability of ANNs when auditing financial accounts. The best results were achieved when the data were scaled either linearly or linearly on yearly bases. In *Publication* 5 we use three individual variable pre-processing approaches: “no pre-processing”, which does not take into consideration any of the comparability concerns, “division with the maximum absolute values”, which handles the first comparability issue and “normalisation”, which addresses both comparability issues. In *Publication* 5 we test whether the choice of the pre-processing approach for individual variables has any impact on the predictive performance of the ANN.

Not many research papers have studied the implications of data distributions on the predictive performance of ANN. Bhattacharyya & Pendharkar (1998) studied the impact of input distribution kurtosis and variance heterogeneity on the classification performance of different machine-learning and statistical techniques for classification. The authors found out that input data kurtosis play an important role in an ANN’s predictive performance. Pendharkar & Rodger (2004) studied the implications of data distributions determined through kurtosis and variance-covariance homogeneity (dispersion) on the predictive performance of GA-based and gradient-descent-based ANNs for classification. In Pendharkar & Rodger (2004) the authors tested three different types of crossover operator (one-point, arithmetic, and uniform crossover). No significant difference was found between the different crossover operators.

One of the goals of *Publication* 5 is to find out whether the combination of pre-processing approach and input data distribution has an impact on the ANN's classification performance. At the same time, we are interested in whether the data distribution has any influence on the choice of training technique when ANNs are applied to financial classification problems. In other words, does the data distribution - training mechanism combination have any impact on the ANN's classification performance? Consequently, data with different distributions have to be generated. We used the characteristics of the real data to derive four fictive datasets with uniform, normal, logistic and Laplace-distributed data.

In *Publication* 5 we discuss the effect of the three factors (data distribution, pre-processing method and training mechanism) and their combinations on the prediction performance of ANN-based classification models. Alander (1995) reviews 1760 references (from 1987 until 2003) on combining GAs and artificial neural networks. We found no report in the literature that studied the combined impact of these factors on ANN classification performance. *Publication* 5 fills this gap in the literature.

In *Publication* 5 we compared our research questions with what was previously reported in the literature (e.g. Bhattacharyya & Pendharkar, 1998; Pendharkhar, 2002; Pendharkar & Rodger, 2004). However, there are some important differences in the assumptions in our study compared with the others:

- The main difference is that in our studies RT and GA-based ANNs are used to *refine* the classification accuracy of an already obtained ANN-based solution for the classification problem. Both the GA and the RT-based ANNs start from a solution provided when determining the ANN architecture and they try to *refine* it. All other studies compared GA and gradient-descent methods starting from random solutions. We expect that the GA-based ANN will outperform the RT-based ANN in refining what the ANN already learned because of the GA's better searching capabilities. We present our results regarding this comparison in section 6.2.2 and *Publication* 5.
- The second main difference is the type of classification problem itself. Here we are interested in separating the input space into more than two parts (e.g. seven financial performance classes) providing more insights into the data.
- We are interested in whether the combination of pre-processing approach, distribution of the data, and training technique has any impact on the classifiers' predictive performances.
- Non-parametric statistical tests are used to validate the hypotheses. Only t-tests or ANOVA were used in the other studies, but no evidence of satisfaction of the

underlying assumptions was provided. We performed a 3-way ANOVA to strengthen the results of the non-parametric tests.

- Also four different crossover operators are used in order to find whether this operator has an influence on the GA's predictive performance. We introduce one crossover operator – multi-point crossover – in addition to the three crossover operators presented in Pendharkar & Rodger (2004).

5.2.4 Related Research in Comparing Different CI Methods for Performing the DM Classification Task

Atiya (2001, pp. 929-931) reviews a number of papers that compared different techniques for predicting bankruptcy that correspond to the DM classification task. Many studies compared Discriminant Analysis (DA) and Logistic Regression (LR) techniques in terms of prediction accuracy, but most of them used as experiments the two-class case. Few studies addressed the multi-class classification problem. In most of the related papers (e.g. Tam & Kiang 1992, Alici 1995) LR outperformed DA. To be in line with current research and, also, to avoid the assumptions of DA (see Section 5.2.1) we chose LR as the statistical classification technique in our experiments.

Altman *et al*. (1994) compared DA with artificial neural networks (ANNs) as to their accuracy in predicting bankruptcy one year ahead using data about 1000 Italian companies. DA performed slightly better than the artificial intelligence method.

Marais *et al*. (1984) used Probit (see Section 5.2.1) and RPA (see Section 5.2.2) techniques to classify commercial bank loans. The conclusion is that RPA is not significantly better, especially when the data do not include nominal variables.

Bramer (2000) proposes *Inducer*, a common platform with a graphical user interface implemented in Java, which can be used to analyse comparatively different rule induction algorithms (TDIDT and N-Prism) using a number of available datasets. The platform offers user-friendly settings and result summaries, and its modular development permits the addition of other algorithms.

Many research studies compared decision-tree induction techniques with other statistical or non-statistical techniques, some of which are: Braun & Chandler (1987), who compared LDA and ID3 in predicting stock market behaviour, ID3 achieving better results; Garrison & Michaelsen (1989) compared ID3, LDA and probit on tax-decision problems, finding that ID3 achieved the best results; Kattan *et al*. (1993) compared ID3 with LDA and ANN classifiers and, again, ID3 outperformed the other classifiers. Elomaa (1994) compares C4.5 with 1R, which is a simple one-level decision tree proposed in Holte (1993) and claims that the differences in accuracy rates are still significantly in favour of C4.5. Marmelstein & Lamont (1998) compared a genetic algorithm-based approach called GRaCCE (Genetic Rule and Classifier Construction Environment) with CART technique and found out that GRaCCE achieved competitive accuracy rates and more compact rule sets.

Serrano Cinca (1996) proposes the SOM for predicting corporate failure and compares SOM with linear discriminant analysis (LDA) and a multilayer perceptron (MLP) trained with a back-propagation algorithm (BP) by "superimposing" the results of LDA and MLP on the solvency map. The author forecasts the combination of SOM with the other techniques for predicting bankruptcy and claims that this will be the future path in related research. Schütze *et al.* (1995) used a weak learning algorithm (relevance feedback) in comparison with complex algorithms (LDA, LR and ANN) for the document routing problem and found that the complex algorithms outperformed the weak one. Jeng *et al.* (1997) constructed a fuzzy inductive learning algorithm (FILM) and compared it with ID3 and LDA for bankruptcy predictions and biomedical applications. ID3 achieved better results than LDA and FILM slightly outperformed ID3. Back *et al.* (1996b, 1997) used LDA, LR and ANN to predict companies' bankruptcy and found that ANN was the best performer in terms of accuracy.

Amin *et al*. (2003) compared DA, ANN and the TDIDT algorithm implemented by the Inducer Rule Induction Workbench (Bramer, 2000) in the problem of rhino-horn fingerprinting identification. Their results showed that the two intelligent methods (neural nets and automatic rule induction) "improve upon DA as a means of analysing the rhino horn data and are less prone to problems of model overfitting" (Amin *et al*., 2003, p. 336).

As we presented in Section 5.2.3 we used two different training mechanisms for learning the connection weights of ANN classification models: gradient-descent-like training algorithms and genetic algorithms. Schaffer *et al.* (1992) listed 250 references that combined ANNs and genetic algorithms. Sexton & Sikander (2001) found GA to be an appropriate alternative to gradient descent-like algorithms for training neural networks and, at the same time, the GA could identify relevant input variables in the dataset.

Other authors (e.g. Schaffer, 1994) found that GA-based ANNs are not as competitive as their gradient descent-like counterparts. Sexton *et al.* (1998) argued that this difference has nothing to do with the GA's ability to perform the task, but rather with the way it is implemented. One reason for GA being outperformed by gradient-descent techniques may be that the candidate solutions (the ANN weights) were encoded as binary strings, which is both unnecessary and unbeneficial (Davis, 1991; Michalewicz, 1992). In our experiments we use non-binary (real) values for encoding the weights.

Another issue in ANN and GA design for classification is represented by the different pre-processing methods used. Zupan *et al.* (1998) used function decomposition for the transformation of the input space. The authors compared their system (HINT) with Quinlan's C4.5 decision-tree algorithm in terms of prediction accuracy, and found that the system based on function decomposition yielded significantly better results.

There is no clear methodology in the related literature on rigorously comparing the methods presented above when they are applied to perform the DM classification task, because of their different research backgrounds and parameter settings. We addressed this problem by presenting at the beginning of Section 5.2 the necessary methodological steps in comparing the statistical, induction-tree and neural approaches for performing the DM classification task.

5.3 ANNs for the DM Regression Task

In this Section we present a general ANN model used to perform the DM regression task.

5.3.1 The ANN Forecasting Model

In Section 5.2.3 ANNs were used to perform the DM classification task, which is a particular case of regression analysis where the outcome is a discrete value (class). In this Section ANNs are used to perform the DM regression task, in which case the outcomes are real values. ANNs are modelling tools that have the ability to adapt to and learn complex topologies of inter-correlated multidimensional data (Basheer & Hajmeer, 2000; Hagan *et al*., 1996; Hornik *et al*., 1989). Constructing reliable time-series models for data forecasting is challenging because of non-stationarities and non-linear effects (Berardi & Zhang, 2003; Lacroix *et al*., 1997; Moller, 1993; Weigend *et al*., 1990; Zhang *et al*., 1998).

The steps in designing the ANN for regression task are mainly the same as for classification. We perform some preliminary steps regarding pre-processing of data, then we empirically determine the ANN architecture, and finally we use the retraining procedure to refine the previous obtained set of weights. However, there are differences: in the ANN classification models the outputs (classes) are independent. In other words, one output does not depend on the values taken by other outputs. Also, the time factor is not present in ANN classification models. Depending on the time-series data specification we can have dependencies between two outputs or between one output and one input at different points in time. For example, one output O_1 at moment t, $O_1(t)$, might be influenced by another output at moment t-1, $O_2(t-1)$, or by one input at different points in time - $I_1(t)$, $I_1(t-2)$, etc. Therefore, one difference in ANN design for classification and ANN design for prediction (forecasting) comes from the way the outputs depend on the inputs (model structure).

In Figure 5-6 we present as example the ANN model structure used in the experiment to estimate relevant process control variables for glass manufacturing. The process consists of 29 inputs and 5 outputs. The temperatures (the outputs) are simulated based on the other *delayed* inputs and outputs. At moment **t**, one output

output_i(t), is affected by the inputs from different past time steps (t-v_i$_1$, …, t-v_i$_n$), and the outputs from other past time steps (t-v_o$_1$, …, t-v_o$_m$). We denote by *delay vectors*, ***Vect_In*** and ***Vect_Out***, two vectors that includes the delays taken into account for the model: $Vect\_In = [\mathrm{v\_i}_1, \mathrm{v\_i}_2, ..., \mathrm{v\_i}_n]$ and $Vect\_Out = [v\_o_1, v\_o_2, ..., v\_o_m]$, where n and m may vary, depending on the problem. For example, for the glass-manufacturing process model we used delay vectors with n = 7, 8, 9 elements and m = 3, 4, 5 elements.

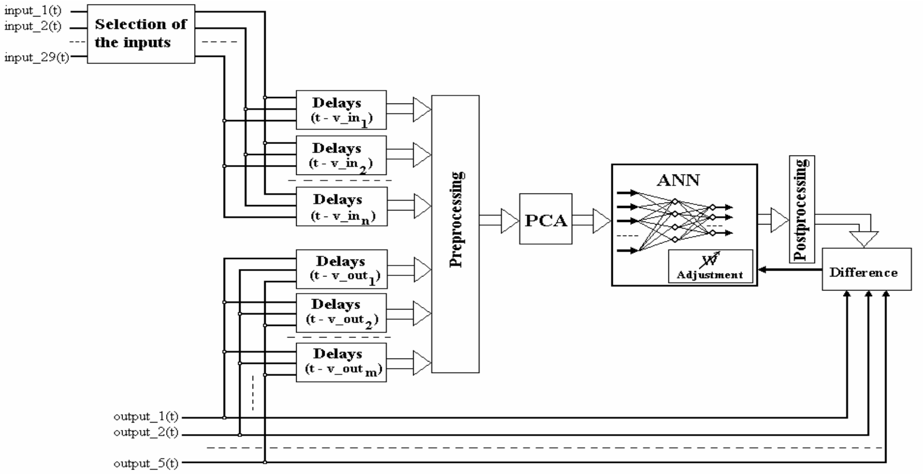


Figure 5-6 Example of an ANN model structure for DM regression task with input selection and input and output delay vectors. Principal Component Analysis (PCA) is used to pre-process the ANN input.
(Source: *Publication* 6)

The recurrent relation performed by our model is as follows:

$$Y(t+1) = F(X(t+1-Vect\_In(i)), Y(t-Vect\_Out(j))) \qquad (5\text{-}25)$$

where: $i = 1,...,n$; $j = 1,...,m$.

Other differences in designing an ANN for classification compared with an ANN for forecasting may come from data pre-processing, training procedure, and evaluation criteria.

In time-series problems the dependency in time between different outputs and different inputs leads to a large number of influences on a single output. Therefore, we need techniques to reduce and uncorrelate the input space (the first pre-processing approach). Principal Component Analysis (PCA) is a technique that does just that: it reduces the dimensionality of the input space and uncorrelates the inputs (Jackson, 1991; Bishop, 1995). PCA seeks to map vectors x^n in a d-dimensional space $(x_1, x_2, ..., x_d)$ on to vectors z^n in an M-dimensional space $(z_1, z_2, ..., z_M)$, where $M < d$. PCA is based on two elementary linear algebra concepts: eigenvectors and eigenvalues which satisfy the following relationship:

$$AX = \lambda X \tag{5-26}$$

where A is the covariance matrix of the input variables, X is an eigenvector of matrix A and λ is the eigenvalue associated with that eigenvector. After calculation of eigenvectors with their associated eigenvalues, the PCA can be used to compute optimal dimension reduction in the sense of *loss of variance* by projecting the data on to the subspace spanned by the eigenvectors with greatest eigenvalues (Alhoniemi, 2002, p. 16). The Neural Network toolbox of Matlab (MathWorks, Inc., 2001) offers a function [transformed_input, transMat] = *prepca*(input, min_frac) which reduces the input space by retaining those eigenvectors that contribute more (have greater eigenvalues) than a specified fraction *min_frac* of the total variation in the dataset. *prepca* has as parameters the *input space* and *min_frac* and returns the *transformed input space* and the *transformation matrix*. Before applying PCA (Jackson, 1991) we normalise both the inputs and the outputs to zero mean and unit standard deviation. We have applied the reverse process of normalisation in order to denormalise the simulated outputs.

The ANN architecture is determined as in the classification case (see Section 5.2.3). We employ Masters' rule (Masters, 1994) to determine the number of hidden neurons. We tested ANNs with both one and two hidden layers and compared them using the model evaluation criteria. The retraining procedure is the same as in the classification case. However, here we apply two retraining mechanisms sequentially. Firstly, RT1 is applied to the network obtained when determining the ANN architecture. Then the result of RT1 is further improved by applying RT3. Consequently, for one single combination of the delay vectors we had three models: one as a result of determining ANN architecture, another as a result of applying RT1, and a third as a result of applying RT3.

As evaluation criteria for the ANN used in forecasting we usually have the mean of the squared differences between real and simulated outputs. This differs from the accuracy rates in the case of ANN classification models. Other formulas can also be used to evaluate the performance of the ANN as a predictor. For example, when modelling the glass-manufacturing process we computed the error ERR (EUNITE competition, 2003), which shows how well we estimated all output data:

$$ERR = \frac{1}{5}\sum_{i=1}^{5}\frac{100}{N}\sum_{k=1}^{N}\frac{|O_{Rki} - O_{Fki}|}{|O_{Rki}|}\cdot f(k) \tag{5-27}$$

where N is number of time steps (observations), O_{Rki} is the real output i at time step k, O_{Fki} is the forecast output i at time step k, and $f(k) = \frac{500}{500 + k}$ is a weight function decreasing with the number of time steps k.

This error (ERR) can be calculated based on the training (ERR_A) or test (ERR_T) set. To calculate ERR_A we split the training dataset into n distinct intervals and calculate for each interval one ERR. ERR_A is the average of these ERRs. We can have different models depending on the model structure (how outputs depend on inputs), the ANN architecture (one or two hidden layers, varying number of neurons in the hidden layer(s)) and calculate for each model one ERR_A and ERR_T. Then, the error vectors can be compared to check whether there is any correlation between the two errors. If there is a correlation, then we can consider ERR_A a good "selection tool" for our ANN model and we can use it when the output for the test data is not known.

ANNs for the DM regression task – Research Issues

The regression task matches many real-world problems ranging from process control prediction problems to predicting companies' monthly/yearly accounts. Artificial neural networks represent one data-mining technique that can perform the DM regression task. There are different types of ANNs, but that which is most used in performing the DM regression task is the multilayer perceptron (MLP). In this paragraph we present some examples of studies that have used ANNs in forms of MLPs to perform the DM regression task. We discuss the ANN research issues addressed in our study when they are applied to the DM regression task. MLPs are the most commonly used form of neural networks. MLPs have been applied extensively in many engineering applications to perform regression tasks.

Khalid & Omatu (1992) proposed neural networks trained with back-propagation to model a temperature control system. The authors compared the neural network to a conventional proportional-plus-integral controller and found that "the neural network performs very well and offers worthwhile advantages".

Yu *et al*. (2000) used ANNs to model a chemical reactor process. The chemical reactor consists of a stirring tank to which chemical solutions and air are added. The outputs of the system are: the liquid temperature (T), the liquid hydrogen concentration (pH), and the percentage of dissolved oxygen (pO_2). The flow rate (f_b) of one chemical solution (ammonium hydroxide – NH_4OH) is used to regulate the pH. The airflow rate (f_a) is used to regulate the pO_2. The tank is also equipped with a heating system to adjust the temperature of the liquid – T. The three inputs (chemical solution and airflow rates and heating power – Q) and the three outputs constitute a Multi-Input-Multi-Output (MIMO) non-linear dynamic system. The authors trained a MLP for each of the three outputs. The MLPs use different delay vectors depending on the output that is forecast. For example, the heating temperature at moment t – $T(t)$ – depends on $T(t\text{-}1)$, $Q(t\text{-}22)$, and $f_a(t\text{-}1)$. The predictions for the outputs are further passed to a numerical optimisation routine that attempts to minimise a specified cost function (Yu *et al*., 2000). The result of the optimisation is the optimal control variables (the process input operating ranges).

Venayagamoorthy *et al*. (2001) used MLPs as neuroidentifiers of different turbo generators in a 3-machine 6-bus power system. The MLP output at time *k*+1 depends on both the past *n* values of the output and past *m* values of the input (Venayagamoorthy *et al*. 2001, p. 1268). This input-output representation form was chosen to avoid a feedback loop in the model and to correctly identify the dynamics of the turbo generator. The MLP consists of 12 inputs (4 real inputs: the deviation in the actual power to the turbine, the deviation in the actual field voltage to the exciter, the deviation in the actual terminal voltage, and the deviation in the actual speed of the turbo generator plus 8 delayed values), one hidden layer and 2 outputs (the estimated terminal voltage deviation and the estimated speed deviation of the generator). The number of neurons in the hidden layer is determined empirically. The training algorithm used was back-propagation. The authors used two sets for training: in *forced training* the 12 inputs described before are used, whereas in *natural training* the outputs depend only on their past values. The network obtained after the forced training is trained further with the natural training procedure. The experimental results show that neuroidentifiers are very promising in identifying highly non-linear MIMO turbo generators.

In addition to process control applications the DM regression task corresponds to other real-world problems. For example, Koskivaara (2004b) proposed an ANN for predicting companies' monthly/yearly accounts from the financial statements. The author proposed an ANN-based prediction system (ANNA) that has been developed in several stages. ANNA.01 used an MLP trained using the back-propagation (BP) algorithm to model the dynamics of different monthly accounts, i.e. to find the function that approximates the relation between account *x* at moment *t*+1, *x*(*t*+1) and three of its past values *x*(*t*-2), *x*(*t*-1), and *x*(*t*). ANNA.02 models used different delay vectors (ANNA.021 used two previous values to predict the third one, while ANNA.022 used four previous values to predict the fifth one) and the most important accounts were chosen in collaboration with a Certified Public Accountant. ANNA.03 is an extension of ANNA.02 with different ANN parameters being tuned. Also new dataset comprising 72 monthly balances of a manufacturing firm was used to test the models. The ANNA.04 models were based on ANNA.03 but here different data pre-processing methods are tested to ensure that the best, in terms of prediction error, is chosen. The ease of use and good visualisation capabilities of the ANNA system make it a feasible tool for supporting analytical review (AR) in auditing.

Compared with the above studies and other related studies that used ANNs (in the form of MLPs) for the DM regression task, our aim is reduced to enhancing the applicability of ANN as a prediction tool, specifically to solving prediction problems involving process control variables. We concentrate on ANN predictions and address some technical problems related to the ANN architecture or ANN prediction performance in the context of process control applications. We introduce an empirical procedure to determine the ANN architecture for performing DM regression tasks. The procedure is similar to the one used to perform the DM classification task. We tackle the problem of how to improve the predictive performance of ANNs when they are used in DM regression tasks by proposing an alternative way of training the ANN based on its past training experience and weights reduction. As our experiments show (Section 6.3.1), the retraining procedure greatly improves the ANN's prediction performance. We use as an experimental study the prediction of control variables of the manufacturing process at the Schott Company, a German glass manufacturer.

In our study we are not interested in comparing different methods for predicting process control. Nonetheless, there are many studies in the literature that compare different techniques for performing the regression task. One recent publication is Razi & Athappilly (2005), which compared ANNs, non-linear regression and regression and classification tree (CART) models in terms of prediction performance. The ANN and CART model outperformed non-linear regression models in terms of predictive performance. Moody (1995) used an ANN as a macroeconomic prediction tool and applied his models to predict the US Index of Industrial Production. The author compared the ANN model with three other models (trivial, univariate and multivariate linear models) in terms of normalised prediction errors, and found out that the neural network model significantly outperformed the trivial predictors and linear models. Kumar (2005) found that statistical regression models outperformed the neural networks when the output variable is skewed. However, the author argues that the two techniques (statistics and ANN) can benefit from one another, i.e. if skewness is present, the data can be transformed using power transformation to reduce the skewness before carrying out neural network analysis (Kumar 2005, p. 430).

Chapter 6 CI Methods in Practice – Experiments

In this chapter, we apply the CI methods presented in the previous chapter using a number of experiments. Statistical methods are used as methods of comparison. Firstly, we assess comparatively the economic performance of Central-Eastern European countries. Secondly, we benchmark companies from two large industrial sectors – the pulp-and-paper and telecommunications sectors – as to their financial performance. Finally, we test our ANN regression tool by making predictions for process control variables for making glass at Schott, a German glass manufacturer.

In Table 6-1 we present the link between the real-world applications and CI methods used to address them, together with the corresponding publications.

Table 6-1 The link between DM tasks, CI methods and experiments

Experiment		Central-eastern European countries' economic benchmarking	Pulp-and-paper companies' financial benchmarking	Telecom companies' financial benchmarking	Schott glass manufacturing process control variables prediction	Chapter/ Section
DM task	Statistical/ CI method					
Clustering	SOM	Costea *et al.* (2001) Publication 2 Costea (2003)	Publication 2	Costea *et al.* (2002a, b) Publication 3	-	5/5.1.1
	C-Means	Costea *et al.* (2001)	-	-	-	5/5.1.2
	FCM	-	-	Alcaraz and Costea (2004a) Publications 4, 5	-	5/5.1.3
	WFCM	-	-	Alcaraz and Costea (2004a) Publication 4	-	5/5.1.3
Classification	MLR	Publication 2	Publication 2	Costea *et al.* (2002a, b) Publication 3	-	5/5.2.1
	DT	Publication 2	Publication 2	Costea *et al.* (2002a, b) Publication 3	-	5/5.2.2
	ANN	Costea (2003)	-	Costea and Nastac (2005) Publication 5	-	5/5.2.3
	RT-based ANN	-	-	Costea and Nastac (2005) Publication 5	-	5/5.2.3
	GA-based ANN	-	-	Costea and Nastac (2005) Publication 5	-	5/5.2.3
Regression	RT-based ANN	-	-	-	Nastac and Costea (2004a) Publication 6	5/5.3
Chapter / Section		6 / 6.1.1	6 / 6.2.1	6 / 6.2.2	6 / 6.3.1	

6.1 Countries' Economic Performance Competitor Benchmarking

The first experiment concerns assessing the economic performance of certain Central-Eastern European countries. We present our findings regarding the economic performance comparisons in Costea *et al.* (2001), *Publication* 2, and Costea (2003).

6.1.1 Central-Eastern European Countries

To measure the countries' economic performance we based our choice of economic variables on the following Convergence Criteria imposed on countries that signed the Maastricht Treaty. The suggested macroeconomic actions concerned inflation (less than or equal to 1.5% + average of the three most stable countries), national deficit (less than or equal to 3% of the Gross National Domestic Product – GNDP), national debt (less than or equal to 60% of GNDP), exchange rate (to fluctuate by no more than X% from the interval established by the European Monetary System – e.g. for EU members X = 2.5%), interest rate (less than or equal to 2% + average of the three most stable countries).

We characterise countries' economic performance with the aid of the following indicators:

- Currency Value (CV) is the inverse of the Exchange Rate (ER), and shows how many US dollars one can buy with 1000 current units of national currency and depicts the purchasing power of each country's currency,
- Domestic Prime Rate (Refinancing Rate – RR), which shows financial performance and level of investment opportunities. This interest rate is established by the central bank of each country and is the interest rate for refinancing the operations of the commercial banks. Hence, it affects all other interest rates.
- Industrial Output (IO[17]) compared to previous periods in percentages, to depict industrial economic development,
- Unemployment Rate (UR), which characterises labour exploitation and, more generally, the social situation in the country, and
- Foreign Trade (FT) in millions of USA dollars, to reveal the surplus/deficit of the trade budget.

The dataset consists of monthly/annual data for six countries (Russia, Ukraine, Romania, Poland, Slovenia and Latvia) during the period 1993-2000, in total 225 cases with five variables each (see Appendix). In Costea *et al.* (2001) there were two more variables in the dataset: exports (EXP) and imports (IMP) in millions of USD, as intermediary measures to calculate the foreign trade. We discarded them

[17] Industrial Output was preferred to GDP per capita as the latter is an annual indicator and we needed monthly data.

in the later studies as these variables are strongly correlated with the foreign trade variable. Also, we replaced the first variable (Exchange Rate) from Costea *et al.* (2001) with the Currency Value variable to ensure comparability between the different countries' currencies. We encountered in some cases missing values, which we have complemented using the means of existing values.

Experiment 1

In Costea *et al.* (2001) we applied our SOM (Section 5.1.1) and compared it to C-Means clustering (Section 5.1.2) to group the countries according to their economic performance. We standardised the data to zero mean and unit standard deviation ("normalisation"). In the case of SOM we performed a two-step clustering: firstly, we built larger maps that contained "raw" clusters and then, we re-grouped the "raw" clusters to form a smaller number of "real" clusters. Vesanto & Alhoniemi (2000) used a similar two-level clustering approach; they found that two-level clustering was computationally more effective than applying the clustering methods directly, while the achieved results were similar.

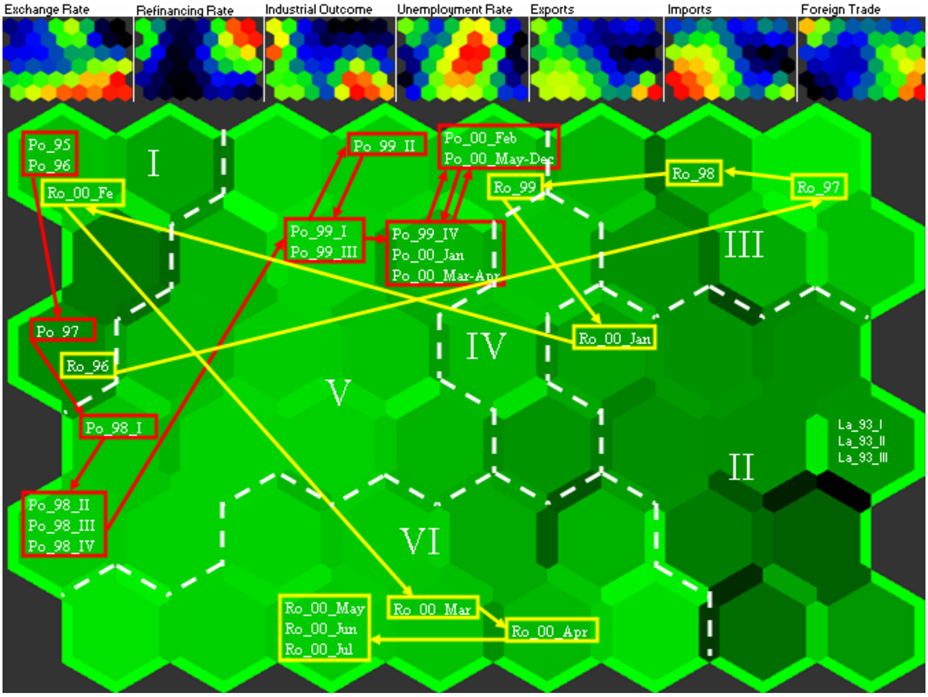


Figure 6-1 The final 8x6 SOM for countries' datasets with identified "real" clusters and feature planes. The borders of the "real" clusters are identified by the dotted lines. Feature planes for each economic variable are shown at the top of the figure ("warm" colours indicate high values, whereas "cold" colours indicate small values). The economic variables are presented at the beginning of Section 6.1.1. Trajectories for Poland (red) and Romania (yellow).

We trained several SOM maps and chose the best in terms of average quantisation error (Equation 5-6) and ease of readability. The experiment parameters were set using the rules of thumb described in Section 5.1.1. The parameters used to train the final map were: $X = 8$, $Y = 6$, $rlen_1 = 2400$, $\alpha_1(0) = 0.5$, $N_1(0) = 8$, $rlen_2 = 24000$, $\alpha_2(0) = 0.05$, $N_2(0) = 1.3$. Indexes 1 and 2 denote the two training phases: 1 – the "rough" phase, and 2 – the "fine tuning" phase. The final trained SOM map (8x6) and the feature planes for each variable are illustrated in Figure 6-1. We subjectively identified six "real" clusters of economic performance by studying the feature planes and the observations corresponding to each cluster.

The first group of countries formed cluster **I** represented in the left-hand top corner of the map. The economic condition of the countries can be characterised by a relatively high national currency value against the US dollar, high unemployment rate, high industrial output index, and small positive foreign trade balance. Cluster **II** is characterised by very low unemployment rates, moderate refinancing rates, a rather high national exchange rate together with relatively high growth of industrial output and good foreign trade balance. Cluster **III** is situated in the right-hand top corner of the map. The cluster is characterised by the best national currency value, a moderate unemployment level, high refinancing rate and decreasing industrial output. The countries with critically poor economic performance or facing financial crisis are grouped in cluster **IV**. Cluster **V** is the largest, in the middle of the map. The economic performance of the countries consists of: the exchange rate is low to moderate; the unemployment rate is moderate, industrial output is rather low, and the foreign trade balance is moderate. Finally, cluster **VI** at the bottom of the map contains economic performance characterised by high exchange rates, stable industrial growth, but the worst trade balance. The characteristics of the identified clusters are summarised in Table 6-2. We characterise each cluster, using for each variable the following linguistic terms: VL – very low, L – low, A – average, H – high, VH – very high. The last column of the Table 6-2 gives the overall characterisation of each cluster.

Table 6-2. S*ubjective* characterisation of the economic clusters based on the feature planes. Each cluster is characterised subjectively by human interpretation of the feature plane from Figure 6-1 (e.g. H or VH linguistic term corresponds to "warm" colours). The economic variables are presented at the beginning of Section 6.1.1.

	ER	**RR**	**IO**	**UR**	**EXP**	**IMP**	**TB**	**Order**
Cluster 1	VL	A&H	H	A	A	L	A&H	Average
Cluster 2	VH	L&A	VH	VL&L	H&VH	VL&L	VH	Best
Cluster 3	VL	VH	VL	L&A	L	L&A	L	Good
Cluster 4	A	A	L	VH	VL	L	L	Worst
Cluster 5	L&A	VL	L&A	A&H	A	H	A	Average
Cluster 6	H	VL	L&A	A&H	H	H&VH	VL	Bad

Using *trajectories* (Alhoniemi *et al.*, 1999) we can reveal the path every country followed during the transition period compared to the performance of other countries. The map shows the movements for two of the six countries included in our study. Romania starts in 1996 in Cluster I (with strong currency, but high interest rates), moves to cluster III (here, the interest rates are even higher), stays there in 1997-1998, moves to cluster V in 1999 and from there stabilises in cluster VI in 2000. Each movement can be analysed more closely. Romania's movement from cluster III to cluster V was due to its doubled exchange rate in 1999, even though there it reduced its interest rates. The massive imports at the end of 1999 and beginning of 2000 dropped Romania to cluster VI. Altogether Romania has been extremely unstable. Poland, on the other hand, showed rather stable economic performance over the1996-2000 period, moving from cluster I to cluster V at the cost of high unemployment. Apart from the geo-strategic advantages, the economic reforms and the closing of the inefficient state-owned enterprises made Poland a stronger competitor in joining the EU compared to Romania, which, finally, led to Poland's acceptance by the EU in 2004.

We compared our SOM-based results with results obtained with C-Means. We set the number of clusters to six to make the results comparable. We ran the C-Means algorithm with SPSS and used the final SOM map with the "raw" clusters to see how C-Means formed the "real" clusters. The results were poorer in the sense that with C-Means we obtained two "real" clusters with few observations: one had one observation and another had three. This is due to the sensitivity of C-Means to the initialisation of the clusters' centres and outliers as pointed out in Section 5.1.2. The C-Means cluster with three observations contained the observations of Latvia's economic performance in 1993 (quarters I, II, and III). These observations are situated in cluster II of the SOM map (see Figure 6-1). It seems that Latvia in 1993 was in better economic shape than that exhibited by SOM cluster II and acts as an outlier for this cluster. Another advantage of the SOM over C-Means comes from the visualisation capabilities of SOM. However, the validation of map dimensionality and of the quantisation error as well as the automation of "real" cluster construction were not yet addressed in Costea *et al.* (2001). In our later experiments (telecommunications sector) we address the above issues.

Experiment 2

In *Publication* 2 and Costea (2003) we go one step further and use the clustering results (of SOM) to build *hybrid* classification models to help position new countries' performance within the existent economic performance clusters. In *Publication* 2 we use MLR and DT techniques to build the classification models, while in Costea (2003) we use ANN classifiers. The dataset is reduced to 5 variables, as explained at the beginning of this section. In *Publication* 2 we retrained the dataset (now with two variables less and one variable, Exchange Rate, replaced by Currency Value) using different parameters for the SOM. Again we "normalise" the dataset as we did in Costea *et al.* (2001). Finally, we obtained a 7x5 SOM map and identified seven "real" clusters (see Figure 6-2). The parameters used to train the final map were: $X = 7$, $Y = 5$, $rlen_1 = 1750$, $\alpha_1(0) = 0.5$, $N_1(0) = 7$, $rlen_2 = 17500$, $\alpha_2(0) = 0.05$, $N_2(0) = 1.2$. The final U-matrix map and feature planes were easier to read compared with the final map in Costea *et al.* (2001), making the process of constructing the "real" clusters easier.

The alphabetical order of the cluster identifiers corresponds to the inverse order of economic performance: A – best performance, B – slightly below best performance, C – slightly above average performance, D – average, E – slightly below average performance, F – slightly above poorest performance, and G – poorest performance.

Again, the SOM trajectories can be used to check the economic performance of the different countries over time. For example, Ukraine made steady progress between 1993 and 2000 with respect to its foreign trade balance (Figure 6-2). In 1993, in spite of its high currency value, Ukraine had the worst economic situation (high negative values for foreign trade), which positioned the country in the worst cluster. Year by year the trade balance improved and in 2000 (April, May, June) became positive, which led to Ukraine being placed in the best economic performance cluster (cluster A).

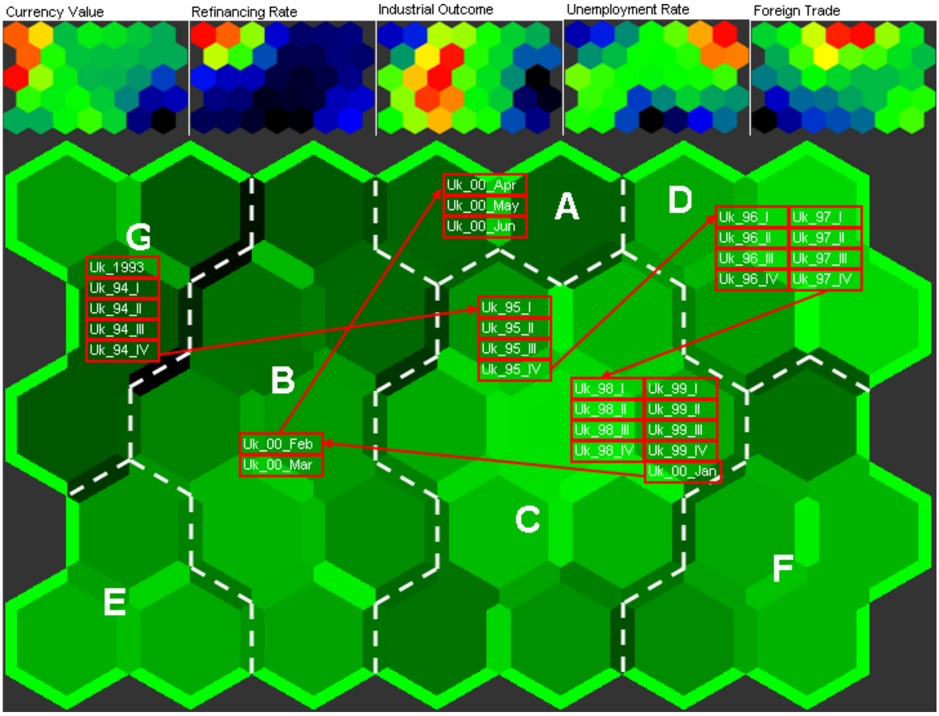


Figure 6-2. Final 7x5 SOM for countries' dataset with identified "real" clusters and feature planes. The borders of the "real" clusters are identified by the dotted lines. Feature planes for each economic variable are shown at the top of the figure ("warm" colours indicate high values, whereas "cold" colours indicate small values). The economic variables are presented at the beginning of Section 6.1.1. Trajectories for Ukraine (red).

Once we had constructed the "real" clusters we built the class variable, assigning a class value (1 to 7) to each observation within a cluster. Next, we applied MLR, DT, and ANN to build the classification models by following the methodological steps from the beginning of Section 5.2. We replaced the missing data with the means of existing values. We used SPSS, SEE5, and Matlab scripts to build MLR, DT (*Publication* 2) and ANN (Costea, 2003) classification models. We standardised the input data to zero mean and unit standard deviation (normalisation). We validated our models based on the training data by using proportional by-chance and maximum by-chance accuracy rates. For example, in the case of MLR the training accuracy rate (61.3%) satisfied the proportional by-chance criterion (61.3% > 1.25*29.92% = 37.4%) but slightly failed to satisfy the maximum by-chance criterion (61.3% < 1.25*49.8 = 62.22%). The significance of the Chi-Square statistic (p<0.0001) and the overall correlation coefficient (Nagelkerke's R^2 = 74.5%) show a relatively strong relationship between class variable and the economic variables.

We interpret the results of MLR by looking at the SPSS output tables. All variables are statistically significant (sig.>0.0001) in explaining the likelihood variations in the dependent variable. Some coefficients in the regression equations are not statistically significant. For example, in differentiating between the average and poorest performance classes (regression equation 4), the "Industrial Output" variable is the only variable that is not statistically significant (sig. of *Wald* statistic = 0.229 > 0.05). Some values in "Std. Error" column are greater than 2, which indicates a multicolinearity problem for our economic dataset. Variable "Unemployment Rate" has a value of 2.086 in column "Exp(B)" for the 5th regression equation, which means that for each unit increase in this variable the likelihood that the observations will be classified in class E (slightly above average) increases by approximately two times. Next, we try to validate our models using the general procedure described at the beginning of Section 5.2 (see Table 6-3).

Table 6-3. Accuracy rate validations for the economic classification models. The techniques used are shown in the first column: MLR, DT, and ANN. The validation is done according to step 5 of the methodology presented at the beginning of Section 5.2.

Technique		Main dataset	Part1 (split=0)	Part2 (split=1)
MLR	Learning Sample	61.3%	67%	58.4%
	Test Sample	no test sample	57.6%	67.1%
DT	Learning Sample	79.1%	77.7%	78.86%
	Test Sample	no test sample	46.9%	54.5%
	Cross-validation	64%	no cross-validation	no cross-validation
ANN	Learning Sample	43%	53.57%	55.36%
	Test Sample	no test sample	42.11%	38.05%
	Cross-validation	31.12%	no cross-validation	no cross-validation

The results of all our classification techniques are rather poor for this experiment. There are major discrepancies between the training and test accuracy rates. Moreover, the classifiers did not learn very well the patterns within the data. The best performer was the decision tree with a training accuracy rate of almost 80 per cent. Both MLR and DT outperformed ANN in this case. The reason for this might be that we chose an ANN architecture with a single output neuron and forced the ANN to learn outputs between 1 and 7. We show in our next experiments how the ANN accuracy increases when we have as many neurons in the output layer as the number of classes. However, ANNs need a lot of data to be trained (usually ten times more than the number of weights) and we had a rather small economic performance dataset. In the case of DT, we had two parameters: minimum number of observations in each leaf node ($m = 5$), and the confidence factor used in pruning the tree ($c = 25\%$). The best discriminant variable is "Unemployment Rate".

In the ANN case (Costea, 2003) we kept all parameters constant except two: the number of hidden neurons (NH) in the hidden layer, which varied between 5 and 9 (cf. Lachtermarcher & Fuller, 1995) and the training algorithm. We tested four training algorithms: Scale Conjugate Gradient (SCG), Levenberg-Marquardt (LM), Resilient Back-Propagation (RP) and Gradient Descent (GD) for each ANN architecture. The best network in terms of test accuracy rate had nine hidden neurons and used standard gradient descent back-propagation training (GD). However, LM outperformed the others in terms of convergence speed.

6.2 Companies' Financial Performance Competitor Benchmarking

As we mentioned in Section 4.1.3, we use financial ratios to assess companies' financial performance. The most common financial ratio classification has proposed the following categories for financial ratios: *profitability*, short-term solvency (*liquidity*), long-term *solvency*, and *efficiency* ratios (Lev 1974, p. 12; Lehtinen 1996, p. 44). *Static* ratios are constructed with entries from the balance sheet, whereas *dynamic* ratios are constructed with entries from the income statement. When both income statement and balance sheet are used to calculate the ratio we have a *mixed* ratio.

The first performance dimension, *profitability*, is the most important performance measure since a company's financial performance is highly dependent on its earnings. Companies must "remain profitable in order to survive and also in order to maintain both good liquidity and solvency" (Eklund 2004, p. 52). In line with Lehtinen (1996, p.50) the profitability ratios chosen for our experiments are the following: *operating margin*, *return on equity*, and *return on total assets*. Operating margin is a dynamic ratio and shows the percentage of the net profit (minus extraordinary and profit distribution items) over total net sales. Return on equity and return on total assets are mixed ratios and show the profitability of the capital supplied by the common stockholders and the profit per unit of asset respectively.

The second performance dimension, *liquidity,* measures the degree to which the company is able to meet its short-term financial obligations. There are two static liquidity ratios that we used in our experiments: *quick ratio* in the case of the pulp-and-paper dataset and *current ratio* in the case of the telecom dataset. The difference between the two ratios is that the quick ratio does not include inventories as liquid assets. This was the correct choice for the pulp-and-paper dataset since most of the pulp-and-paper companies “had rather large inventories, and these inventories are in some ways not nearly as liquid or stable as other current assets” (Eklund 2004, p. 53).

Solvency refers to the ability of the companies to meet their long-term financial obligations. Two solvency ratios were used in our experiments: *equity-to-capital* and *interest coverage*. Equity-to-capital is a static ratio and shows how much debt (equity) is used to finance the assets of the firm. Interest coverage is a dynamic ratio and shows the firm’s ability to manage its daily interest expenses concerning long-term debts (Lehtinen 1996, pp. 67-68).

The *efficiency* ratio used in our experiments was *receivables turnover*. This ratio measures the efficiency of the firm in collecting receivables. There are other ratios that measure different aspects of a company’s efficiency: storage, assets and working capital efficiency. They are all mixed ratios. We chose only receivables turnover in our experiments as this is the most commonly used efficiency ratio and in line with Lehtinen’s experiments (Lehtinen, 1996).

The ratios were calculated with the aid of information from the companies’ annual reports. In Table 6-4 we present the formulas for the chosen ratios together with the degree of their validity and reliability in international comparisons.

Table 6-4 Formulas, validity and reliability of financial ratios

Dim.	Ratio	Formula	Validity	Reliability
Profitability	*Operating Margin*	$OM=\frac{Operating\ Profit}{Net\ Sales}\times 100$	Low[18]	High
	Return on Equity	$ROE\ =\ \frac{Net\ Income}{(Share\ Capital+Retained\ Earnings)\ Average}\times 100$	High	Good
	Return on Total Assets	$ROTA=\frac{Total\ Income+\ Interest\ Expense}{(Total\ Assets)\ Average}\times 100$	High	High
Liquidity	*Quick Ratio*	$QR\ =\ \frac{Current\ Assets\text{-}Inventory}{Current\ Liabilities}$	Good	High
	Current Ratio	$CR\ =\ \frac{Current\ Assets}{Current\ Liabilities}$	Good	High
Solvency	*Equity to Capital*	$EC=\frac{Share\ Capital\ +\ Retained\ Earnings}{(Total\ Assets)\ Average}\times 100$	Good	Good
	Interest Coverage	$IC=\frac{Interest\ Expense+Income\ Tax\ +\ Net\ Income}{Interest\ Expense}$	High	High
Efficiency	*Receivables Turnover*	$RT=\frac{Net\ Sales}{(Accounts\ Receivable)\ Average}\times 100$	High	High

(Cf.: Lehtinen 1996, pp. 60-70)

[18] This ratio is very popular among practitioners as are all margin ratios (Lehtinen 1996, p. 50).

6.2.1 Pulp-and-Paper Industry

In this experiment we assess comparatively the financial performance of different international pulp-and-paper companies. Our results are presented in *Publication* 2. The dataset[19] spans from 1995 to 2000 and consists of the annual financial ratios of 77 companies taken from their income statements and balance sheets (see Appendix). Also, annual averages were calculated for each of the following regions: Finland (4), Sweden (7), Norway (2), USA (30), Canada (12), Japan (13), and Continental Europe (9). The companies were chosen in accordance with annual rankings (based on net sales) from *Pulp and Paper International*'s report (Rhiannon *et al.*, 2001). In total, the dataset consisted of 474 rows of data and the seven financial ratios suggested in Lehtinen (1996) (see Table 6-4). Further details in choosing the companies and how the data were collected can be found in Eklund (2004, pp. 67-71).

Experiment 1
In *Publication* 2 we apply our two-level methodology to build *hybrid* models for classifying financial performance. Firstly, SOM is applied to the pulp-and-paper dataset. Then, with the newly constructed class variable based on the SOM output, MLR and DT techniques are used to build the classification models. The final 7x5 SOM map with the identified "real" clusters is shown in Figure 6-3. SOM parameters: $X = 7$, $Y = 5$, $rlen_1 = 1750$, $\alpha_1(0) = 0.5$, $N_1(0) = 9$, $rlen_2 = 17500$, $\alpha_2(0) = 0.06$, $N_2(0) = 1$.

The data were pre-processed using histogram equalisation technique (Guiver & Klimasauskas, 1991). We tested other pre-processing techniques such as normalisation according to standard deviation or variance but the results were poor: the maps were flat except for some regions at the extreme ends of the map (Eklund 2004, p. 76).

The labels of the clusters in Figure 6-3 are interpreted in the same way as for the economic performance dataset, cluster A containing the best performers, and cluster G the worst. We interpret each cluster by looking at the feature planes and at the observations that are included in the cluster. This is a *subjective*[20] way to characterise the clusters. Cluster A includes the best performing companies with very high profitability, high liquidity, very high solvency and high efficiency. Cluster B contains the second best companies with fairly good profitability, the highest liquidity, average solvency and efficiency. Cluster C is a slightly above average cluster characterised by slightly better profitability ratios than the average cluster D. In Cluster D all the performance dimensions have average values except solvency, which is good, and efficiency, which is somewhat high. Cluster E is

[19] The pulp-and-paper data (1995-2000) were collected by Tomas Eklund for his doctoral thesis (Eklund, 2004).

[20] In *Paper* 4 we introduce an *objective* way to automatically characterise each cluster using linguistic variables.

slightly below average with drops in profitability ratios, but rather good liquidity. Clusters F and G contain the worst performing companies, the companies in F being more liquid than those in G.

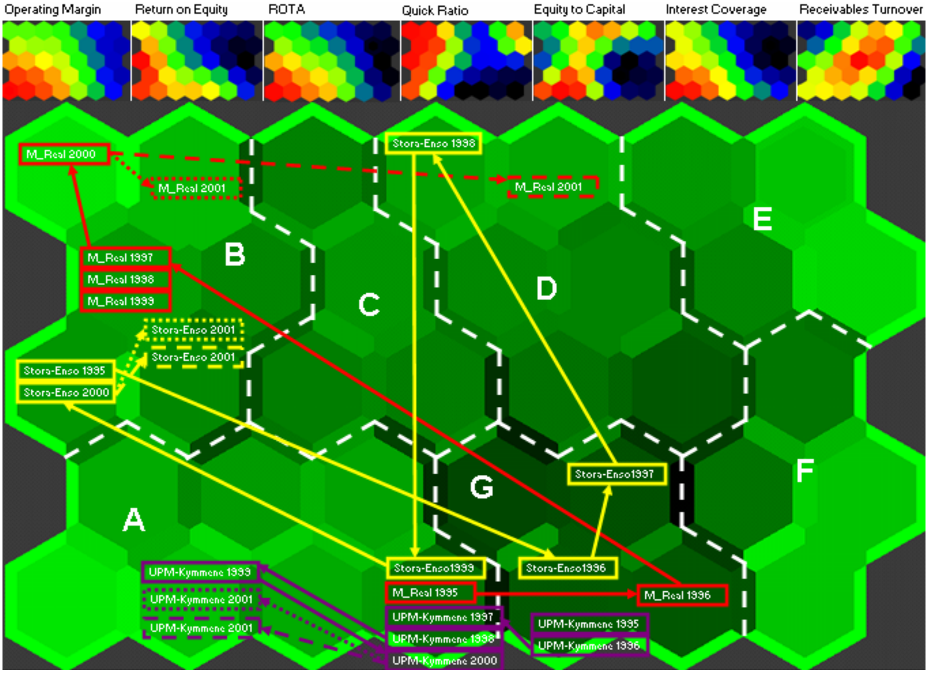


Figure 6-3. Final 7x5 SOM for the pulp-and-paper data set with identified "real" clusters and feature planes. The borders of the "real" clusters are identified by the dotted lines. Feature planes for each financial ratio are shown at the top of the figure ("warm" colours indicate high values, whereas "cold" colours indicate small values). The financial ratios are presented in Table 6-4. Trajectories (solid lines) for M-Real (red), Stora-Enso (yellow), and UPM-Kymmene (violet) between 1995 and 2000, and the predicted classes for 2001: MLR (dashed lines) and DT (dotted lines).

The trajectories in Figure 6-3 show the movements of the three largest Finnish companies, M-Real (red), Stora-Enso (yellow), and UPM-Kymmene (violet), between 1995 and 2000. M-Real was placed in the best cluster (A) in 1995, but dropped to the worst cluster (G) one year later. Then, in 1997 it recovered and it stabilised (1998, 1999, and 2000) in the second best cluster (B). Stora-Enso had a similar evolution. It started in 1995 in cluster B and moved for the next two years into the worst performance cluster (G) because of falling liquidity. In 1998 it moved to cluster D and it stabilised in the best performance clusters in the next two years. UPM-Kymmene was the best performer, being placed in the best performance cluster from 1997 to 2000.

Next, the class variable is constructed using the identified "real" clusters on the SOM map. Then, MLR and DT are applied following the methodology described at the beginning of Section 5.2. The dataset exceeded the limit of 15-20 training observations for each independent variable. There were few missing data and they were replaced using simple regression models (see Eklund 2004, pp. 68-69). SPSS and SEE5 software programs were used to build the classifiers. The training accuracy rates were validated with both proportional by-chance and maximum by-chance criteria for both MLR and DT. Nagelkerke's R^2 = 97.8% shows a very strong relationship between class and independent variables.

To interpret the results in the case of MLR we look at two tables: "Likelihood Ratio Test" and "Parameter Estimates" tables. The "Likelihood Ratio Test" table shows that all ratios are statistically significant (sig.>0.0001) in explaining the likelihood variations in the class variable. Also, the number of variable coefficients that are statistically significant in regression equations is doubled compared to the economic performance classification from experiment 1 ("Parameter Estimates" table). We had very few standard errors (column "Std Error" in "Parameter Estimates" table) that were above two, which means that the multicolinearity problem does not exist in this case. The best splitting ratio for the DT was "Return on Equity". The second best was "Equity-to-Capital" and "Receivables Turnover". In Table 6-5 we present the validation of the training accuracy rates for both techniques.

Table 6-5 Accuracy rate validations for the classification models (pulp-and-paper). The techniques used are shown in the first column: MLR, DT. The validation is done according to step 5 of the methodology presented at the beginning of Section 5.2.

Technique		Main dataset	Part1 (split=0)	Part2 (split=1)
MLR	Learning Sample	88%	89%	89,5%
	Test Sample	no test sample	76,1%	82,4%
DT	Learning Sample (75%)	84.8%	86.5%	86.5%
	Test Sample (25%)	74.6%	71.7%	76.8%
	cross-validation	74.4%	no cross-validation	no cross-validation

Compared with the economic performance dataset, here we achieved better results. The accuracy rates were higher, and the discrepancies between the training and test accuracy rates were smaller. MLR and DT achieved comparable results in terms of both training and test accuracy rates with MLR performing slightly better (88% vs. 84.8%, 89% vs. 86.5% and 89.5% vs. 86.5% for training, and 76,1% vs. 71.7% and 82.4% vs. 76.8% for testing). The reason for this small difference might be that we were restricted by the demo-version of the SEE5 software to no more than 400 training observations. Consequently, we had to split the main dataset to 75 per cent (356 observations) for training and the remaining 25 per cent (118 observations) for testing.

Another way to compare the two techniques is based on their predicting capabilities. We tested MLR and DT predictions with three new observations, which correspond to companies that published their 2001annual reports earliest. The prediction classes for both techniques are presented in Table 6-6.

Two out of three observations were placed in the same clusters by both MLR and DT. MLR placed M-Real 2001 in cluster D (in Figure 6-3 the dashed red line), while DT placed it in cluster B (in Figure 6-3 the dotted red line). M-Real in 2000 is included in cluster B by the SOM (in Figure 6-3 the solid red line). Compared with 2000, in 2001 the company's profitability dropped somewhat, whereas its solvency increased considerably. The coefficients of the regression equations in MLR indicate that the MLR technique placed greater emphasis on Equity-to-Capital, which is a solvency ratio. Cluster D has a higher solvency than cluster B, therefore MLR placed M-Real 2001 in cluster D. DT placed M-Real 2001 in the same cluster as M-Real 2000 because DT emphasised profitability ratios (especially ROE) and M-Real's profitability remained almost the same in 2001 as in 2000. Overall, we can conclude that the two classification techniques achieved similar results in terms of both accuracy rates and class predictions.

Table 6-6. Class predictions for 3 pulp-and-paper companies: M-Real, Stora-Enso, UPM-Kymmene in 2001. The financial ratios are presented in Table 6-4. The techniques used are: MLR and DT.

Company	OM	ROE	ROTA	QR	EC	IC	RT	Predicted Class	
								MLR	DT
M-Real	5.621597	17.75955	8.979317	0.857129	27.02372	2.314056	6.8226657	D	B
Stora-Enso	11.0069	15.31568	7.67552	0.830754	31.23215	4.189956	6.2295596	B	B
UPM-Kymmene	16.27344	22.78149	11.16978	0.629825	34.59247	5.205047	6.0291793	A	A

(Source: *Publication* 2)

Our *hybrid* classification models overcome one of the problems associated with the SOM models: insertion of new data into an existing SOM model. In order to place a new observation in the already trained SOM map we have to standardise the new observation according the pre-processing technique used in training the SOM. The histogram equalisation technique does not allow us to standardise a new observation. Moreover, this method "is not revertible if it is applied to values which are not part of the original value set" (Vesanto *et al.*, 2000). Our hybrid classification models place the new observations in the SOM maps regardless of how the data were pre-processed to train the SOM. An alternative solution is to retrain the SOM including the new observations. This is a much more time-consuming task than directly positioning the observations using the classification models.

6.2.2 Telecommunications Sector

The most elaborate experiment in our dissertation is concerned with assessing the financial performance of international companies from the telecommunications sector. Our results are presented in the following publications: Costea *et al.* (2002a, b), extended in *Publication* 3, applies SOM and the three classification techniques to assess telecom companies' financial performance, *Publication* 4 introduces Weighting FCM to benchmark the telecom companies, *Publication* 5 studies three factors that can affect the classification performance of ANNs in the telecom sector data.

The dataset[21] consists of 88 worldwide companies and the time spans from 1995 to 2001 (see Appendix). Sixteen observations for 2002 were used to test the prediction power of our classification models. Annual averages were calculated for each of the following regions: Northern Europe (10), Continental Europe (20), USA (32), Canada (6), and Asia (20). In total, the dataset consisted of 630 rows of data between 1995 and 2001 and 16 rows for 2002. The seven financial ratios used were the same as for pulp-and-paper experiment, except the "quick ratio", which was replaced by the "current ratio", and they were selected according to the Lehtinen (1996) study (see Table 6-4). Further details in choosing the companies and how the data were collected can be found in Karlsson (2002).

Experiment 1

Costea *et al.* (2002a) apply SOM and two classification techniques (MLR and DT) to assess the financial performance of telecom companies. We used data from 1995 to 1999 (462 rows) to build the SOM map and train our classification models and the data for 2000 for the Scandinavian companies (11 rows) to test the prediction power of the models. We also benchmark the Scandinavian companies between 1995 and 1999 using SOM trajectories. To avoid the SOM placing too much emphasis on extreme values the ratios were forced to take values in the closed interval [-50, 50] by taking off their peaks. Consequently, we obtained good SOMs in terms of both quantisation error and ease of readability. To train the SOM we standardise the dataset according to different standardisation methods: the standard deviation of the entire dataset, the standard deviation of each individual variable, the variance of the entire dataset, and the variance of each individual variable. Finally, the best map in terms of quantisation error and ease of readability was obtained for the standardisation according to the variance of the entire dataset (Figure 6-4). The parameters used to train the final SOM map were: $X = 9$, $Y = 6$, $rlen_1 = 2700$, $\alpha_1(0) = 0.4$, $N_1(0) = 13$, $rlen_2 = 27000$, $\alpha_2(0) = 0.03$, $N_2(0) = 1.3$. We identified six "real" clusters by studying the feature planes of the final SOM map and constructed the class variable.

The clusters can be characterised as follows:

- Cluster A_1 contains the best performing companies with high profitability, good solvency, but slightly worse liquidity. Sample companies: British Telecom (97-99), Nokia (97-99), Samsung (95, 99), etc.

- In cluster $A_{2,}$ SOM groups the second best performing companies with slightly lower profitability than cluster A_1 and strong liquidity and solvency. Sample companies: Benefon (95-97), Motorola (95), Sonera (98), etc.

[21] The telecom data (1995-1999) used in this study were collected by Jonas Karlsson in his early licentiate studies (Karlsson, 2002). We updated Karlsson's dataset with complete data for 2000 and 2001 and with available data for 2002.

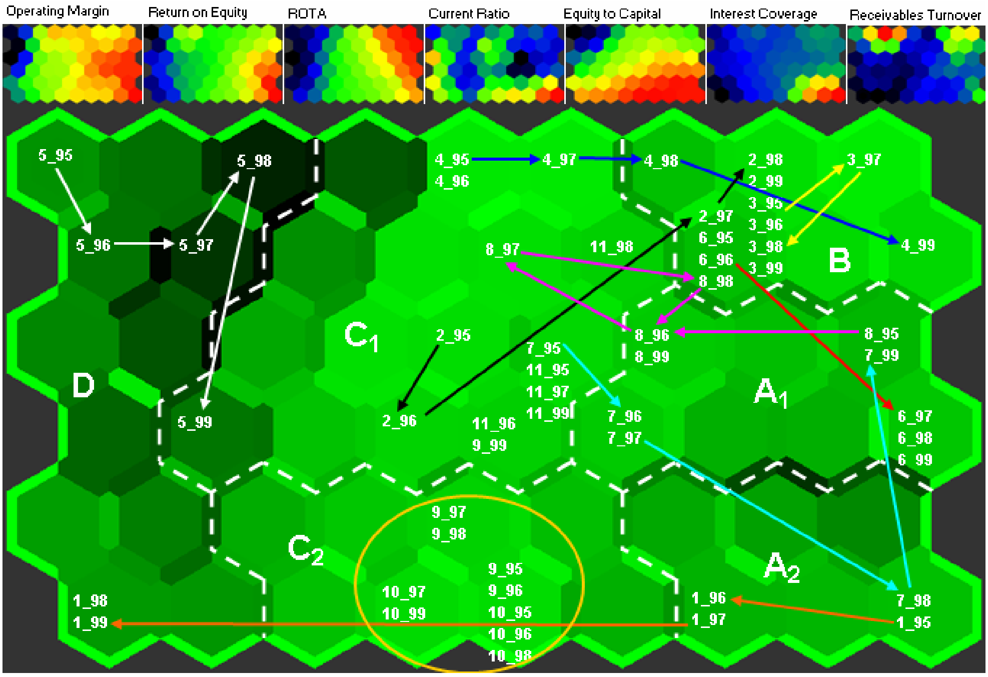


Figure 6-4. Final 9x6 SOM for the telecom data set with identified "real" clusters and feature planes. The borders of the "real" clusters are identified by the dotted lines. Feature planes for each economic variable are shown at the top of the figure ("warm" colours indicate high values, whereas "cold" colours indicate small values). The financial ratios are presented in Table 6-4. Trajectories for the Scandinavian companies between 1995 and 1999.
(Adapted from: Costea *et al*., 2002a)

- Cluster B includes companies with good profitability (especially ROE and ROTA ratios), poorer liquidity and solvency than A companies, but highest efficiency. Sample companies: Alcatel (97-98), Nokia (95-96), etc.

- Cluster C_1 exhibits average profitability, good liquidity, poorer solvency and efficiency. Sample companies: DoCoMo (95-99), Sonera (95), etc.

- Cluster C_2 resembles C_1 but liquidity and efficiency ratios are poorer than in C_1. Examples: British Telecom (95-96), Motorola (96-99), Telia (95-99), etc.

- The worst performers are grouped in cluster D, which exhibits poor profitability and solvency, and average to worst liquidity. It mainly contains service providers from Europe and USA, and Japanese companies in 98-99 (because of the Asian financial crisis that peaked in 1997-98).

After we attached the classes to each data row, we applied the two classification techniques MLR and DT as we did in our previous experiments. The training and testing accuracy rates for both MLR and DT are presented in Table 6-7.

Table 6-7. Accuracy rate validations for the classification models (telecom – exp. 1). The techniques used are shown in the first column: MLR and DT. The validation is done according to step 5 of the methodology presented at the beginning of Section 5.2.

Technique		Main dataset	Part1 (split=0)	Part2 (split=1)
MLR	Learning Sample	92,4%	90,5%	99,6%[22]
	Test Sample	No test sample	83,9%	85,5%
DT	Learning Sample (75%)	95,1%	91,8%	93,5%
	Test Sample (25%)	87,9%	89,6%	85,7%
	cross-validation	86,4%	no cross-validation	no cross-validation

Both classification techniques achieved high accuracy rates in both training and testing. Also, both MLR and DT training accuracy rates are validated against the testing accuracy rates (small differences).

In Figure 6-4 we show the financial performance of the Scandinavian telecommunications companies during 1995-99.

- Benefon (No. 1, orange arrows), a small Finnish mobile phone manufacturer, shows excellent performance during the years 1995-97, remaining in Group A_1. However, the effects of the Russian and Asian financial crises on the company were dramatic, and Benefon slipped into the poorest group, group D. Profitability dropped considerably during 1998-99, but solvency remained high. In 2000, profitability was still heavily negative, but less so than during 1999. However, solvency was much lower.

- Doro (No. 2, black arrows) is a Swedish manufacturer of telecom equipment that showed steady improvement in its financial performance. In 1995-96 the company is in Group C1, but increasing profitability (especially in ROE) places the company in Group B, quite near Ericsson, for the rest of the period. In 2000 Doro's profitability was negative, especially in ROE.

- Ericsson (No. 3, yellow arrows), a Swedish major manufacturer of mobile phones and network technology, shows very good performance during 1995-99, remaining in Group B. Profitability, solvency, and liquidity are very good, although not quite as good as for Group A_1. Ericsson also has very high values in receivables turnover. In 2000, Ericsson's performance continued to be strong, with slight increases in nearly all ratios.

- Helsingin Puhelin Yhtiöt (No. 4, blue arrows) is the second largest Finnish service provider, and like Sonera, shows good performance. In 1995-97 the

[22] This high accuracy rate is due to quasi-complete separation of the data (probably, too small a sample size).

company is in Group C1, but steadily improving financial performance brings the company into Group B in 1998. The year 2000 brought problems for HPY, and the values in nearly all ratios dropped. In 2000 HPY changed its name to Elisa Communications.

- NetCom (No. 5, white arrows) is a Swedish service provider that operates in a number of Scandinavian countries. Heavy start-up costs have kept the company in Group D for the entire period. The results for 1998 and 1999 were actually positive, but poor values in ROE have kept the company in Group D. In 2000, NetCom's equity problems were finally solved, resulting in a considerable improvement in ROE and Equity to Capital.

- Nokia (No. 6, red arrows), the leading mobile phone manufacturer, is consistently the best performing Scandinavian telecommunications company. The company was located in Group B during 1995-96, but increased values in all financial ratios pushed the company into Group A_1. Nokia's performance continued to be strong in 2000, with slight improvements in nearly all ratios.

- Sonera (No. 7, turquoise arrows), the largest Finnish service provider, performs well, rising from Group C_1 in 1995 to Group A_1 in 1996-97. In 1998 a drop in profitability forces Sonera into Group A_2. In 1999 profitability increased again, and Sonera moved back into Group A_1. In 2000 Sonera's profitability improves but solvency decreases, indicating increasing indebtedness. In fact, Sonera's Equity-to-Capital has been falling steadily, from 36.22 in 1996 to 18.47 in 2000.

- Tele Denmark (No. 8, pink arrows) remains in the same area of the map, starting out in Group A_1, but dropping into Group C_1 because of decreasing profitability in 1997. However, in 1998 increasing profitability brings the company into Group B, and then in 1999, to Group A_1. In 2000, Tele Denmark's performance continues to improve.

- Telia (No. 10, Sweden) and TeleNor (No. 9, Norway) are interestingly similar in performance, and the companies actually discussed a merger during the course of 1999-2000. However, the deal never materialised because of ownership disagreements. The performance of the two companies is very similar, although Telia shows slightly better profitability and liquidity, while TeleNor shows slightly higher solvency. In 2000 TeleNor's profitability drops, while Telia's profitability increases. Both companies' solvency decreases somewhat, more for TeleNor.

In Table 6-8, the class predictions based on financial data for the year 2000 are illustrated. In this experiment we applied MLR and DT methods.

Table 6-8 Class predictions for Scandinavian telecommunications companies in 2000. The financial ratios are presented in Table 6-4. The techniques used are: MLR and DT.

Company	OM	ROTA	ROE	CR	EC	IC	RT	label	Predicted Cluster	
									MLR	DT
Benefon	-17,03	-30,02	-74,64	1,22	25,10	-12,05	5,93	1_00	D	D
Doro	-2,12	-3,90	-63,70	2,24	13,87	-1,27	6,85	2_00	D	D
Ericsson	11,40	14,63	67,26	1,89	16,81	7,52	3,95	3_00	B	B
HPY	11,99	7,702	10,87	0,53	15,71	3,84	5,81	4_00	C_1	C_1
NetCom	18,77	14,77	66,37	2,75	16,02	4,14	10,10	5_00	B	B
Nokia	19,02	34,98	52,50	1,57	52,23	50,76	5,10	6_00	A_2	A_2
Sonera	84,98	30,11	140,16	0,80	18,47	12,77	1,08	7_00	A_1	B
TeleDenmark	29,08	22,27	49,47	1,19	36,96	6,86	2,65	8_00	A_1	A_1
TeleNor	9,91	5,58	6,77	1,02	24,53	2,00	4,42	9_00	C_1	C_1
Telia	22,21	12,08	22,40	2,38	51,02	41,57	2,52	10_00	A_2	A_2
Average	18,82	10,82	27,75	1,56	27,07	11,61	4,84	11_00	A_1	C_1

(Source: Costea *et al.*, 2002a)

Comparing the two classification techniques in terms of their financial class predictions, we can state that while MLR is more optimistic than DT, the results of the two methods are very similar. There are two cases out of 11 that are classified differently: Sonera (7_00) and Average (11_00). This is because our MLR and DT models emphasise different variables: the DT model relies heavily on ROE, Interest Coverage, Equity-to-Capital, while the first MLR equation (which calculates the probability that class = A_1) has a higher coefficient (greater weight) for Operating Margin, and a lower coefficient for ROE than the third MLR equation (class = B). Also, the value 140 for ROE can be considered an outlier compared to the other ROE values, and consequently, can negatively affect the DT classification model.

Experiment 2

Costea *et al.* (2002b) is an extension of Costea *et al.* (2002a) in the sense that we use the SOM obtained in Costea *et al.* (2002a) to benchmark and make class predictions for the four largest telecom companies (Nokia, Ericsson, Motorola, Sony). As in Costea *et al.* (2002a) we use data from 1995 to 1999 (462 rows), obtaining the same SOM and use data for 2000 and 2001 to make class predictions with MLR and DT (Figure 6-5). Firstly, we illustrate each company's performance during 1995-99 using the SOM trajectories. After this, we use our methodology to predict the classification based on data for 2000-01. We then compare the results achieved using our classification models with how SOM would have classified the new data.

In Figure 6-5 we show the movements for the four largest telecom companies.

- We already showed the financial performance of Nokia (red arrows) and Ericsson (yellow arrows) between 1995 and 1999 in the previous experiment

(Figure 6-4). Based on the results of our class prediction models, Nokia's performance continued to be strong in 2000, despite dropping into Group A_2. In 2001, while slightly dropping in all ratios, Nokia has remained in the same group as in 2000: A_2.

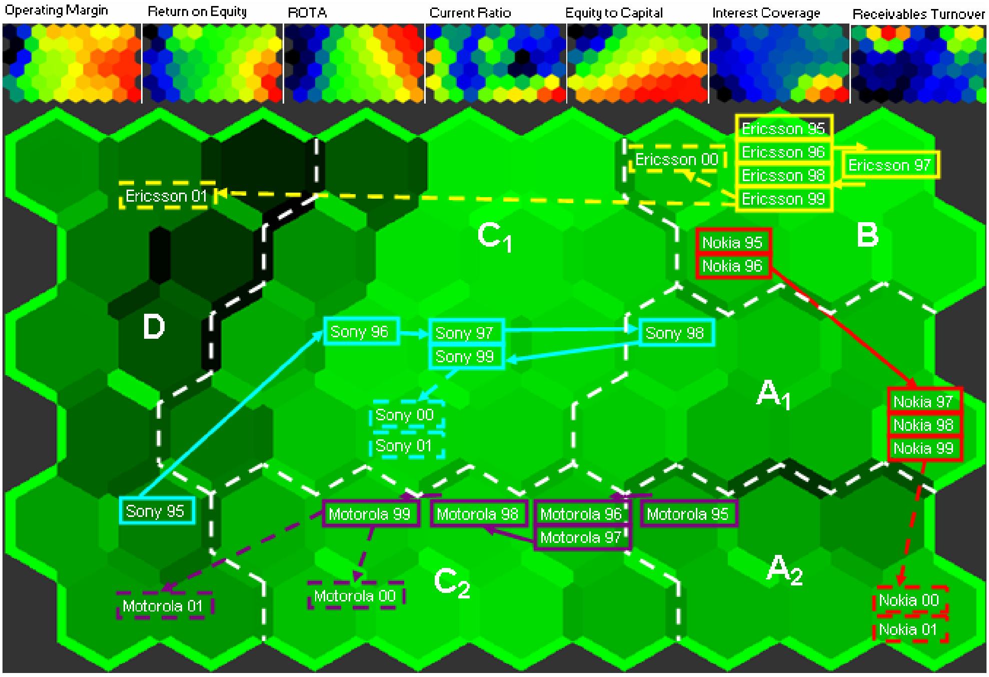


Figure 6-5. Final 9x6 SOM for the telecom data set with identified "real" clusters and feature planes (obtained in experiment 1). The borders of the "real" clusters are identified by the dotted lines. Feature planes for each economic variable are shown at the top of the figure ("warm" colours indicate high values, whereas "cold" colours indicate small values). The financial ratios are presented in Table 6-4. Trajectories (solid lines) for Nokia (red), Ericsson (yellow), Motorola (violet), and Sony (turquoise) between 1995 and 1999, and the predicted classes for 2000 and 2001 by MLR, DT and SOM (dashed lines).

- Based on the results of our class prediction models, in 2000 Ericsson's performance continued to be strong (group B), with slight increases in nearly all ratios. 2001 was a poor year for Ericsson, showing decreases in all important ratios (including negative profitability values). This put Ericsson among the poorest performing companies, group D.

- Motorola (violet arrows), the largest US manufacturer of mobile phones, shows a steady decrease in performance. The SOM model shows that in 1995 the company was situated in Group A_2, but by 1999 it had fallen into the slightly poorer of the middle groups. Motorola was unable to compete with Nokia and Ericsson, even in its own market, and soon experienced difficulties. Examining Motorola's financial statements of reveals, for example, that net income has been decreasing steadily since 1995. Motorola exhibits very good Equity-to-

Capital but, on the other hand, its profitability has decreased during the last four years. Our class prediction models show that in 2000 Motorola experienced a very slight increase in two out of three profitability ratios, as well as improved solvency. Like Ericsson and nearly all other large telecom companies, Motorola experienced a poor year in 2001 in terms of all its financial ratios, dropping to group D.

- Sony (turquoise arrows) improves its performance from Group D in 1995 to Group A_1 in 1998, according to the SOM. However, a reduction in profitability drops the company out of Group A_1 in 1999. In 2000 and 2001, Sony's profitability continues to drop, as do most of the other ratios. Like the other companies, Sony feels the effects of the financial downturn, and the telecommunications industry was one of the hardest hit. Our class prediction models have placed the company in group C_1 for these two years.

Concerning the class predictions both MLR and DT predicted the same classes for all 8 new observations (see Figure 6-5). The class predictions performed by MLR and DT perfectly matched the class predictions produced by SOM[23]. Consequently, and in line with Rudolfer *et al.* (1999), we conclude that both MLR and DT achieved similar results in terms of accuracy rates and class predictions.

Our results correspond with what had really happened: among the four big telecommunications actors, only Nokia remained "untouched" (or slightly affected, moving from group A_1 to A_2) by the financial recession that started in the late 90's while, for the others, this led to drops in all their financial ratios. The classification achieved with our class prediction models corresponds very well with the groups produced by the SOM model, based on the values of the financial ratios for 2000-01.

Experiment 3

Publication 3 extends the telecom experiment in several ways. Firstly, the data between 1995 and 2001 (630 rows) are used for training and available data for 2002 (16 rows) are used to test the classifiers' prediction power. Secondly, the outliers and far-outliers are individually levelled using quartiles for each variable (see Section 5 in Publication 3). Thirdly, we introduce two functions to validate map dimensionality and the quantisation error. Fourthly, the "real" clusters are no longer determined based on the subjective analysis of feature planes, but they are objectively determined by means of Ward's hierarchical-clustering method. Fifthly, besides MLR and DT, we also use ANN to build the classification models. The results from *Publication* 3 are summarised in the subsequent paragraphs.

[23] In both Costea *et al.* (2002) and *Paper* 4 we showed how SOM would have classified the new data by calculating the Euclidean distances between the new observations and the weight vectors. The new data are classified by SOM in the raw cluster that has the smallest difference between its weight vector and that standardised observation.

The procedure used to level the outliers is presented in *Publication* 5, Section 5. After we levelled the outliers, we standardised the data to zero mean and unit standard deviation (normalisation) and trained several SOM maps with different dimensionalities. We tried to validate the map dimensionalities according to empirical measures presented in DeBodt *et al.* (2002). For each map dimensionality (4x4, 5x5, 6x6, 7x7, 8x8, 9x9) we used 100 bootstrap datasets to train the SOM. We expected the variation coefficients[24] of the quantisation error vectors to increase with the map dimensionality. However, we obtained very small variation coefficients (approx. 2%) for all architectures, which did not allow us to reject any architecture. Therefore, a final 9x7 SOM map was chosen based on the ease-of-readability criterion. For this SOM architecture we tested three quantisation errors: one obtained when all the data are used for training and testing the SOM ("100-100" case), another when 90 per cent of data are used for both training and testing ("90-90" case), and the other when 90% is used for training and the remaining 10 per cent for testing ("90-10" case). Again, for each training-testing dataset combination we extracted 100 bootstrap datasets from the original data and obtained a quantisation error vector for each combination. Then, we used *t*-tests to compare the means of the three vectors. The *t* statistic is obtained by dividing the mean difference (of the two vectors) by its standard error. The significance of the *t* statistic (p-values < 0.05) tells us that the difference in quantisation error is not due to chance variation, and can be attributed to the way we select the training and testing sets. Even though we found some differences between the quantisation error vectors the confidence in the results was rather poor (p-value for "100-100" – "90-90" pair was 0.051). Finally, we followed the "100-100" case using the entire dataset to train and test the 9x7 SOM. Even if in this particular case they were not of much help, these empirical validation procedures allow us to choose more rigorously the SOM parameters. Finally, the SOM parameters chosen were: $X = 9$, $Y = 7$, $rlen_1 = 3150$, $\alpha_1(0) = 0.5$, $N_1(0) = 10$, $rlen_2 = 31500$, $\alpha_2(0) = 0.05$, $N_2(0) = 1$.

The 63 "raw" clusters are further grouped into seven "real" clusters using Ward's hierarchical clustering method. Viscovery® SOMine software was used to form the "real" clusters using the Ward method.

After we identified the seven clusters and attached the class labels to each data row, we applied MLR, DT and ANN to build the classification models. For each technique we performed the methodological steps from the beginning of Section 5.2 as we did in the previous experiments. In Table 6-9 we present the training and testing accuracy rates for all three techniques.

MLR, DT and ANN achieved similar results when all the data were used in training (83.3%; 90.3% and 84.13%), with the decision tree achieving the best result. Also, the accuracy rates were validated against the test accuracy rates for all

[24] A variation coefficient of a vector *v* is the ratio between the standard deviation and the mean of *v* and measures the relative dispersion of vector *v*.

three models (the differences were small). In contrast with previous experiments, we here use cross-validation to test the MLR training accuracy rate. Another addition as compared to the previous models is cross-validation of the training accuracy rates for DT and ANN for both splitting datasets. Even though it provides the smallest accuracy rates, MLR seems to be the most robust model of the three since in this case we have the smallest difference between the training and testing accuracy rates. ANN is the most unreliable model from the same perspective.

Table 6-9. Accuracy rate validations for the classification models (telecom – exp. 3). The techniques used are shown in the first column: MLR, DT, and ANN. The validation is done according to step 5 of the methodology presented at the beginning of Section 5.2.

Technique		Main dataset	Part1 (split=0)	Part2 (split=1)
MLR	Learning Sample	83.3%	86%	87%
	Test Sample	no test sample	82.5%	82.6%
	Cross-validation	84.1%	no cross-validation	no cross-validation
DT	Learning Sample	90.3%	86%	86.7%
	Test Sample	no test sample	78.1%	74.6%
	Cross-validation	81.3%	75.8%	76.2%
ANN	Learning Sample	84.13%	85.4%	74.29%
	Test Sample	no test sample	53.97%	63.18%
	Cross-validation	84.29%	53.55%	63.55%

Next, we tested the prediction power of our models using 16 observations of Asian companies from 2002 and compared the results with SOM predictions. MLR and DT performed very similarly (12 out of 16 were classified in the same clusters), MLR having three and DT four misclassified cases when compared to SOM classification. Therefore, we can again conclude that these methods perform quite similarly. ANN was more optimistic than the other methods – nearly all companies were placed in higher classes. This might be due again to the ANN architecture chosen with one single neuron in the output layer. In *Publication* 5 we thoroughly investigate the ANN classification models and apply them to the telecom dataset.

In *Publication* 4 we use fuzzy logic to group the telecommunications companies by financial performance. We apply both normal FCM and Weighting FCM to group the telecom companies and compare the results with those produced by SOM. We used the same dataset as in *Publication* 3 (630 rows) with levelled outliers and far-outliers. The parameters for the FCM algorithms were: the weighting exponent m = 1.5, as this was the value used when we calculated the linguistic matrix, and c = 7 to make the results comparable with SOM clustering. We used a Matlab platform to implement our FCM-related algorithm. Firstly, we determined the linguistic matrix that contains a linguistic term (VL, L, A, H, or VH) for each observation and for each ratio. Then, we selected the "certain" observations and, using the linguistic terms of these "certain" observations (Step 1 of the Weighting FCM algorithm), we characterised the clusters as follows: one cluster is characterised by one linguistic term (for one ratio) if that linguistic term has at least 40 per cent occurrences in that cluster (and for that ratio). In Table 6-10 we present the characterisation of the clusters based on the linguistic variables.

Table 6-10. *Objective* characterisation of the clusters based on the linguistic variables. Each cluster is characterised objectively as follows: we assign a linguistic term to one variable in a cluster if that linguistic term has at least 40% occurrences. The financial ratios are presented in Table 6-4.

	OM	ROTA	ROE	CR	EC	IC	RT	Order
Cluster 1	VL	VL	VL&L	-	A&H	VL&L	-	Bad
Cluster 2	A	A	A&H	-	A	A	-	Average
Cluster 3	VL&L	VL	VL	-	VL&L	L	-	Worst
Cluster 4	H	H	VH	VL	A	A	A	Good
Cluster 5	A	A	A&H	H	H	VH	-	Good
Cluster 6	L	L	A	L	L	L	-	Bad
Cluster 7	VH	VH	H	VH	VH	VH	-	Best

(Source: *Publication* 4)

There cannot be more than two linguistic terms to satisfy the above criterion. For some clusters, some variables did not have any discriminatory power (e.g. RT for all clusters except cluster 4). With the help of linguistic variables we can automatically characterise each cluster in an objective way as opposed to the characterisation based on the feature planes, which is more subjective. Also, by comparing the clusters based on their linguistic characteristics we can label them as being good, bad, worst, etc.

After step 1 of the Weighting FCM algorithm, we obtained 110 "uncertain" observations, while the remaining 520 certain observations were distributed between different clusters as shown in Table 6-11 (column "Step 1"):

Table 6-11.Distribution of the telecom observations in the clusters. The second column shows the distribution of the "certain" observations allocated by Weighting FCM in Step 1. Next two columns show the distribution of all observations allocated by FCM, and Weighting FCM respectively.

	Step 1	Normal FCM	Weighting FCM
Cluster 1	46	57	61
Cluster 2	135	167	170
Cluster 3	52	61	60
Cluster 4	64	80	79
Cluster 5	56	77	71
Cluster 6	120	137	134
Cluster 7	47	51	53
Total	520	630	628

After we applied the Weighting FCM to allocate the 110 "uncertain" observations, 19 were allocated differently by the Weighting FCM as compared with the normal FCM. To compare the two algorithms we calculated the coincidences that each of the 19 observations had in terms of number of linguistic terms with the cluster characterisation in Table 6-10. For example, in Table 6-12 we present the qualitative form of the observation 158:

Table 6-12. Qualitative form of observation 158. Columns 2-8 show the linguistic terms associated with each financial ratio (see Table 6-4). Columns 9 and 11 show how Normal FCM and Weighting FCM, respectively positioned the observation. Columns 10 and 12 represent the number of coincidences (in linguistic terms) between the observation and the clusters' characterisation for Normal FCM and Weighting FCM, respectively.

Obs.	OM	ROTA	ROE	CR	EC	IC	RT	Normal FCM	X	Weighting FCM	Y	SOM
158	L	L	L	VL	VH	L	H	5 – Good	1	1 – Bad	4	Bad

The Normal FCM classifies this observation in cluster 5, while the Weighting FCM classifies it in cluster 1. The number of coincidences in linguistic terms between observation 158 and cluster 5 is one, while the number of coincidences with cluster 1 is four (see Table 6-10). Therefore we conclude that the Weighting FCM better allocated the "uncertain" observation 158. We proceed likewise for all 19 observations. Overall, our implementation overcame the normal FCM. We, also, compared the Weighting FCM with SOM in terms of pattern allocation. We used the SOM results from *Publication* 3, which uses the same dataset and checked how SOM allocated the 19 uncertain observations. As for the normal FCM, so also the Weighting FCM outperformed SOM in terms of "uncertain" pattern allocation. There were two observations that were not allocated at all by the Weighting FCM. The reason was that these observations had their two highest membership degree values very close to each other. In *Publication* 4 we treat these observations separately explaining why our algorithm did not allocate them.

Experiment 4

Another experiment related to financial performance classifications in the telecommunication sector is presented in *Publication* 5. In this publication we, again, build *hybrid* classification models as following: we use FCM to partition the input space and build the class variable and, then, we apply ANNs to construct the classification models. We try to validate the following hypotheses:

H1. The training mechanism used to refine the solution obtained when determining the ANN architecture will have an influence on the classification performance of ANNs. The GA-based ANN will outperform the RT-based ANN both in training and testing in the refining process. In our experiments up to now, when building ANN-based classification models, we used the first neural approach, the approach where we use the ANN obtained when we determine the ANN architecture as our ANN classification model. This was the case in *Publication* 3 and Costea (2003). In *Publication* 5, we go one step further and improve the accuracy of the ANN obtained when determining the ANN architecture with two training mechanisms: one is a gradient descent-like technique improved by a retraining procedure (RT), and the other is the genetic algorithm (GA). In our previous experiments (*Publication* 3 and Costea, 2003) ANN classification models were outperformed by both MLR and DT models in terms of training and testing accuracy rates. We stated there that one reason might be the way we codify the class variable: using one output neuron. In *Publication* 5 we used as many neurons as the number of classes in the output layer and the results improved substantially.

H2. Data pre-processing will have an influence on both RT and GA-based ANN training and testing performances. Three pre-processing approaches are undertaken: “no pre-processing”, “division by absolute maximum values” and “normalisation”.

H3. Data distribution will have an influence on both RT and GA-based ANN training and testing performances. Five different distributions are examined: the real data, uniform, normal, logistic and Laplace distributions.

H4. The crossover operator will have an influence on GA-based ANN training and testing performances. We test four types of crossover operators: arithmetic, one-point, multi-point and uniform crossover.

H5. The stage at which we generate the effective training and validation sets will have an influence on RT-based ANN training and testing performances. Three types of retraining procedures are tested as explained in Section 5.2.3.

The main hypothesis of *Publication* 5 is formulated as follows:
H6. All binary and ternary combinations of the above three factors (training mechanism, pre-processing method and data distribution) will have an influence on both RT and GA-based ANN training and testing performances.

The experiments undertaken in *Publication* 5 differ in two respects: the hypothesis they try to validate, and the type of statistical test used (parametric or non-parametric). The first three experiments use non-parametric tests (Siegel & Castellan, 1998) to validate hypotheses *H1*, *H2* and *H3*. All three hypotheses are strongly supported. In experiment I the starting solution had relatively low accuracy rates (80-90%) and GA clearly outperformed the RT mechanism.

Experiment IV uses non-parametric tests and tries to validate *H4* and *H5*. *H4* claims that the choice of the crossover operator has an influence on GA performance. As it was reported (Yao, 1999; Pendharkar & Rodger, 2004) we find no evidence to support *H4*, all crossover operators performing similarly. The same result was obtained for *H5*, all three retraining mechanisms achieving similar results.

In experiment V we tested ***H6*** performing a parametric test (a 3-way ANOVA – Garson, 2005), and again, all individual factors have a statistically significant influence on both ANN training and testing performances. At the same time, the influence of any combination of two of the three factors was found to be statistically significant. The results of comparing pairs for each factor validate once again the first three hypotheses. However, we have one amendment in the case of *H1*: in experiment V the starting solution had relatively high accuracy rates (95-

98%) which resulted in very small differences between GA and RT-based ANN training/testing accuracy rates.

In our experiments GA was much slower than RT. Further research must be conducted to properly tune the GA parameters to make it more efficient.

6.3 Predicting Process Variables

The last experiment presented in this dissertation evaluates the use of ANNs in performing the DM regression task.

6.3.1 Glass Manufacturing Process at Schott

The aim of this experiment is to construct an ANN model that would help in monitoring/controlling the glass-melting process at Schott, a glass manufacturer from Germany. We would like to have a measure of the glass quality in real time. In Figure 6-6 we present a typical glass flow scheme for a Schott melting tank.

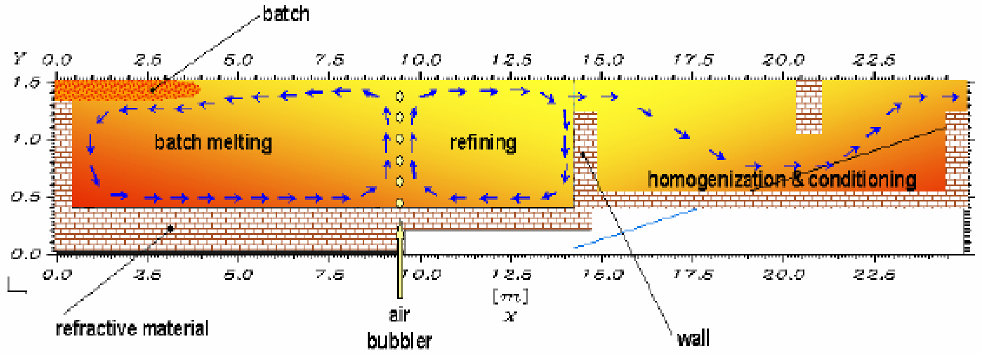


Figure 6-6. Scheme of Glass Flow in a Melting Tank
(Source: Lankers & Strackeljan, 2004)

There are different parts of the tank where different operations take place. The left part of the tank is the melting area where raw materials batch-mixed with cullet are heated by burners from above. The temperature in the melting tank is around 1500 °C, which makes exact measurement of the process variables and visual inspection extremely difficult (Lankers & Strackeljan, 2004). After complete melting, the molten glass contains small seeds containing CO_2 or air. In the fining (degassing) phase, these seeds are removed by dissolving fining gases (sodium sulphate, antimony oxide, arsenic oxide, etc.) into the molten glass. The fining gas diffuses into the existing seeds making it easier to bring the bubbles to the surface of the glass melt (Beerkens, 2001). After the fining process, the melt has a low concentration of dissolved gases and only very small bubbles filled with the fining gas. The refining phase is a controlled cooling phase where the melt is cooled from the fining temperatures (1400-1600 °C) to temperatures of typically 1200-1300 °C. For most gases that can dissolve chemically, such as CO_2, O_2, SO_2, solubility increases at decreasing temperatures (Beerkens, 2001). In other words, cooling makes the gases in the bubbles dissolve in the melt. Then, the melt is forced to the homogenisation and conditioning area of the tank. In this area the melt is homogenised in the sense that through diffusion process the concentration of aluminium oxide or zirconium oxide in the glass is reduced (Beerkens, 2001).

The glass quality depends on many input variables, some of which are measurable (e.g. energy entries, raw material components, environmental temperature) and have been included in the model. The outputs are temperatures in different parts of the melting tank. The correct adjustment of temperature behaviour directly affects quality of the final product (Lankers & Strackeljan, 2004). The Schott Company provided the dataset to EUNITE[25], which organised a competition in 2003. All data were rescaled and made anonymous for the sake of confidentiality. The dataset consists of 9408 observations (one observation every 15 minutes over a period of 14 weeks). There are 29 inputs and five outputs (Figure 4-1). The ANN forecasting model (see Section 5.3.1) is used to predict the relevant output variables (temperatures) of the glass-manufacturing process. The ANN forecasting model was developed iteratively in Nastac & Costea (2004a) and *Publication* 6.

The main improvement in *Publication* 6 is the introduction of the delay vector *Vect_Out* (see Figure 5-5). In Nastac & Costea (2004a) one output at moment *t* could only depend on the inputs at different previous moments and on the same output or other outputs at moment *t*-1 (Figure 6-7). The introduction of *Vect_Out* allows one output to be influenced by other outputs at different previous time steps. For example, a *Vect_Out* = [0, 2, 6, 12, 20] would cause each output $output_i(t)$ to be influenced by the following outputs: $output_i(t\text{-}1)$, $output_i(t\text{-}3)$, $output_i(t\text{-}7)$, $output_i(t\text{-}13)$, and $output_i(t\text{-}21)$, $i = 1,...,5$.

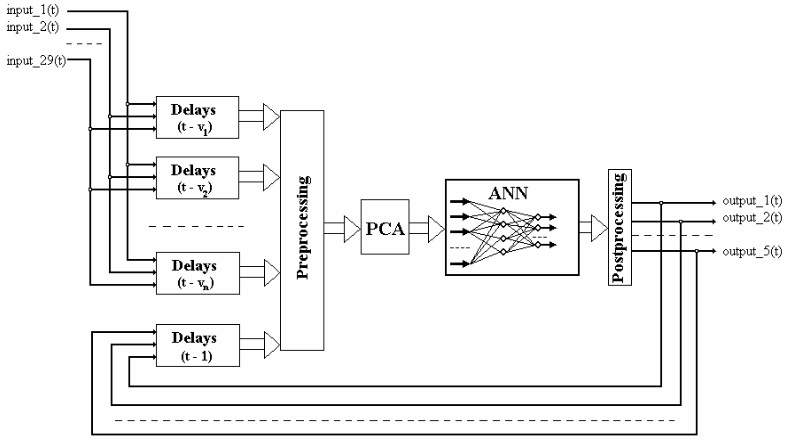


Figure 6-7. The ANN forecasting model without input selection and output delay vector. Principal Component Analysis (PCA) is used to preprocess the ANN input. (Source: Nastac & Costea, 2004a)

[25] The EUropean Network on Intelligent TEchnologies (EUNITE) started its activity on January 1, 2001 to increase the use of intelligent technologies for Smart Adaptive Systems.

In *Publication* 6 we build three new models based on different *Vect_Out* delay vectors, and/or different ANN architectures (one or two hidden layers) and compare the performance of these models with the performance of the models presented in Nastac & Costea (2004a). Another improvement in *Publication* 6 was the selection of the relevant input variables according to the glass manufacturer's suggestion: $input_1$, $input_2$, $input_4$, $input_{10}$, $input_{19}$, $input_{20}$, $input_{23}$, $input_{29}$, as well as two derived inputs ($E_1 = input_1 + input_2 + ... + input_{10}$) and ($E_2 = input_{20} + input_{21} + ... + input_{28}$). The sums represent the heating energies (*E*1 and *E*2, respectively) of the melting furnace. In Nastac & Costea (2004a) the data for the first 12 weeks (8064 observations) are used to train the ANN. Then, the model is used to predict the outputs for the next two weeks (1344 observations). In *Publication* 6 data from the first 13 weeks are used for training and the temperatures are predicted for week 14.

In Table 6-13 we reproduce the different ANN forecasting models along with their parameters.

Table 6-13. Four selected models along with their parameters. The columns identify the models (A, B, C, D). One row shows the parameter choice for each of the four models.

Model	A	B	C	D
Inputs	All	All	All	1, 2, 4, 10, 19, 20, 23, 29, E1, E2
Prediction	weeks 13&14	week 14	week 14	week 14
Vect_In	[10 20 35 55 80 120 185 290]	[5 10 20 30 45 65 95 145 210]	[1 3 6 10 15 22 31 45]	[1 3 6 10 15 22 31 45]
Vect_Out	-	[0 4 10 20]	[0 2 6 12 20]	[0 2 6 12 20]
PCA - *transMat*	132 × 237	139 × 281	113 × 257	48 × 105
ANN	132 : 35 : 5	139 : 35 : 5	113 : 45 : 5	48 : 30 : 19 : 5
Model no.	2	3	2	2
ERR_A	0.3602	0.2680	0.2151	0.2514
ERR_T	0.4095	0.2877	**0.2702**	0.2832

(Source: *Publication* 6)

Both the determination of the ANN architecture and the retraining techniques are the same as in the classification case (Section 5.2.3). For each combination of the inputs and delayed vectors (Selected inputs – *Vect_In* – *Vect_Out*) we had three models: the first model (1) is the trained network obtained when we determine the ANN architecture, the second (2) consists of applying RT1 retraining mechanism to improve the first solution, and to obtain the third model (3) we apply RT3 to improve the result of RT1.

In total we had 69 models (23x3). Twenty-three models used different delay vectors and/or different inputs. Our "selection tool" was ERR_A (see Section 5.3.1) when we did not have the output test data (before the Competition ended) and ERR_T when we had the test data (after the Competition ended). In Table 6-13 we selected the best four models in terms of both ERR_A and ERR_T. The second row of the Table 6-13 indicates the inputs used in the model. The third row indicates the prediction horizon. In the following two rows we present the delay vectors used. The sixth row indicates the dimension of the PCA transformation matrix. For example, model A has a transformation matrix 132x237, which means that PCA has been able to reduce the input space from 237 inputs[26] to 132 uncorrelated inputs. For PCA we set the minimum fraction of the total variation in the dataset (*min_frac* = 0.001). Model A was that submitted to EUNITE and reported in detail in Nastac & Costea (2004a). In the seventh row we have the dimensions of the ANN architectures. The “Model no.” row indicates which of the three models (1, 2, and 3) achieved the best result. The final two rows include the training errors (ERR_A) and test errors (ERR_T). The best performance in terms of both ERR_A and ERR_T is achieved by model C with the RT1 as the improving mechanism. For each model the accuracy of the outputs decreases in time, as each new forecasting step subsumes the errors of the previous predictions. Moreover, there might be new patterns in the test interval (week 14) that were not taken into consideration during the training process. However, in case of model C we observed that the forecasting process was the most stable.

The outputs of the model submitted to the EUNITE Competition 2003 (model A) are presented in the following figures. Figure 6-8 shows the output predictions compared with the real outputs for the training data. As can be seen, the thick lines (output predictions) cover very well the thin lines (real outputs).

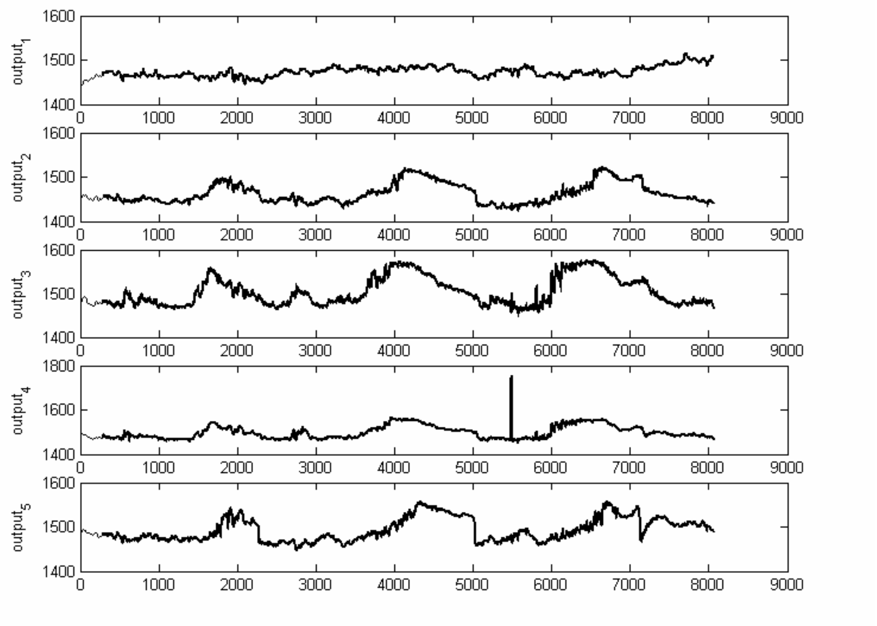


Figure 6-8. Process outputs for training (observations 1-8064): the real outputs (thin lines) and the predicted outputs (thick lines)
(Source: Nastac & Costea, 2004a)

[26] 237 = 29 (original inputs) x 8 (the size of *Vect_In*) + 5 (original outputs) x 1 (here, size of *Vect_Out* is 1 since only the outputs from the previous time step *t*-1 are used)

Figure 6-9 shows the simulated outputs compared with the real outputs for the testing data (weeks 13 and 14). We took 3rd prize in the competition and our ANN model was 5th in terms of the test ERR prediction error – ERR_T (Figure 6-10 (a)). However, compared with the first four, our solution was the second best in terms of the variance of all five output errors: only solution 3 had a smaller variance (Figure 6-10 (b)).

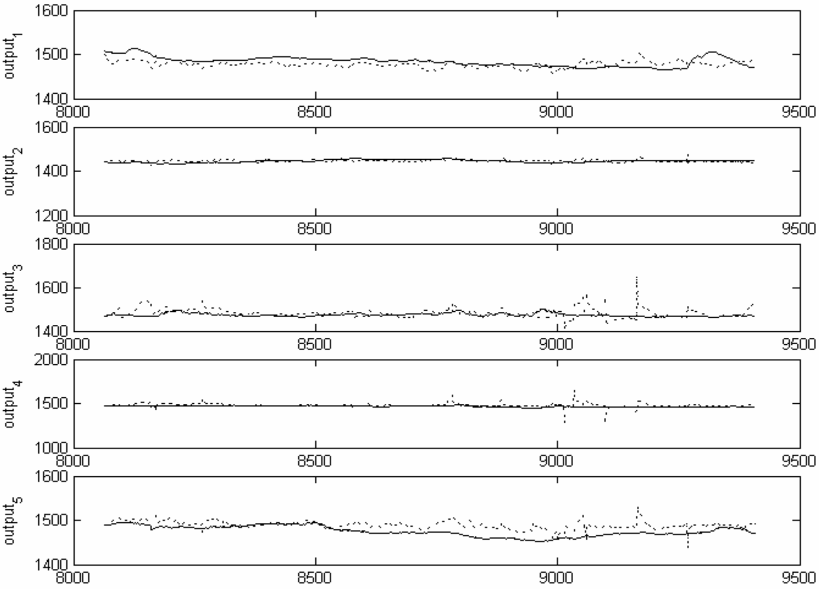


Figure 6-9. Process outputs for testing (observations 8065-9408): the real outputs (solid lines) and predicted outputs (dotted lines)
(Source: Nastac & Costea, 2004a)

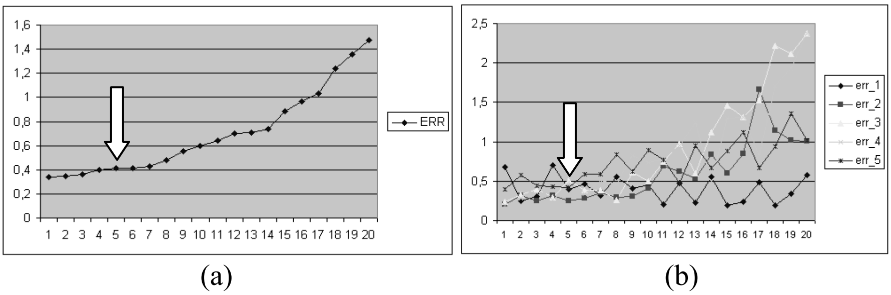


Figure 6-10. (a) ERR_T for all 20 solutions submitted (our model is no. 5 with ERR_T = 0.4095) (b) Errors of all five outputs for the 20 solutions.
(Source: EUNITE Competition, 2003)

The winning solution of the competition (the solution which best corresponded to the ERR_T criterion) was unusable, even though it had the smallest ERR_T (0.3418). For each output, the predicted values were equal to the last given value for the output in question, i.e. $O_i(t) = O_i(8064)$, $t = 8065,\ldots,9408$ and $i = 1,\ldots,5$. In contrast with the other solutions, ours had two advantages from the beneficiary's point of view: ANNs were considered to be the most obvious technique that could be applied to this kind of multi-input multi-output modelling problem (our solution being the only one among the winning solutions that employed this technique) and the fact that our solution was second best in terms of the overall variance of the predicted outputs.

We are convinced that closer scrutiny of the choices of inputs and ANN parameters can further improve the ANN performance. Further research is needed to implement an adaptive ANN system that will be periodically updated as data become available.

Our forecasting model is easily adaptable to any kind of regression task. The model is parameterised in the sense that the user can specify the input and output datasets and the delayed vectors. Next, the model determines the ANN architecture using the empirical procedure from Section 5.2.3. Then, PCA reduces and uncorrelates the input space. Two retraining procedures are then used to refine the solution. During our experiments we used SCG as the training algorithm. Depending on the problem this can be changed to include faster or more efficient algorithms. As we showed in Section 5.3.1 other time series (e.g. Koskivaara, 2004b) present similar behaviour with the one used in our experiment. The delayed vectors permit the user to specify exactly how one output is influenced by the inputs, by the same output or by other outputs at different previous time steps, thus extending the applicability of the ANN forecasting model.

Chapter 7 Conclusions

In this dissertation we explored and compared different computational intelligence (CI) methods such as decision-tree induction (DT), self-organising maps (SOMs) and multilayer perceptrons (MLPs), genetic algorithms (GAs) and Fuzzy C-Means (FCM) to address three different business problems: benchmarking countries' economic performance, benchmarking companies' financial performance and predicting process control variables. We addressed these problems by transforming them in a combination of quantitative data-mining tasks. The corresponding data-mining tasks were: clustering and classification (for the first two problems) and regression (for the third one). We showed how CI methods can support business players in addressing the above business problems. We contributed to the research on using CI methods in performing the DM tasks by both exploring and comparing different CI methods and, also, by solving some technical problems associated with the implementation of each method. Statistical methods (e.g. C-Means, MLR) were used as benchmarking techniques for the CI methods.

In our study we use a pluralistic research strategy emphasising constructivism (Iivari *et al.*, 1998; Kasanen *et al.*, 1993). All seven guidelines for doing effective constructive research (Hevner *et al.*, 2004) are satisfied as explained in Section 2.2.

In accordance with Hevner *et al.* (2004) we had two main research questions: one for management-oriented audiences and the other for the technology-oriented ones.

1. *How could CI methods be used to construct business models with which business problems such as benchmarking countries'/companies' economic/financial performance and predicting the control variables of internal processes could be addressed?*

2. *What technical problems need to be considered when constructing these business models?*

Next, we present the managerial implications related to the solutions to the business problems and the main contributions to research in using CI methods for performing DM tasks. Finally, we outline the limitations of our study and our future research directions.

7.1 Managerial Implications

We demonstrated how our models can help to solve business-related problems (first main research question) by implementing them using a number of experiments.

The first experiment concerned assessing the economic performance of Central-Eastern European countries. We revealed groups with similar economic performance and showed how countries' economic performance evolved over time. Feature planes were used to characterise the economic performance of the groups. In addition, by studying individually each feature plane, we were able to characterise the countries on the basis of each economic variable: currency value, refinancing rate, industrial outcome, unemployment rate, exports, imports, and the foreign trade. Trajectories were used to trace the countries' movements over time. For example, Figure 6-1 shows the trajectories for Romania and Poland between 1996 and 2000. Overall, Romania was unstable with respect to all the economic variables. Poland, on the other hand, had a stable economic performance, which led to its acceptance as a member of the EU in 2004. Figure 6-2 shows that Ukraine had steadily progressed between 1993 and 2000 with respect to its foreign trade balance. Different investors or international corporations who want to invest or open new subsidiaries in Eastern Europe and would like to have an overall picture about what the economic situation is in this part of the world can benefit from this kind of analysis. Other beneficiaries might be the countries involved in the analysis, i.e. the countries that are not yet EU members (e.g. Romania, Russia, Ukraine) and would like to learn from the best performers. The conclusion from the experiment is that our models can support business players in their investment decisions.

In the second and third experiment we benchmarked and predicted the financial performance of international companies from two major sectors: the pulp-and-paper and telecommunications sectors. With our benchmarking models we grouped the companies in terms of profitability, liquidity, solvency and efficiency. Each group was *automatically* characterised with the use of linguistic variables. With the financial classification models we entered new observations into the already constructed groups (clusters) without having to re-run the experiments. In our experiments we showed comparatively how some of the best companies performed financially over the years. In the pulp-and-paper experiment we benchmarked the best three Finnish companies (Figure 6-3), UPM Kymmene being the best performer. In the telecom experiment we benchmarked the Scandinavian telecom companies (Figure 6-4) and the four largest telecom companies (Figure 6-5) with Nokia achieving the best result. All stakeholders (decision-makers, creditors, investors) can benefit from this type of analysis. Decision-makers in the companies involved in the analysis would understand the causes of their business problems by learning from others' achievements/mistakes. Creditors would obtain a general picture about the financial situation of different companies, which would reduce their credit risk. Using our models, investors would be able to weigh the different investment opportunities by performing the comparisons themselves.

The last experiment involved constructing good models for controlling the glass-manufacturing process at Schott, a German-based company. We predicted the melting tank temperatures based on different process inputs. We used ANN to perform the DM regression task associated with this business problem. The validation of our implementation resulted from the comments made by the final beneficiary of the model, the Schott Company, through its senior researcher Dr.

Katharina Lankers: "*We could not actually apply the concrete values from the prediction for plant control, perhaps because some decisive parameters had not yet been recorded, but the proposed model approaches are of great value to us"*.

The high parameterisation and flexibility of our ANN forecasting model allows the potential users to apply it with small modifications to similar process control tasks.

7.2 Contributions to Research

We divide our contributions in using the CI methods into three parts according to which DM task is performed. Here, we address the second main research question. We answer the second main research question by addressing some technical problems of the CI methods.

7.2.1 Contributions to CI Methods for DM Clustering Task

We explore and compare different statistical and CI methods for the clustering task (SOM and C-Means in Costea *et al.*, 2001; SOM, FCM, Weighting FCM in *Publication* 4). In *Publication* 3 we validate the SOM map dimensionality and the quantisation error according to De Bodt *et al*. (2002). We validated statistically the map dimensionality by building 100 bootstrap samples for each map dimensionality. One map dimensionality is validated if the variation coefficient of the quantisation error vectors increases with the increase in map dimensionality. The quantisation error is statistically validated if there is no difference in the quantisation error vectors for two different training samples. We added two functions to our Visual C++ SOM implementation for map dimensionality and quantisation error validations.

Another contribution to CI methods for DM clustering task is the introduction of Weighting FCM (*Publication* 4), which proved to better allocate the "uncertain" observations compared with SOM and normal FCM. Also, with the introduction of linguistic variables we can now *automatically* characterise each cluster in terms of performance dimensions.

The other contribution is the two-step clustering of SOM introduced in *Publication* 2, and used throughout the study. The two-step clustering consists of building a larger SOM map (with many "raw" clusters), and, then, by using the visualisation capabilities of the SOM, re-group the similar "raw" clusters into "real" clusters. In *Publication* 3 we substitute the subjective way of determining the "real" clusters using the SOM feature planes by objectively constructing the "real" clusters using the Ward's hierarchical clustering.

The use of clustering results to further develop the classification models (which was suggested by different authors: Witten & Franck, 2000; Costa, 2000; De Andres, 2001) can be seen as an implementation of contributions from the literature. However, even though this combination (clustering + classification) was suggested elsewhere as well, few studies have implemented it efficiently.

7.2.2 Contributions to CI Methods for the DM Classification Task

Our first contribution to the research in using CI methods for the DM classification task consists of exploring and comparing different *hybrid* classification models. The hybrid models are based on a two-phase methodology: the first phase applies a clustering technique and builds the class variable and in the second phase we model the relationship between the class variable and the explanatory variables by building hybrid classifiers. In *Publication* 2 and Costea *et al*. (2002a, b) two classification techniques are compared (MLR and DT) and the best one is chosen in terms of accuracy rate. In *Publication* 2 we analyse the Central-Eastern European countries in terms of economic performance and companies from the pulp-and-paper sector with respect to their financial performance. In Costea *et al*. (2002a, b) another industry is analysed, namely the telecommunications sector, by employing SOM for the clustering task and MLR and DT for the classification task. In *Publication* 3 we introduce ANN to classify telecom companies and compare the neural approach with MLR and DT in terms of classification accuracy. We find, for each particular sector, the *most adequate hybrid classification model*. Costea (2003) compares MLR, DT and ANN in classifying Central-Eastern European countries based on their economic performance. In Costea (2003) we, also, test different ANN training algorithms and different ANN architectures.

Another contribution is the introduction of a standard method to compare the different approaches to the classification task. This is done by presenting the methodological steps at the beginning of Section 5.2. In section 5.2 we also present (with our research publications' support) practical ways of tuning the parameters of each classification approach. In line with De Andres (2001) we state that the choice of the hybrid system is context and problem-dependent.

Publication 5 brings several research contributions: investigation of three different factors (pre-processing method, data distribution and training mechanism) influences the classification performance of ANNs, introduction of an empirical procedure for determining the ANN architecture, and finding the best crossover operator in terms of GA-based ANN classification performance.

7.2.3 Contributions to CI Methods for DM Regression Task

Our contribution to ANNs for regression tasks reside in presenting the steps necessary in designing the ANN (tuning of ANN parameters) as a forecasting tool, in introducing an empirical procedure to determine the ANN architecture. We also introduce an alternative way of training an ANN based on its past training experience and weights reduction and present different ways of applying it in the context of forecasting models. The retraining technique significantly improved the achieved result in terms of prediction error. Our contributions are presented in Nastac & Costea (2004a) and *Publication* 6, which also include an empirical procedure for validating the ANN prediction performance when the test outcomes are not known (ERR_A). The independence between the ANN architecture, retraining procedure and training algorithm confers upon our ANN forecasting tool great flexibility, which allows the user to set different parameters in the context of other business problems.

7.3 Limitations and Future Directions for Research

The first limitation is the lack of external validity of our models. We validated our models using different internal validity measures: quantisation error, accuracy rate, mean squared error and class prediction performance. External validity would require measuring the satisfaction of potential users (e.g. investors, creditors, etc) with the proposed models. An attempt in this direction was presented in Eklund (2004, pp. 91-106), where the author used a structured questionnaire to measure the users' satisfaction with a benchmarking SOM model. However, according to Hevner *et al.*'s (2004) guidelines for doing effective constructive research, the validation criteria used in this dissertation are accepted in the research community.

Another limitation of the study is the restrictive number of CI methods employed to address the business problems. Moreover, we did not use all the methods described in Chapter 5 in all the experiments (see Table 6-1). In this study we were more interested in providing guidelines of how one can apply and compare CI methods in addressing certain business problems. We did not attempt to address each business problem with all the methods available. With respect to the above concern about the restrictive number of CI methods used, we are based on Hevner *et al.*'s (2004) sixth guideline for designing science research, which says that we should seek a *satisfactory* number of solutions to a specific problem and that it is not feasible to test all possible solutions (methods).

A further limitation in our research comes from the great amount of time needed to collect the information for our experiments. This is not unexpected: Romeu (2001) claims that up to 60 per cent of total project time is dedicated to data preparation. The Data Collection Agent of our Knowledge-Building System (*Publication* 1) is supposed to do just that. However, because of the lack of standard financial reporting on the Internet, the Data Collection Agent might not be able to perform the task. The development of different standard reporting languages such as XBRL (eXtensible Business Reporting Language) would permit effective implementation of the collection agents.

With the growing amount of information about competitors and the increasing demand for better and more efficient products and services, the need for intelligent tools to assess competitors' performance and optimise internal production processes is likely to increase in the future. As future directions of our studies we will concentrate on making our models more user-friendly. Consequently, new interfaces have to be constructed that will allow potential users to set the relevant parameters of the models. Regarding the CI methods used in our experiments, we will focus in the immediate future on those for which we obtained the best results and improve them yet further.

References

1. Alander JT. 1995. Indexed bibliography of genetic algorithms and neural networks. *Report* **94-1-NN**, University of Vaasa, Department of Information Technology and Production Economics, 1995. [Available at: ftp://ftp.uwasa.fi/cs/report94-1/gaNNbib.ps.Z, *Key*: gaNNbib] (Accessed on: 26.07.2005).
2. Alcaraz-Garcia AF, Costea A. 2004a. On Assessing Companies' Financial Performances by the Means of Weighting FCM Clustering. *Proceedings of MCO 2004*, Le Thi Hoai An, Pham Dinh Tao (eds.), published by Hermes Science, Metz, France, July 1-3, 2004, pp. 265-272.
3. Alcaraz-Garcia AF, Costea A. 2004b. A Weighting FCM Algorithm for Clusterization of Companies as to their Financial Performances. *Proceedings of the IEEE 4th International Conference on Intelligent Systems Design and Applications (ISDA 2004)*, Rudas I. (ed.), CD-ROM Edition, Budapest, Hungary, August 26-28, 2004, Track: Intelligent Business, pp. 589-594 (***Publication 4***).
4. Alhoniemi E, Hollmen J, Simula O, Vesanto J. 1999. Process Monitoring and Modeling Using the Self-Organizing Map. *Integrated Computer-Aided Engineering* **6**(1): 3-14.
5. Alhoniemi E. 2000. Analysis of Pulping Data Using the Self-Organizing Map. *Tappi Journal* **83**(7): 66. [Available at: http://lib.hut.fi/Diss/2002/isbn951226093X/article3.pdf] (Accessed on: 11.04.2005).
6. Alici Y. 1995. Neural networks in corporate failure prediction: The UK experience. *Proceedings of Third International Conference on Neural Networks in the Capital Markets*. Refenes AN, Abu-Mostafa Y, Moody J, Weigend A (eds.), London, UK, Oct. 1995, pp. 393–406.
7. Altman EI. 1968. Financial Ratios, Discriminant Analysis and the Prediction of Corporate Bankruptcy. *The Journal of Finance* **23**: 589-609.
8. Altman EI, Marco G, Varetto F. 1994. Corporate distress diagnosis: Comparisons using linear discriminant analysis and neural networks. *Journal of Banking and Finance* **18**(3): 505–529.
9. American Productivity and Quality Centre 2005. [Available at: http://www.apqc.org/portal/apqc/site/generic2?path=/site/benchmarking/benchmarking_bebefits ofbenchmarking.jhtml] (Accessed on: 24.06.2005).
10. Amin R, Bramer MA, Emslie R. 2003. Intelligent Data Analysis for Conservation: Experiments with Rhino Horn Fingerprint Identification. *Knowledge Based Systems* **16**(5-6): 329-336. ISSN 0950-7051.
11. An A, Chan C, Shan N, Cercone N, Ziarko W. 1997. Applying Knowledge Discovery to Predict Water-Supply Consumption. *IEEE Expert* **12**(4): 72-78.
12. Anandarajan M, Lee P, Anandarajan A. 2001. Bankruptcy Prediction of Financially Stressed Firms: An Examination of the Predictive Accuracy of Artificial Neural Networks. *International Journal of Intelligent Systems in Accounting, Finance and Management* **10**(2): 69-81.
13. Anderson E. 1935. The irises of the Gaspe' peninsula. *Bull Am Iris Soc* **59**: 2-5.
14. Ankenbrandt CA. 1991. An extension to the theory of convergence and a proof of the time complexity of genetic algorithms. *Proceedings of 4th International Conference on Genetic Algorithm*, pp. 53-68.
15. Atiya AF. 2001. Bankruptcy Prediction for Credit Risk Using Neural Networks: A Survey and New Results. *IEEE Transactions on Neural Networks* **12**(4): 929-935.
16. BACH database. 2005. [Available at: http://europa.eu.int/comm/economy_finance/indicators/ bachdatabase_en.htm] (Accessed on: 11.09.2005).
17. Back B, Sere K, Vanharanta H. 1996a. Data Mining Accounting Numbers Using Self Organising Maps. *Proceedings of Finnish Artificial Intelligence Conference*, Vaasa, Finland, August 20-23, 1996.
18. Back B, Laitinen T, Sere K, van Wezel, M. 1996b. Choosing Bankruptcy Predictors Using Discriminant Analysis, Logit Analysis, and Genetic Algorithms. *TUCS Technical Report* **40**, September 1996.
19. Back B, Laitinen T, Hekanaho J, Sere K. 1997. The Effect of Sample Size on Different Failure Prediction Methods. *TUCS Technical Report* **155**, December 1997.
20. Back B, Sere K, Vanharanta H. 1998. Managing Complexity in Large Databases Using Self-Organizing Maps. *Accounting Management and Information Technologies* **8**(4):191-210.

21. Back B, Toivonen J, Vanharanta H, Visa A. 2001. Comparing numerical data and text information from annual reports using self-organizing maps. *International Journal of Accounting Information Systems* **2**(4): 249-269.
22. Banville C, Landry M. 1989. Can the Field of MIS be Disciplined? *Communications of the ACM* **32**(1): 48-60.
23. Barth S. 2004. Integrating Knowledge Management and Competitive Intelligence; Integrating Offense and Defense. *Knowledge Management – Lessons Learned: What Works and What Doesn't* (Chapter 28 – pp. 461-481). Michael ED. Koenig and T. Kanti Srikantaiah (eds.). Information Today, Inc., Medford, New Jersey.
24. Basheer IA, Hajmeer M. 2000. Artificial neural networks: fundamentals, computing, design, and application. *Journal of Microbiological Methods* **43**: 3-31.
25. Beaver WH. 1966. Financial Ratios as Predictors of Failure, Empirical Research in Accounting: Selected Studies. *Supplement to Journal of Accounting Research* **4**: 71-111.
26. Beerkens R. 2001. Future industrial glass melting concepts. *Proceedings of International Congress on Glass* – Invited Paper, Edinburgh, Scotland, July 1-6, 2001, pp. 180-192.
27. Bendell T, Boulter L, Goodstadt P. 1998. *Benchmarking for Competitive Advantage*. Pitman Publishing: London.
28. Berardi VL, Zhang GP. 2003. An Empirical Investigation of Bias and Variance in Time Series Forecasting: Model Considerations and Error Evaluation. *IEEE Transactions on Neural Networks* **14**: 668-680.
29. Berthold M. 1999. Fuzzy Logic. *Intelligent Data Analysis – An Introduction* (Chapter 8 – pp. 269-298). Michael Berthold and David J. Hand (eds.). Springer, Legoprint S.r.l., Lavis, Italy.
30. Bhattacharyya S, Pendharkar PC. 1998. Inductive, evolutionary and neural techniques for discrimination: An empirical study. *Decision Sciences* **29**(4): 871-899.
31. Bhutta KS, Huq F. 1999. Benchmarking – best practices: an integrated approach. *Benchmarking: An International Journal* **6**(3): 254-268.
32. Bishop CM. 1995. *Neural Networks for Pattern Recognition*. Oxford University Press, Oxford.
33. Bock HH. 2002. The Goal of Classification. *Handbook of Data Mining and Knowledge Discovery – Data Mining Tasks and Methods* (Section 16.1.1 – pp. 254-258). Willi Klösgen and Jan M. Zytkow (eds.), Oxford University Press, New York, NY.
34. Bradley PS, Fayyad UM, Mangasarian OL. 1999. Mathematical programming for data mining: formulations and challenges. *INFORMS Journal on Computing* **11**(3): 217-238. [Available at: http://citeseer.ist.psu.edu/bradley98mathematical.html] (Accessed on: 26.07.2005).
35. Bramer MA. 2000. Inducer: a Rule Induction Workbench for Data Mining *Proceedings of the IFIP World Computer Congress Conference on Intelligent Information Processing*, Z.Shi, B.Faltings and M.Musen (eds.), Publishing House of Electronics Industry (Beijing), pp. 499-506.
36. Braun H, Chandler JS. 1987. Predicting Stock Market Behavior through Rule Induction: an Application of the Learning-from-Examples Approach. *Journal of Decision Sciences* **18**(3): 415-429.
37. Breiman L, Friedman JH, Olshen R, Stone C. 1984. *Classification and Regression Trees*. Wadsworth and Brooks, Monterey, CA.
38. Burrell G, Morgan G. 1979. *Sociological Paradigms and Organisational Analysis*. Heineman: London.
39. Camp RC. 1989. *Benchmarking – The Search for Industry Best Practices that Lead to Superior Performance*. American Society for Quality, Quality Press, Milwaukee, Wisconsin.
40. Carlsson C, Fuller R. 2002. *Fuzzy Reasoning in Decision Making and Optimization*. Series of Studies in Fuziness and Soft Computing, Janusz Kacprzyk (ed.), Physica-Verlag, Heidelberg, Germany.
41. Central Intelligence Agency. 2005. [Available at: http://www.cia.gov/cia/publications/facttell/intelligence_cycle.html] (Accessed on: 17.07.2005).
42. Checkland P, Holwell S. 1998. *Information, Systems and Information Systems*. John Wiley & Sons, UK.
43. Chellapilla K, Fogel DB. 1999. Evolution, Neural Networks, Games, and Intelligence. *Proceedings of the IEEE*, September: 1471-1496.
44. Chua WF. 1986. Radical Developments in Accounting Thought. *Accounting Review* **61**(5): 583-598.

45. Coakley JR, Brown CE. 2000. Artificial Neural Networks in Accounting and Finance: Modelling Issues. *International Journal of Intelligent Systems in Accounting, Finance & Management* **9**: 119-144.
46. Costa E. 2000. Classification Problems: an old question with new solutions? [Available at: http://citeseer.ist.psu.edu/468768.html] (Accessed on: 07.10.2004).
47. Costea A, Kloptchenko A, Back B. 2001. Analyzing Economical Performance of Central-East-European Countries Using Neural Networks and Cluster Analysis. *Proceedings of the Fifth International Symposium on Economic Informatics*, I. Ivan. and I. Rosca (eds), Academy of Economic Studies Press, Bucharest, Romania, pp. 1006-1011.
48. Costea A, Eklund T, Karlsson J. 2002a. Making Financial Class Predictions Using SOM and Two Different Classification Techniques: Multinomial Logistic Regression and Decision Tree Induction. *Proceedings of the Central & Eastern European Workshop on Efficiency and Productivity Analysis.* AES Press, Bucharest, Romania, June 28-29.
49. Costea A, Eklund T, Karlsson J. 2002b. A framework for predictive data mining in the telecommunication sector. *Proceedings of the IADIS International Conference - WWW/Internet*, Isaías P. (ed.). IADIS Press, Lisbon, Portugal. November 13-15.
50. Costea A. 2003. Economic Performance Classification Using Neural Networks. *Proceedings of the Sixth International Symposium on Economic Informatics*, I. Ivan. and I. Rosca (eds), Academi of Economic Studies Press, Bucharest, Romania.
51. Costea A, Eklund T. 2003. A Two-Level Approach to Making Class Predictions. *Proceedings of 36th Annual Hawaii International Conference on System Sciences (HICSS 2003)*, Sprague Jr RH. (ed.), IEEE Computer Society, Hawaii, USA, January 6-9, 2003, Track: Decision Technologies for Management, Minitrack: Intelligent Systems and Soft Computing (***Publication 2***).
52. Costea A, Eklund T. 2004. Combining Clustering and Classification Techniques for Financial Performance Analysis. *Proceedings of 8th World Multi-Conference on Systemics, Cybernetics and Informatics (SCI 2004)*, Callaos *et al*. (eds.), Organized by IIIS, Orlando, Florida, USA, July 18-21, 2004, Volume I: Information Systems, Technologies and Applications, Track: Management Information Systems, pp. 389-394 (***Publication 3***).
53. Costea A, Nastac I. 2005. Three Factors Affecting the Predictive Performance of ANNs: preprocessing method, data distribution and training mechanism. *TUCS Technical Report* **679**, April 2005.
54. Costea A, Nastac I. 200x. Assessing the Predictive Performance of ANN-based Classifiers Based on Different Data Preprocessing Methods, Distributions and Training Mechanisms. Submitted to the *International Journal of Intelligent Systems in Accounting, Finance and Management* (***Publication 5***).
55. Dattakumar R, Jagadeesh R. 2003. A Review of Literature on Benchmarking. *Benchmarking: An International Journal* **10**(3): 176-209.
56. Davenport TH, Prusak L. 1998. *Working knowledge: how organizations manage what they know*. Harvard Business School Press, Boston, Massachusetts.
57. Davis L (ed.). 1991. *Handbook of Genetic Algorithms*. Van Nostrand Reinhold: New York.
58. De Andres J. 2001. Statistical Techniques vs. SEES Algorithm. An Application to a Small Business Environment**.** *The International Journal of Digital Accounting Research* **1**(2): 153-178.
59. De Bodt E, Cottrell M, Verleysen M. 2002. Statistical tools to assess the reliability of self-organizing maps. *Neural Networks* **15**: 967-978.
60. Deboeck G. 1998. Financial Applications of Self-Organizing Maps. *Neural Network World* **8**(2): 213-241.
61. Debreceny R, Gray GL. 2001. The production and use of semantically rich accounting reports on the Internet: XML and XBRL. *International Journal of Accounting Information Systems* **2**(1): 47-74.
62. DeJong KA. 1975. *An Analysis of the Behavior of a Class of Genetic Adaptive Systems*. Ph.D. dissertation, University of Michigan, Ann Arbor, MI.
63. Demuth H, Beale M. 2001. *Neural Network Toolbox*. The MathWorks Inc, Natick Press: MA, USA.
64. Dorsey RE, Mayer WJ. 1995. Genetic Algorithms for Estimation Problems with Multiple Optima, Non-differentiability, and other Irregular Features. *Journal of Business and Economic Statistics* **13**(1): 53-66.

65. Drobics M, Winiwarter W, Bodenhofer U. 2000. Interpretation of Self-Organizing Maps with Fuzzy Rules. *Proceedings of the ICTA 2000 – The Twelfth IEEE International Conference on Tools with Artificial Intelligence*, Vancouver, Canada.
66. Edmister RO. 1972. An Empirical Test Of Financial Ratio Analysis For Small Business Failure Prediction. *Journal of Financial and Quantitative Analysis* **7**: 1477-1493.
67. Eisenbeis RA. 1977. Pitfalls in the application of discriminant analysis in business, finance, and economics. *Journal of Finance* **32** (3): 875-900.
68. Eklund T. 2004. *The Self-Organizing Map in Financial Benchmarking*. TUCS Ph.D. Dissertation, Åbo Akademi University, Turku, Finland, 2004.
69. Eklund T, Back B, Vanharanta H, Visa A. 2003. Financial Benchmarking Using Self-Organizing Maps – Studying the International Pulp and Paper Industry. *Data Mining - Opportunities and Challenges* (Chapter 14 – pp. 323-349). J. Wang, Ed. Hershey, PA, Idea Group Publishing.
70. Eklund T, Back B, Vanharanta H, Visa A. 2004. Financial Benchmarking Tools in Finnish Companies - A State of the Art Survey. *TUCS Technical Report* **618**, August 2004.
71. Elomaa T. 1994. In Defense of C4.5: Notes on Learning One-Level Decision Trees. In WW Cohen and H Hirsh (eds.), *Machine Learning: Proceedings of the Eleventh International Conference*, San Francisco, CA: Morgan Kaufmann, pp. 62-69.
72. EUNITE Competition 2003. *Prediction of product quality in glass manufacturing*. [Available at: http://www.eunite.org/eunite/events/eunite2003/competition2003.pdf] (Accessed on: 10.04.2005).
73. European Benchmarking Code of Conduct. 2005. [Available at: http://www.benchmarking.gov.uk/about_bench/whatisit.asp] (Accessed: 24.06.2005).
74. Fayyad U, Piatetsky-Shapiro G, Smyth P. 1996a. Knowledge Discovery and Data Mining: Towards a Unifying Framework. *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD'96)*, E. Simoudis, J. Han, and U. Fayyad (eds.), AAAI Press, Portland, Oregon, August 2-4, pp. 82-88.
75. Fayyad U, Piatetsky-Shapiro G, Smyth P. 1996b. From Data Mining to Knowledge Discovery in Databases. *AI Magazine* **17**(3):37-54.
76. Fayyad U, Piatetsky-Shapiro G, Smyth P. 1996c. From Data Mining to Knowledge Discovery: An Overview. *Advances in Knowledge Discovery and Data Mining*. Fayyad U, Piatetsky-Shapiro G, Smyth P, Uthurusamy R (eds.), AAAI Press, Menlo Park, California, pp. 1-30.
77. Fisher RA. 1936. The use of multiple measurements in taxonomic problems. *Ann. Eugenics* **7**: 179-188.
78. Fitzgerald B, Howcroft D. 1998. Competing Dichotomies in IS Research and Possible Strategies for Resolution. *Proceedings of International ACM Conference on Information Systems*, Helsinki, Finland.
79. Fogel DB, Wasson EC, Boughton EM. 1995. Evolving Neural Networks for Detecting Breast Cancer. *Cancer Letters* **96**: 49-53.
80. Fogel DB, Wasson EC, Boughton EM, Porto VW. 1998. Evolving artificial neural networks for screening features from mammograms. *Artificial Intelligence in Medicine* **14**: 317-326.
81. Fogel GB, Weekes DG, Sampath R, Ecker DJ. 2004. Parameter Optimization of an Evolutionary Algorithm for RNA Structure Discovery. *Proceedings of 2004 Congress on Evolutionary Computation*, IEEE Press, Piscataway, NJ, pp. 607-613.
82. Fortune magazine. 2005. [Available at: http://www.fortune.com/fortune/] (Accessed on: 9.09.2005).
83. Foster G. 1986. *Financial Statement Analysis*. Englewood Cliffs, New Jersey, Prentice-Hall Inc.
84. Fox J. 1997. *Applied Regression Analysis, Linear Models, and Related Methods*. Newbury Park, CA: Sage.
85. Frydman H, Altman EI, Kao DL. 1985. Introducing Recursive Partitioning for Financial Classification: The Case of Financial Distress. *The Journal of Finance* **40**: 269-291.
86. Garrison LR, Michaelsen RH. 1989. Symbolic Concept Acquisition: A New Approach to Determining Underlying Tax Law Constructs. *Journal of American Tax Association* **11**(1): 77-91.
87. Garson GD. 2005. *PA 765 Statnotes: An Online Textbook*. Work in progress. [Available at: http://www2.chass.ncsu.edu/garson/pa765/statnote.htm] (Accessed on: 16.09.2005).
88. Glass Service, Inc. 2005. [Available at: http://www.gsl.cz/] (Accessed on: 7.07.2005).

89. Greengard S. 1995. Discover best practices from benchmarking. *Personnel Journal* **74**(11): 62-73.
90. Guiver JP, Klimasauskas CC. 1991. Applying Neural Networks, Part IV: Improving Performance. *PC/AI Magazine* **5**(4): 34-41.
91. Hagan MT, Demuth HB, Beale M. 1996. *Neural Network Design*. PWS Publishing Company, Boston, USA.
92. Hair JF, Anderson-Jr. R, Tatham RL. 1987. *Multivariate Data Analysis with readings*. 2nd Edition, Macmillan Publishing Company, New York, New York, USA.
93. Hamer M. 1983. Failure Prediction: Sensitivity of classification accuracy to alternative statistical method and variable sets. *Journal of Accounting and Public Policy* **2**(Winter): 289-307.
94. Han J, Kamber M. 2000. *Data Mining: Concepts and Techniques*. Morgan Kaufmann Publishers, ISBN: 1-55860-489-8.
95. Hand DJ, Mannila H, Smyth P. 2001. *Principles of Data Mining*. The MIT Press, Cambridge.
96. Hand DJ. 2002. Statistics. *Handbook of Data Mining and Knowledge Discovery – Data Mining Tasks and Methods* (Chapter 25 – pp. 637-643). Willi Klösgen and Jan M. Zytkow (eds.), Oxford University Press, New York, NY.
97. Hansen RJ, Hall DL, Nickerson GW, Phoha S. 1996. Integrated Predictive Diagnostics: An Expanded View. *Proceedings of International Gas Turbine and Aeroengine Congress and Exhibition*, Birmingham, UK.
98. Hassard J. 1991. Multiple Paradigms and Organizational Analysis: A Case Study. *Organizational Studies* **12**(2): 275-299.
99. Haykin S. 1995. *Neural Networks: A Comprehensive Foundation*. Second edition, New York: Prentice-Hall.
100. Hesser J, Männer R. 1990. Towards an Optimal Mutation Probability for Genetic Algorithms. *Proceedings of the 1st Workshop on Parallel Problem Solving from Nature*, Sringer-Verlag, October 01-03, pp.23-32.
101. Hevner AR, March ST, Park J, Ram S. 2004. Design Science in Information Systems Research. *MIS Quarterly* **28**(1): 75-105.
102. Hoehn PT. 1998. Wolves in Sheep's Clothing? The Effects of "Hidden" Parental Mutation on Genetic Algorithm Performances. *Proceedings of ACM 36th annual Southeast regional conference*, pp. 221-227.
103. Holte R. 1993. Very Simple Classification Rules Perform Well on Most Commonly Used Datasets. *Machine Learning* **11**: 63–91.
104. Hornik K, Stinchcombe M, White H. 1989. Multilayer feedforward networks are universal approximators. *Neural Networks* **2**: 359-366.
105. Hosmer DW, Lemeshow S. 2000. *Applied Logistic Regression*. 2nd Edition, John Wiley and Sons, New York, USA.
106. Hunt EB. 1962. *Concept Learning: An Information Processing Problem*. Wiley and Sons, New York, NY.
107. Hunt EB, Marin J, Stone PJ. 1966. *Experiments in Induction*. Academic Press, New York.
108. Iivari J. 1991. A paradigmatic analysis of contemporary schools of IS development. *European Journal of Information Systems* **1**(4): 249-272.
109. Iivari J, Hirscheim R, Klein H. 1998. A Paradigmatic Analysis Contrasting Information System Development Approaches and Methodologies. *Information Systems Research* **9**(2): 164-193.
110. Jackson JE. 1991. *A user guide to principal components*. John Wiley, New York.
111. Jain AK, Murty MN, Flynn PJ. 1999. Data Clustering: A Review. *ACM Computing Survey* **31**(3).
112. Järvinen P. 2001. *On research methods*. Opinpajan Kirja, Tampere, Finland.
113. Jeng B, Jeng YM, Liang TP. 1997. FILM: A Fuzzy Inductive Learning Method for Automatic Knowledge Acquisition. *Decision Support Systems* **21**(2): 61-73.
114. Jennings NR, Wooldridge M. 1998. Applications of Intelligent Agents. *Agent Technology Foundations, Applications, and Markets*. NR Jennings and MJ Wooldridge (eds.), Springer-Verlag.
115. Jones F. 1987. Current techniques in bankruptcy prediction. *Journal of Accounting Literature* **6**: 131-164.

116. Kaplan B, Duchon D. 1988. Combining Qualitative and Quantitative Methods in IS Research: A Case Study. *MIS Quaterly* **12**(4): 571-587.
117. Karlsson J, Back B, Vanharanta H, Visa A. 2001. Financial Benchmarking of Telecommunications Companies. *TUCS Technical Report* **395**, February 2001.
118. Karlsson J. 2002. *Data-Mining, Benchmarking and Analysing Telecommunications Companies*. Licentiate dissertation, Åbo Akademi University, Turku, Finland, 2002.
119. Kasabov N, Erzegovezi L, Fedrizzi M, Beber A, Deng D. 2000. Hybrid Intelligent Decision Support Systems and Applications for Risk Analysis and Prediction of Evolving Economic Clusters in Europe. *Future Directions for Intelligent Information Systems and Information Sciences*. N. Kasabov (ed.). Springer Verlag, pp. 347-372.
120. Kasanen E, Lukka K, Siitonen A. 1993. The Constructive Approach in Management Accounting Research. *Journal of Management Accounting Research* **5**: 243-264.
121. Kaski T, Kohonen T. 1996. Exploratory data analysis by the Self-Oraganizing Map: structures of welfare and poverty in the world. In Apostolos-Paul N. Refenes, Yaser Abu-Mostafa, John Moody, and Andreas Weigend (eds.) Neural Networks in Financial Engineering. *Proceedings of the Third International Conference on Neural Networks in the Capital Markets*, London, England 11-13 October, 1995. World Scientific, Singapore, pp. 498-507.
122. Kattan MW, Adams DA, Parks MS. 1993. A Comparison of Machine Learning with Human Judgment. *Journal of Management Information Systems* **9**(4): 37-58.
123. Kaymak U, van den Berg J. 2004. On Constructing Probabilistic Fuzzy Classifiers from weighted Fuzzy Clustering. *Proceedings of IEEE International Joint Conference on Neural Networks (IJCNN'04)*, Budapest, Hungary, July 25-29.
124. Khalid M, Omatu S. 1992. A neural network controller for a temperature control system. *IEEE Control Systems Magazine* **12**(3): 58–64.
125. Khasnabis S, Arciszewski T, Hoda S, Ziarko W. 1994. Urban Rail Corridor Control through Machine Learning: An Intelligent Vehicle-Highway System Approach. *Transportation Research Record* **1453**: 91-97.
126. Kiang MY. 2001. Extending the Kohonen self-organising map networks for clustering analysis. *Computational Statistics & Data Analysis* **38**: 161-180.
127. Kling R. 1987. Defining the boundaries of computing across complex organizations. In Boland and Hirschheim (eds.) Critical issues in information systems research. Wiley, New-York, pp. 307-362.
128. Kloptchenko A. 2003. *Text Mining Based on the Prototype Matching Method*. TUCS Ph.D. dissertation, Åbo Akademi University, Turku, Finland, 2003.
129. Kloptchenko A, Eklund T, Costea A, Back B. 2003. A Conceptual Model for a Multiagent Knowledge Building System. *Proceedings of 5th International Conference on Enterprise Information Systems (ICEIS 2003)*, Camp O, Filipe J, Hammoudi S, Piattini M (eds.), published by Escola Superior de Tecnologia do Instituto Politécnico de Setúbal, Angers, France, April 23-26, 2003, Volume 2: Artificial Intelligence and Decision Support Systems, pp. 223-228 (***Publication 1***).
130. Kloptchenko A, Eklund T., Karlsson J, Back B, Vanharanta h, Visa A. 2004. Combining Data and Text Mining Techniques for Analysing Financial Reports. *International Journal of Intelligent Systems in Accounting, Finance and Management* **12**(1):29-41.
131. Klösgen W, Zytkow JM. 2002. *Handbook of Data Mining and Knowledge Discovery*, Oxford University Press, New York, NY.
132. Klösgen W, Anand TS. 2002. Subgroup Discovery. *Handbook of Data Mining and Knowledge Discovery – Data Mining Tasks and Methods* (Section 16.3 – pp. 354-367). Willi Klösgen and Jan M. Zytkow (eds.), Oxford University Press, New York, NY.
133. Kogut B, Zander U. 1992. Knowledge of the Firm. Combinative Capabilities, and the Replication of Technology. *Organization Science* **3**(3): 383-397.
134. Kohavi R, Quinlan JR. 2002. Decision-Tree Discovery. *Handbook of Data Mining and Knowledge Discovery – Data Mining Tasks and Methods* (Section 16.1.3 – pp. 267-276). Willi Klösgen and Jan M. Zytkow (eds.), Oxford University Press, New York, NY
135. Kohonen T, Hynninen J, Kangas J, Laaksonen J. 1995. *SOM_PAK the selforganizing map program package, v:3.1*. Laboratory of Computer and Information Science, Helsinki University

of Technology, Finland, April, 1995. [Available at: http://www.cis.hut.fi/research/som_pak] (Accessed on: 11.04.2005)

136. Kohonen T. 1997. *Self-Organizing Maps*. 2nd edition, Springer-Verlag, Heidelberg.
137. Kohonen T. 1998. The self-organizing map. *Neurocomputing* **21**:1-6.
138. Kolb G. 2000. Entrepreneurial CI. *Competitive Intelligence* **3**(2):56, April 2000. [Available at: http://www.scip.org/news/cimagazine_article.asp?id=266] (Accessed on: 26.07.2005).
139. Koskivaara E. 2000. Different Pre-Processing Models for Financial Accounts when Using Neural Networks for Auditing. *Proceedings of the European Conference on Information Systems*, Hans Robert Hansen - Martin Bichler - Harald Mahrer (eds.), Viena, Austria, July 3-5, 2000, pp. 326-332.
140. Koskivaara E. 2004a. Artificial Neural Networks in Analytical Review Procedures. *Managerial Auditing Journal* **19**(2): 191-223.
141. Koskivaara E. 2004b. *Artificial Neural Networks for Analytical Review in Auditing*, Ph.D dissertation, Turku, 2004.
142. Kumar CS, Chandrasekharan MP. 1990. Grouping efficacy: a quantitative criterion for goodness of block diagonal forms of binary matrices in group technology. *International Journal of Production Research* **28**(2): 233-243.
143. Kumar UA. 2005. Comparison of Neural Networks and Regression Analysis: A New Insight. *Expert Systems with Applications* **29**: 424-430.
144. Kvassov V. 2002. *Information Technology and the Productivity of Managerial Work*. TUCS Ph.D. dissertation, Åbo Akademi University, Turku, Finland, 2002.
145. Lachtermacher G, Fuller JD. 1995. Backpropagation in time series forecasting. *Journal of Forecasting* **14**: 381–393.
146. Lacroix R, Salehi F, Yang XZ, Wade KM. 1997. Effects of data preprocessing on the performance of artificial neural networks for dairy yield prediction and cow culling classification. *Transactions of the ASAE* **40**(3): 839-846.
147. Lai H, Chu T. 2000. Knowledge Management: A Theoretical Framework and Industrial Cases. *Proceedings of 33rd Hawaii International Conference on System Science*, IEEE Press, Hawaii.
148. Lainema T. 2003. *Enhancing Organizational Business Process Perception – Experiences from Constructing and Applying a Dynamic Business Game*. Ph.D dissertation, Turku School of Economic and Business Administration, Turku, Finland, 2003.
149. Lankers K, Strackeljan J. 2004. Joint Research Activities: Experiences and Results of the Competition Working Team. Paper Presented at *EUNITE 2004 Conference*, Aachen, June 2004.
150. Lee AS. 1989. A Scientific Methodology for MIS Case Studies. *MIS Quarterly* **13**(1): 33-50.
151. Lee AS. 1991. Integrating Positivist and Interpretive Approaches to Organizational Research. *Organization Science* **2**(4): 342-365.
152. Lehtinen. 1996. *Financial Ratios in an International Comparison. Validity and Reliability*. Acta Wasaensia 49, Vaasa, Finland.
153. Leski J. 2003. Towards a robust fuzzy clustering. *Fuzzy Sets and Systems* **137**: 215-233.
154. Lev B. 1974. *Financial Statement Analysis*. Englewood Cliffs, New Jersey, Prentice-Hall Inc.
155. Li EY. 1994. Artificial Neural Networks and their Business Applications. *Information & Management* **27**: 303-313.
156. Liau L C-K, Chen B S-C. 2005. Process Optimization of Gold Stud Bump Manufacturing Using Artificial Neural Networks. *Expert Systems with Applications* **29**: 264-271.
157. Lindholm CK, Liu S. 2003. Fuzzy Clustering Analysis of the Early Warning Signs of Financial Crisis. *Proceedings of the FIP2003, an International Conference on Fuzzy Information Processing: Theory and Applications*, Beijing, March 1–4, 2003.
158. Lindström T. 1998. A fuzzy design of the willingness to invest in Sweden. *Journal of Economic Behavior and Organization* **36**: 1-17.
159. Liu S. 1998. Business Environment Scanner for Senior Managers: Towards Active Executive Support with Intelligent Agents. *Expert Systems with Applications* **15**: 111-121.
160. Liu S. 2000. *Improving Executive Support in Strategic Scanning with Software Agent Systems*. Ph.D. dissertation, Åbo Akademi University, Turku, Finland, 2000.
161. Loskiewicz-Buczak A, Uhrig RE. 1992. Probabilistic Neural Network for Vibration Data Analysis. *Intelligent Engineering Systems Through Artificial Neural Networks*, CH Dagli, LI Burke, and YC Shin (eds.), New York: ASME Press, pp. 713-718.

162. Löffler G. Avoiding the Rating Bounce: Why Rating Agencies Are Slow to React to New Information. *Journal of Economic Behavior and Organization* (in press).

163. Lyman P, Varian HR. 2003. "How Much Information", 2003. [Available at: http://www.sims.berkeley.edu/how-much-info-2003] (Accessed on: 26.11.2004).

164. Maag GD, Flint JA. 2004. The Role of Corporate Intelligence Gathering in the Modern Business Decision-Making Process. *Knowledge Management – Lessons Learned: What Works and What Doesn't* (Chapter 26 – pp. 403-440). Michael ED. Koenig and T. Kanti Srikantaiah (eds.). Information Today, Inc., Medford, New Jersey.

165. MacQueen JB. 1967. Some Methods for classification and Analysis of Multivariate Observations. *Proceedings of 5th Berkeley Symposium on Mathematical Statistics and Probability*, Berkeley, University of California Press **1**, pp. 281-297.

166. Mangiameli P, Chen SK, West D. 1996. A Comparison of SOM neural network and hierarchical clustering methods. *European Journal of Operational Research* **93**(2): 402-417.

167. Marais ML, Patell JM, Wolfson MA. 1984. The experimental design of classification models: An application of recursive partitioning and bootstrapping to commercial bank loan classification. *Journal of Accounting Research* **22**: 87-114.

168. March ST, Smith GF. 1995. Design and natural science research on information technology. *Decision Support Systems* **15**(4): 251-266.

169. Marmelstein RE, Lamont GB. 1998. Evolving Compact Decision Rule Sets. In Koza, John R. (eds.). *Late Breaking Papers at the Genetic Programming 1997 Conference*, Omni Press, pp. 144-150.

170. Martín-del-Brío B, Serrano Cinca C. 1993. Self Organizing Neural Networks for the Analysis and Representation of Data: some Financial Cases. *Neural Computing & Applications*. Springer Verlag (ed.) **1**(2): 193-206.

171. Masters T. 1994. *Practical Neural Network Recipes in C++*. Academic Press, Boston, MA.

172. MathWorks, Inc. 2001. *Neural Networks Toolbox User's Guide*. Natick Press.

173. McNair CJ, Liebfried KHJ. 1992. *Benchmarking – A Tool for Continuous Improvement*. John Wiley and Sons, Inc., New York, NY.

174. McNeilly MR. 2000. *Sun Tzu and the Art of Business: Six Strategic Principles for Managers*. New York, NY: Oxford University Press.

175. Meireles MRG, Almeida PEM, Simões MG. 2003. A Comprehensive Review for Industrial Aplicability of Artificial Neural Networks. *IEEE Transactions on Industrial Electronics* **50**(3):585-601.

176. Michalewicz Z. 1992. *Genetic Algorithms + Data Structures = Evolution Programs*. Springer, Berlin.

177. Miller JF, Thomson P. 1998. Aspects of Digital Evolution: Geometry and Learning. *Proceedings of the 2nd International Conference on Evolvable Systems - ICES98*, September 23-25, 1998, EPFL, Lausanne, Switzerland.

178. Moczulski W. 2002. Automated Search for Diagnostic Knowledge on Rotating Machinery. *Handbook of Data Mining and Knowledge Discovery – Industry* (Section 46.6 – pp. 890). Willi Klösgen and Jan M. Zytkow (eds.), Oxford University Press, New York, NY.

179. Moller MF. 1993. A scaled conjugate gradient algorithm for fast supervised learning. *Neural Networks* **6**: 525-533.

180. Moody J. 1995. Economic Forecasting: Challenges and Neural Network Solutions. Keynote talk at *International Symposium on Artificial Neural Networks*, Hsinchu, Taiwan, December.

181. Moxon B. 1996. Defining Data Mining - The Hows and Whys of Data Mining, and How It Differs From Other Analytical Techniques. [Available at: http://www.dbmsmag.com/9608d53.html] (Accessed on: 6.12.2004).

182. Mrozek A, Plonka L. 1998. Rough Sets in Industrial Applications. *Rough Sets in Knowledge Discovery*, Polkowski L, Skowron (eds.), Heidelberg: Springer-Verlag, pp. 214-237.

183. Müller J, Bódi R, Matuštík. 2004. Expert System ES III – An Advanced System for Optimal Glass Furnace Control. *Glass - Monthly Journal* **81**(9), October 2004.

184. Nagelkerke NJD. 1991. A note on the general definition of the coefficient of determination. *Biometrika* **78**: 691-692.

185. Nastac I, Koskivaara E. 2003. A Neural Network Model for Prediction: Architecture and Training Analysis. *TUCS Technical Report* **521**, April 2003. [Available at: http://www.tucs.fi/publications/insight.php?id=tNaKo03a&table=techreport]. (Accessed on: 25.04.2005).
186. Nastac I, Costea A. 2004a. A Neural Network Retraining Approach for Process Output Prediction. *Proceedings of 8th World Multi-Conference on Systemics, Cybernetics and Informatics (SCI 2004)*, Callaos *et al*. (eds.), Organized by IIIS, Orlando, Florida, USA, July 18-21, 2004, Volume V: Computer Science and Engineering, pp. 388-393.
187. Nastac I, Costea A. 2004b. A Retraining Neural Network Technique for Glass Manufacturing Data Forecasting. *Proceedings of 2004 IEEE International Joint Conference on Neural Networks (IJCNN 2004)*, IEEE, Budapest, Hungary, July 25-29, 2004, Volume 4, Track: Time Series Analysis, pp. 2753-2758 (***Publication 6***).
188. Nelson KM, Kogan A, Srivastava RP, Vasarhelyi MA, Lu H. 2000. Virtual auditing agents: the EDGAR Agent challenge. *Decision Support Systems* **28**(3): 241-253.
189. Newman M, Robey D. 1992. A Social Process Model of User-Analyst Relationships. *MIS Quarterly* **16**(2): 249-266.
190. Nguyen DH, Widrow B. 1989. The Truck Backer-Upper: An Example of Self-Learning in Neural Networks. *Proceedings of International Joint Conference on Neural Networks (IJCNN-89)*, Washington, DC, June 1989, Vol. II, pp. 357-363.
191. Nissen H, Klein H, Hirschheim R (eds.). 1991. *Information Systems Research: Contemporary Approaches and Emergent Traditions*. North-Holland, Amsterdam.
192. Nobes C, Parker R. 2002. *Comparative International Accounting*. 7th Edition, Pearson Education Limited, Essex, England.
193. Nonaka I, Takeuchi H. 1995. *The knowledge-creating company: how Japanese companies create the dynamics of innovation*. Oxford University Press, New York, NY.
194. Ohlson JA. 1980. Financial ratios and the probabilistic prediction of bankruptcy. *Journal of Accounting Research* **18**(1):109-131.
195. Oja M, Kaski S, Kohonen T. 2003. Bibliography of Self-Organizing Map (SOM) Papers: 1998-2001 Addendum. *Neural Computing Surveys* **3**: 1-156.
196. O'Leary DE. 1998. Using Neural Networks to Predict Corporate Failure. *International Journal of Intelligent Systems in Accounting, Finance & Management* **7**: 187-197.
197. Oliga JC. 1988. Methodological Foundations of System Design Methodologies. *System Practice* **1**(1): 87-112.
198. Ong J, Abidi SSR. 1999. Data Mining Using Self-Organizing Kohonen maps: A Technique for Effective Data Clustering & Visualisation. *Proceedings of International Conference on Artificial Intelligence (IC-AI'99)*, Las Vegas, June 28 – July 1, 1999.
199. Orlikowski W, Baroudi J. 1991. Studying Information Technology in Organizations: Research Approaches and Assumptions. *Information Systems Research* **2**(1): 1-28.
200. Pallett TJ, Ahmad S. 1992. Real-Time Neural Network Control of a Miniature Helicopter in a Vertical Flight. *Proceedings of the 17th International Conference on Applications of Artificial Intelligence in Engineering (AIENG-92)*, Waterloo, Canada.
201. Pearson K. 1900. *The grammar of science*. London: Black.
202. Pendharkar PC. 2002. A computational study on the performance of artificial neural networks under changing structural design and data distribution. *European Journal of OperationalResearch* **138**: 155-177.
203. Pendharkar PC, Rodger JA. 2004. An empirical study of impact of crossover operators on the performance of non-binary genetic algorithm based neural approaches for classification. *Computers & Operations Research* **31**: 481-498.
204. Popper KR. 1963. *Conjectures and Refutations: The Growth of Scientific Knowledge*. New York: Harper and Row.
205. Porter ME, Schwab K, Sala-i-Martin X, Lopez-Claros A. 2004. *The Global Competitiveness Report 2004-2005*, Palgrave MacMillan, October 2004.
206. Quigley EJ, Debons A. 1999. Interrogative Theory of Information and Knowledge. *Proceedings of SIGCPR '99*, ACM Press, New Orleans, LA., pp. 4-10.
207. Quinlan JR. 1979. Discovering rules by induction from large collections of examples. *Expert Systems in the Micro Electronic Age*, D. Michie (ed.), Edinburgh University Press, Edinburgh, UK.

208.Quinlan JR. 1983. Learning Efficient Classification Procedures. *Machine Learning: An Artificial Intelligence Approach*, Michalski RS, Carbonell JG, Mitchell TM (eds.), Tioga Press, Palo Alto, CA.

209.Quinlan JR. 1986. Induction of Decision Trees. *Machine Learning 1*, **1**: 81-106.

210.Quinlan JR. 1988. Decision trees and multi-valued attributes. *Machine Intelligence* **11**: 305-318.

211. Quinlan JR. 1993a. A Case Study in Machine Learning. *Proceedings of ACSC-16 Sixteenth Australian Computer Science Conference*, January, Brisbane, pp. 731-737.

212. Quinlan JR. 1993b. *C4.5 Programs for Machine Learning*. Morgan Kaufmann Series in Machine Learning. Morgan Kaufmann Publishers, San Mateo.

213. Quinlan JR. 1997. *See5/C5.0*. [Available at: http://www.rulequest.com] (Accessed on: 18.04.2005).

214. Rao JS, Potts WJE. 2002. Multidimensional Regression Analysis. *Handbook of Data Mining and Knowledge Discovery – Data Mining Tasks and Methods* (Section 16.4.3 – pp. 380-386). Willi Klösgen and Jan M. Zytkow (eds.), Oxford University Press, New York, NY.

215. Razi MA, Athappilly K. 2005. A comparative predictive analysis of neural networks (NNs), nonlinear regression and classification and regression tree (CART) models. *Expert Systems with Applications* **29** (2005): 65-74.

216. Rhiannon J, Jewitt C, Galasso L, Fortemps G. 2001. Consolidation Changes the Shape of the Top 150. *Pulp and Paper International* **43**(9): 31-41.

217. Richard MD, Lippmann, RP. 1991. Neural network classifiers estimate Bayesian a posteriori probabilities. *Neural Computation* **3**(4): 461–483.

218. Rogero JM. 2002. *A Genetic Algorithms Based Optimisation Tool for the Preliminary Design of Gas Turbine Combustors*. PhD dissertation, Cranfield University, November 2002.

219. Romeu JL. 2001. Operations Research/Statistics Techniques: A Key to Quantitative Data Mining. *Proceedings of FCSM (Federal Committee on Statistical Methodology) Conference*, Key Bridge Marriott, Arlington,Virginia, November 14-16.

220. Rovithakis GA, Gaganis VI, Perrakis SE, Christodoulou MA. 1999. Real-Time Control of Manufacturing Cells Using Dynamic Neural Networks. *Automatica* **35**(1): 139-149.

221. Rudolfer S, Paliouras G, Peers I. 1999. A Comparison of Logistic Regression to Decision Tree Induction in the Diagnosis of Carpal Tunnel Syndrome. *Computers and Biomedical Research* **32**: 391-414.

222. Rudolph G. 1994. Convergence analysis of canonical genetic algorithms. *IEEE Transactions on Neural Networks* **5**: 96-101.

223. Schaeffer HA. 2000. *Credit Risk Management: A Guide to Sound Business Decisions*, Wiley.

224. Schaffer JD, Whitley D, Eshelman LJ. 1992. Combinations of Genetic Algorithms and Neural Networks: A survey of the state of the art. *COGANN-92 Combinations of Genetic Algorithms and Neural Networks*. IEEE Computer Society Press: Los Alamitos, CA, pp. 1-37.

225. Schaffer JD. 1994. Combinations of genetic algorithms with neural networks or fuzzy systems. *Computational Intelligence: Imitating Life*, Zurada JM, Marks RJ, Robinson CJ (eds). IEEE Press: New York, pp. 371-382.

226. Schütze H, Hull D, Pedersen J. 1995. A comparison of classifiers and document representations for the routing problem. *Proceedings of the 18th annual international ACM SIGIR conference on Research and development in information retrieval*, Seattle, Washington, United States. ACM Press: New York, NY, USA, pp. 229-237.

227. Schwartz EI. 1992. Where Neural Networks are Already at Work: Putting AI to Work in the Markets. *Business Week*, November 2, pp. 136-137.

228. Schwartz EI, Treece JB. 1992. Smart Programs Go To Work: How Applied Intelligence Software Makes Decisions for the Real World. *Business Week*, March 2, pp. 97-105.

229. Serrano Cinca C. 1996. Self Organizing Neural Networks for Financial Diagnosis. *Decision Support Systems*. Elsevier Science **17**: 227-238.

230. Serrano Cinca C. 1998a. Self-organizing Maps for Initial Data Analysis: Let Financial Data Speak for Themselves. *Visual Intelligence in Finance using Self-organizing Maps*. Guido Deboeck and Teuvo Kohonen (eds.). Springer Verlag, July 1998.

231. Serrano Cinca C. 1998b. From Financial Information to Strategic Groups - a Self Organizing Neural Network Approach. *Journal of Forecasting*. John Wiley and Sons (ed.), September 1998, **17**: 415-428.

232. Sexton RS, Dorsey Re, Johnson JD. 1998. Toward a global optimum for neural networks: A comparison of the genetic algorithm and backpropagation. *Decision Support Systems* **22**(2): 171-186.
233. Sexton RS, Sikander NA. 2001. Data Mining Using a Genetic Algorithm-Trained Neural Network. *International Journal of Intelligent Systems in Accounting, Finance & Management* **10**: 201-210.
234. Shearer C. 2000. The CRISP-DM Model: The New Blueprint for Data Mining. *Journal of Data Warehousing* **5**(4): 13-22.
235. Shimodaira H. 1996. A New Genetic Algorithm Using Large Mutation Rates and Population-Elitist Selection (GALME). *Proceedings of the 8th International Conference on Tools with Artificial Intelligence (ICTAI '96)*, pp. 25-32.
236. Shirata CY. 2001. The Relationship between Business Failure and Decision Making by Manager: Empirical Analysis. *Proceedings of 13th Asian-Pacific Conference on International Accounting Issues*, October 28, 2001, pp. 20-23.
237. Siegel S, Castellan-Jr. NJ. 1988. *Nonparametric Statistics for the Behavioral Sciences*. 2nd edition, McGraw-Hill International Editions.
238. Simon HA. 1981. *The Sciences of the Artificial*. Second edition. Cambridge, MA: MIT Press.
239. Simon HA. 1996. *The Sciences of the Artificial*. Third edition. Cambridge, MA: MIT Press.
240. Smyth P. 2002. Selection of Tasks. *Handbook of Data Mining and Knowledge Discovery – Task and Method Selection* (Section 17.1 – pp. 443-444). Willi Klösgen and Jan M. Zytkow (eds.), Oxford University Press, New York, NY.
241. Spirtes PL. 2002. Probabilistic and Causal Networks. *Handbook of Data Mining and Knowledge Discovery – Data Mining Tasks and Methods* (Section 16.6 – pp. 396-409). Willi Klösgen and Jan M. Zytkow (eds.), Oxford University Press, New York, NY.
242. SPSS for Windows. Release 11.5.1 (16 Nov 2002). Chicago: SPSS Inc.
243. Staib WE, Staib RB. 1992. The Intelligence Arc Furnace Controller: A Neural Network Electrode Position Optimization System for the Electric Arc Furnace. *Proceedings of IEEE International Joint Conference on Neural Networks*, New York, 1992.
244. Stenmark D. 2002. Information vs. Knowledge: The Role of intranets in Knowledge Management. *Proceedings of HICSS-35*, IEEE Press, Hawaii, January 7-10, 2002.
245. Tam K, Kiang M. 1992. Managerial applications of the neural networks: The case of bank failure predictions. *Management Science* **38**: 416–430.
246. Tan RPGH, van den Berg J, van den Bergh W-M. 2002. Credit Rating Classification using Self Organizing Maps. *Neural Networks in Business: Techniques and Applications*. Smith K, Gupta J (eds.). IDEA Group Publishing, Hershey, USA, pp. 140-153.
247. Theodoridis S, Koutroumbas K. 2003. *Pattern Recognition*. 2nd Edition. Academic Press, San Diego, CA, USA.
248. Toivonen J, Visa A, Vesanen T, Back B, Vanharanta H. 2001. Validation of Text Clustering Based on Document Contents. *Machine Learning and Data Mining in Pattern Recognition (MLDM 2001)*, Leipzig, Germany.
249. Treacy WF, Carey MS. 1998. Credit Risk Rating at Large U.S. Banks. *Federal Reserve Bulletin*: 897 - 921.
250. Tuson A, Ross P. 1998. Adapting Operator Settings in Genetic Algorithms. *Evolutionary Computation* **6**(2): 161-184.
251. Ultsch A. 1993. Self organized feature maps for monitoring and knowledge aquisition of a chemical process. Gielen S, Kappen B. (eds.), *Proceedings of the International Conference on Artificial Neural Networks (ICANN93)*, London, Springer-Verlag, pp. 864-867.
252. Upadhyaya BR, Eryurek E. 1992. Application of neural network for sensory validation and plant monitoring. *Neural Technology* **97**: 170–176.
253. Vafaie H, DeJong K. 1998. Feature Space Transformation Using Genetic Algorithms. *IEEE Intelligent Systems* **13**(2): 57-65.
254. Venayagamoorthy GK, Harley RG. 1999. Experimental Studies with a Continually Online Trained Artificial Neural Network Controller for a Turbogenerator. *Proceedings of International Joint Conference on Neural Networks*, Vol. 3, Washington, July, pp. 2158-2163.
255. Venayagamoorthy GK, Harley RG, Wunsch DC. 2001. Experimental studies with Continually Online Trained Artificial Neural Network Identifiers for Multiple Turbogenerators on the

Electric Power Grid. *Proceedings of IEEE International Conference on Neural Networks*, Vol. 2, Washington, DC, USA, July 15-19, 2001, pp. 1267-1272.

256. Vesanto J, Alhoniemi E. 2000. Clustering of the Self-Organizing Map. *IEEE Transactions on Neural Networks* **11**(3): 586-600.
257. Vesanto J, Himberg J, Alhoniemi E, Parhankangas J. 2000. SOM Toolbox for Matlab 5. *Technical Report* **A57**, Helsinki University of Technology, Espoo.
258. Visa A, Toivonen J, Back B, Vanharanta H. 2002. Contents Matching Defined by Prototypes: Methodology Verification with Books of the Bible. *Journal of Management Information Systems* **18**(4): 87-100.
259. Wallenius D. 1975. Interactive Multiple Criteria Decision Methods: An Investigation and an Approach. *Acta Academiae Oeconomiae Helsingiensis*, Series A: 14, 1975.
260. Walsham G. 1995. The Emergence of Interpretivism in IS Research. *Information Systems Research* **6**(4): 376-394.
261. Wang H, Mylopoulos J, Liao S. 2002. Intelligent Agents and Financial Risk Monitoring Systems. *Communications of the ACM* **45**(3): 83-88.
262. Wang S. 2001. Cluster Analysis Using a Validated Self-Organizing Method: Cases of Problem Identification. *International Journal of Intelligent Systems in Accounting, Finance and Management* **10**(2): 127-138.
263. Ward JH. 1963. Hierarchical grouping to optimize an objective function. *Journal of the American Statistical Association* **58**: 236-244.
264. Weigend A, Rumelhart D, Huberman B. 1990. Predicting the future: A connectionist approach. *International Journal of Neural Systems* **1**:193-209.
265. Widrow B, Stearns SD. 1985. *Adaptive Signal Processing*. Prentice-Hall, Englewood Cliffs, NJ.
266. Widrow B, Rumelhart DE, Lehr MA. 1994. Neural Networks: Applications in Industry, Business and Science. *Communications of the ACM* **37**(3):95-105.
267. Wikipedia. 2005. *Mathematical model*. [Available at http://en.wikipedia.org/wiki/Mathematical _model]. (Accessed on: 22.09.2005)
268. Williams GJ, Huang Z. 1997. Mining the Knowledge Mine - The Hot Spots Methodology for Mining Large Real World Databases. Sattar A. (Ed) *Advanced Topics in Artificial Intelligence*. Lecture Notes in Computer Science, Vol. 1342, Springer-Verlag: 340-348.
269. Witten I, Frank E. 2000. *Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations*, Morgan Kaufmann Publishers, 2000.
270. Wong BK, Selvi Y. 1998. Neural Networks Applications in Finance: A Review and Analysis of Literature (1990-1996). *Information & Management* **34**: 129-139.
271. Yao X, Liu Y. 1997. A new evolutionary system for evolving artificial neural networks. *IEEE Transactions on Neural Networks*, **8**(3):694-713.
272. Yao X. 1999. Evolving Artificial Neural Networks. *Proceedings of the IEEE*, **87**(9):1423-1447.
273. Yu DL, Williams D, Gomm JB. 2000. Online Implementation of a Neural Network Model Predictive Controller. Paper presented at *IEE Seminar – Practical Experiences with Predictive Control*, March 2000.
274. Zadeh LA. 1965. Fuzzy Sets*.* *Information and Control* **8**: 338-353.
275. Zavgren C. 1985. Assessing the vulnerability to failure of American industrial firms: A logistics analysis. *Journal of Business Finance and Accounting* (Spring): 19-45.
276. Zhang G, Patuwo BE, Hu MY. 1998. Forecasting with artificial neural networks: The state of the art. *International Journal of Forecasting* **14**(1): 35-62.
277. Zupan B, Bohanec M, Demsar J, Bratko I. 1998. Feature Transformation by Function Decomposition. *IEEE Intelligent Systems* **13**(2): 38-43.

Appendix

The selected countries:

No.	Country	Year	Monthly data
1	Russia	1994	quarterly averages
2	Russia	1995	quarterly averages
3	Russia	1996	quarterly averages
4	Russia	1997	1, ..., 12
5	Russia	1998	1, ..., 12
6	Russia	1999	1, ..., 12
7	Russia	2000	1, ..., 9
8	Ukraine	1993	10
9	Ukraine	1994	quarterly averages
10	Ukraine	1995	quarterly averages
11	Ukraine	1996	quarterly averages
12	Ukraine	1997	quarterly averages
13	Ukraine	1998	quarterly averages
14	Ukraine	1999	quarterly averages
15	Ukraine	2000	1, ..., 6
16	Latvia	1993	quarterly averages
17	Latvia	1994	quarterly averages
18	Latvia	1995	quarterly averages
19	Latvia	1996	quarterly averages
20	Latvia	1997	quarterly averages
21	Latvia	1998	quarterly averages
22	Latvia	1999	quarterly averages
23	Latvia	2000	1, ..., 10
24	Poland	1995	quarterly averages
25	Poland	1996	quarterly averages
26	Poland	1997	quarterly averages
27	Poland	1998	quarterly averages
28	Poland	1999	quarterly averages
29	Poland	2000	1, ..., 12
30	Slovenia	1993	1, ..., 12
31	Slovenia	1994	1, ..., 12
32	Slovenia	1995	1, ..., 12
33	Slovenia	1996	1, ..., 12
34	Slovenia	1997	1, ..., 8
35	Romania	1996	yearly average
36	Romania	1997	yearly average
37	Romania	1998	yearly average
38	Romania	1999	yearly average
39	Romania	2000	1, ..., 7

The selected pulp-and-paper companies:

Country	No.	Company	Years	Website
Finland	1	Average	1995-2000	
Finland	2	Ahlström	1995-2000	http://www.ahlstrom.com
Finland	3	Metsä-Serla OY	1995-2000	http://www.metsopaper.com
Finland	4	Stora Enso OY	1995-2000	http://www.storaenso.com
Finland	5	UPM-Kymmene OY	1995-2000	http://www.upm-kymmene.com
Sweden	6	Average	1995-2000	
Sweden	7	AssiDomän	1995-2000	http://www.asdo.se
Sweden	8	Korsnäs	1995-2000	http://www.korsnas.com
Sweden	9	MoDo AB	1995-2000	http://www.holmen.com
Sweden	10	Munskjö AB	1995-2000	http://www.munksjo.com
Sweden	11	Rottneros AB	1995-2000	http://www.rottneros.se
Sweden	12	SCA AB	1995-2000	http://www.sca.se
Sweden	13	Södra AB	1995-2000	http://www.sodra.se
Norway	14	Average	1995-2000	
Norway	15	Norske Skog A.S.	1995-2000	http://www.norskeskog.no
Norway	16	Peterson Group	1995-2000	http://www.peterson.no
USA	17	Average	1995-2000	
USA	18	Boise Cascade	1995-2000	http://www.boisecascade.com
USA	19	Bowater	1995-2000	http://www.bowater.com
USA	20	Buckeye Technologies	1995-2000	http://www.bkitech.com
USA	21	Caraustar Industries	1995-2000	http://www.caraustar.com
USA	22	Champion International	1995-1999	http://www.championinternational.com
USA	23	Consolidated Papers	1995-1999	http://www.consolidatedpapers.com
USA	24	Crown Vantage	1995-1999	http://www.crownvantage.com
USA	25	Fort James	1995-1999	http://www.fortjames.com
USA	26	Gaylord Container Corp	1995-2000	http://www.gaylordcontainer.com
USA	27	Georgia-Pacific Corp	1995-2000	http://www.gp.com
USA	28	International Paper	1995-2000	http://www.internationalpaper.com
USA	29	Jefferson-Smurfit Corp	1995-2000	http://www.smrfit.ie
USA	30	Kimberly-Clark	1995-2000	http://www.kimberly-clark.com
USA	31	Longview Fiber Corp	1995-2000	None
USA	32	Mead	1995-2000	http://www.mead.com
USA	33	P.H. Glatfelter	1995-2000	http://www.glatfelter.com
USA	34	Pope & Talbot	1995-2000	http://www.poptal.com
USA	35	Potlatch Corp	1995-2000	http://www.potlatchcorp.com
USA	36	Rayonier	1995-2000	http://www.rayonier.com
USA	37	Riverwood Holding	1995-2000	http://www.riverwood.com
USA	38	Rock-Tenn Company	1995-2000	http://www.rocktenn.com
USA	39	Schweitzer-Mauduit Intl.	1995-2000	None
USA	40	Sonoco Products	1995-2000	http://www.sonoco.com
USA	41	Stone Container	1995-1997	Part of Smurfit
USA	42	Temple-Inland	1995-2000	http://www.templeinland.com
USA	43	Union Camp.	1995-1998	Part of International Paper

USA	44	Wausau-Mosinee Paper	1995-2000	http://www.wausaumosinee.com
USA	45	Westvaco	1995-2000	http://www.westvaco.com
USA	46	Weyerhaeuser	1995-2000	http://www.weyerhaeuser.com
USA	47	Willamette Industries	1995-2000	http://www.wii.com
Canada	48	Average	1995-2000	
Canada	49	Abitibi Consolidated	1995-2000	http://www.abitibiconsolidated.com
Canada	50	Alliance	1995-2000	http://www.alliance-forest.com
Canada	51	Canfor	1995-2000	http://www.canfor.com
Canada	52	Cascades Inc.	1995-2000	http://www.cascades.com
Canada	53	Crestbrook Forest Ind. Ltd.	1995-1997	http://www.crestbrook.com
Canada	54	Doman Industries	1995-2000	http://www.domans.com
Canada	55	Domtar Inc.	1995-2000	http://www.domtar.com
Canada	56	Donohue	1995-1999	http://www.donohue.ca
Canada	57	MacMillan Bloedel	1995-1998	Part of Weyerhaeuser
Canada	58	Nexfor	1995-2000	http://www.nexfor.com
Canada	59	Tembec Inc.	1995-2000	http://www.tembec.ca
Canada	60	West Fraser Timber	1995-2000	http://www.westfrasertimber.com
Japan	61	Average	1995-1999	
Japan	62	Daio Paper	1995-1999	http://www.daio-paper.co.jp
Japan	63	Daishowa Paper Manuf	1995-1999	http://www.daishowa.co.jp
Japan	64	Chuetsu Paper	1995-1999	http://www.nsknet.or.jp/chupa
Japan	65	Hokuetsu Paper Mills	1995-1999	None
Japan	66	Japan Paperboard Industries	1995-1999	None
Japan	67	Mitsubishi Paper	1995-1999	None
Japan	68	Nippon Kakoh Seishi	1995-1999	None
Japan	69	Nippon Paper Industries	1995-2000	http://www.np-g.com/e/index.html
Japan	70	Oji Paper	1995-2000	http://www.ojipaper.co.jp
Japan	71	Pilot (Lintec)	1995-2000	http://www.lintec.co.jp
Japan	72	Rengo	1995-1999	http://www.rengo.co.jp
Japan	73	Settsu	1995-1998	None
Japan	74	Tokai Pulp & Paper	1995-1999	None
Europe	75	Average	1995-2000	
Spain	76	ENCE Group	1996-2000	http://www.ence.es/uk/el_grupo.htm
Austria	77	Frantschach	1995-1999	http://www.frantschach.com
Switzerland	78	Industrieholding Cham	1995-2000	http://www.iccham.com/index.htm
United Kingdom	79	Inveresk	1995-2000	http://www.inveresk.co.uk
Austria	80	Mayr-Melnhof	1995-2000	http://www.mayr-melnhof.co.at
Italy	81	Reno de Medici	1995-2000	http://www.renodemedici.it
Australia	82	Amcor	1995-2000	http://www.amcor.co.au
New Zealand	83	Fletcher Challenge Group	1995-1999	http://www.fcl.co.nz/home.asp
Italy	84	Cartiere Burgo	1995-2000	http://www.burgo.com

The selected telecom companies:

Country	No.	Company	Years	Website
Finland	1	Benefon	1995-2001	http://www.benefon.fi

Sweden	2	Doro	1995-2001	http://www.doro.com
Sweden	3	Ericsson	1995-2001	http://www.ericsson.com
Finland	4	Elisa Comm.	1995-2001	http://www.elisa.com
Norway	5	Netcom	1995-2001	http://www.netcom.no
Finland	6	Nokia	1995-2001	http://www.nokia.com
Finland	7	Sonera	1995-2001	http://www.sonera.com
Denmark	8	Tele Danmark	1995-2001	http://www.teledanmark.dk
Norway	9	Telenor	1995-2001	http://www.telenor.com
Sweden	10	Telia	1995-2001	http://www.telia.com
The Nordic	11	Average	1995-2001	
France	12	Alcatel	1995-2001	http://www.alcatel.com
Switzerland	13	Ascom	1995-2001	http://www.ascom.com
UK	14	British Telecom	1995-2001	http://www.bt.com
UK	15	Cable & Wireless	1995-2001	http://www.cwc.com
UK	16	Colt	1995-2001	http://www.colt-telecom.com
Germany	17	Deutsche Telekom	1995-2001	http://www.dtag.de
France	18	France Telecom	1995-2001	http://www.francetelecom.com
UK	19	Marconi	1995-2001	http://www.marconi.com
Hungary	20	Matav	1995-2001	http://www.matav.hu
Germany	21	Mobilcom	1996-2001	http://www.mobilcom.de
Italy	22	Olivetti	1995-2001	http://www.olivetti.com
UK	23	Orange	1995-2001	http://www.orange.co.uk
The Netherlands	24	Philips	1995-2001	http://www.philips.com
Portugal	25	Portugal Telecom	1995-2001	http://www.telecom.pt
Russia	26	Rostelecom	1995-2001	http://www.rostelecom.ru
France	27	Sagem	1996-2001	http://www.sagem.com
Germany	28	Siemens	1995-2001	http://www.siemens.com
Switzerland	29	Swisscom	1995-2001	http://www.swisscom.com
UK	30	TeleWest	1995-2001	http://www.telewest.co.uk
UK	31	Vodafone	1995-2001	http://www.vodafone.com
Europe	32	Average	1995-2001	
USA	33	3Com	1995-2001	http://www.3com.com
USA	34	ADC	1995-2001	http://www.adc.com
USA	35	Alltel	1995-2001	http://www.alltel.com
USA	36	Andrew Corp.	1995-2001	http://www.andrew.com
USA	37	AT&T	1995-2001	http://www.att.com
USA	38	Audiovox	1995-2001	http://www.audiovox.com
USA	39	Verizon	1995-2001	http://www.verizon.com
USA	40	BellSouth	1995-2001	http://www.bellsouth.com
USA	41	CenturyTel	1995-2001	http://www.centurytel.com
USA	42	Cisco	1995-2001	http://www.cisco.com
USA	43	Comsat	1995-1999	None
USA	44	Comverse	1995-2001	http://www.comverse.com
USA	45	Elcotel	1995-2001	http://www.elcotel.com
USA	46	GTE	1995-1999	http://www.gte.com
USA	47	IDT	1995-2001	http://www.idt.net

USA	48	Wireless Webcon.	1995-2001	http://www.intellicall.com
USA	49	Interdigital	1995-2001	http://www.interdigital.com
USA	50	LSI Logic	1995-2001	http://www.lsilogic.com
USA	51	Lucent	1995-2001	http://www.lucent.com
USA	52	MCI	1995-2001	http://www.wcom.com
USA	53	Molex	1995-2001	http://www.molex.com
USA	54	Motorola	1995-2001	http://www.motorola.com
USA	55	Nextel	1995-2001	http://www.nextel.com
USA	56	Powertel	1995-2001	http://www.powertel.com
USA	57	Powerwave	1995-2001	http://www.powerwave.com
USA	58	Qualcomm	1995-2001	http://www.qualcomm.com
USA	59	SBC	1995-2001	http://www.sbc.com
USA	60	Sprint	1995-2001	http://www.sprint.com
USA	61	Tellabs	1995-2001	http://www.tellabs.com
USA	62	Qwest	1995-2001	http://www.qwest.com
USA	63	Viatel	1995-2000	http://www.viatel.com
USA	64	Xircom	1995-2000	http://www.xircom.com
USA	65	Average	1995-2001	
Canada	66	Bell Mobility	1995-2001	http://www.bce.ca
Canada	67	Clearnet	1995-1999	http://www.clearnet.com
Canada	68	Zarlink	1995-2000	http://www.mitel.com
Canada	69	Nortel	1995-2001	http://www.nortel.com
Canada	70	Sasktel	1995-2001	http://www.sasktel.com
Canada	71	Telus	1995-2001	http://www.telus.ca
Canada	72	Average	1995-2001	
Japan	73	DDI	1995-2002	http://www.kddi.com
Indonesia	74	Indosat	1995-2001	http://www.indosat.com
Japan	75	Iwatsu	1995-1999	http://www.iwatsu.com
Japan	76	Japan Radio	1995-2002	http://www.jrc.co.jp
Japan	77	Japan Telecom	1995-2002	http://www.japan-telecom.co.jp
Japan	78	Kokusai	1995-1999	http://www.kokusaidenki.co.jp
Japan	79	Kyocera	1995-2002	http://www.kyocera.co.jp
Japan	80	Matsushita	1995-2001	http://www.panasonic.com
Japan	81	Mitsubishi Elec.	1995-2002	http://www.mitsubishi.com
Japan	82	NEC	1995-2002	http://www.nec.com
Japan	83	NTT DoCoMo	1995-2002	http://www.ntt.co.jp
Japan	84	OKI	1995-2002	http://www.oki.com
Japan	85	Samsung	1995-2001	http://www.samsung.com
Japan	86	Sanyo	1995-2002	http://www.sanyo.co.jp
Japan	87	Sharp	1995-2002	http://www.sharp-world.com
Japan	88	Sony	1995-2002	http://www.sony.net
Australia	89	Telstra	1995-2002	http://telstra.com
Japan	90	Toshiba	1995-2002	http://www.toshiba.com
Japan	91	Uniden	1995-2002	http://www.uniden.co.jp
India	92	Videsh Niagam	1996-2002	http://www.vsnl.net.in
Asia	93	Average	1995-2002	

PART TWO: ORIGINAL RESEARCH PUBLICATIONS

Publication 1

Kloptchenko A, Eklund T, Costea A, Back B. 2003. A Conceptual Model for a Multiagent Knowledge Building System. In *Proceedings of 5th International Conference on Enterprise Information Systems (ICEIS 2003)*, Camp O, Filipe J, Hammoudi S, Piattini M (eds.), published by Escola Superior de Tecnologia do Instituto Politécnico de Setúbal, Angers, France, April 23-26, 2003, Volume 2: Artificial Intelligence and Decision Support Systems, pp. 223-228. ISBN: 972-98816-1-8.

A CONCEPTUAL MODEL FOR A MULTIAGENT KNOWLEDGE BUILDING SYSTEM

Antonina Kloptchenko, Tomas Eklund, Adrian Costea, Barbro Back
Turku Centre for Computer Science and IAMSR / Åbo Akademi University, Lemminkäisenkatu 14 B, 20520 Turku, Finland
Email: akloptch@abo.fi, toeklund@abo.fi, acostea@abo.fi, bback@abo.fi

Keywords: Software agents and multiagent systems, financial analysis, data mining, text mining

Abstract: Financial decision makers are challenged by the access to massive amounts of both numeric and textual financial information made achievable by the Internet. They are in need of a tool that makes possible rapid and accurate analysis of both quantitative and qualitative information, in order to extract knowledge for decision making. In this paper we propose a conceptual model of a knowledge-building system for decision support based on a society of software agents, and data and text mining methods.

1 INTRODUCTION

A huge amount of electronic information concerning different companies' financial performance and market situation is available in various databases and on the Internet today. This information can potentially be very valuable to companies' decision makers, their partners, competitors, investors, analysts, and stakeholders. These individuals want to extract relevant information for decision-making purposes from the widely available data storages on time and, preferably, by the click of a mouse button. The enormous supply of data available often exceeds our capacity to analyze it, leading to information overload. Users need to transform new data into valuable knowledge very quickly in order to react to rapidly changing conditions and make crucial decisions in time.

Although there are a number of methods and technologies available for creating, storing, and monitoring new data, there are not very many comprehensive and popular techniques for transforming all data into valuable information and knowledge. The fields of *knowledge discovery in databases (KDD), data mining (DM), and text mining (TM)* have provided a number of new approaches for analysis of large databases of financial data. KDD is the entire process of discovering interesting knowledge, such as patterns, associations, changes and anomalies, and significant structures from large amounts of stored data, while DM refers to the actual use of data mining tools for identifying patterns in the data (Fayyad et al. 1996). Most data mining techniques for financial applications deal with quantitative data. The analysis of qualitative information (company strategy, economic market outlook, i.e. the textual parts of financial statements, as well as information from outside sources) is very important and can be done using text mining approaches. TM refers to the nontrivial extraction of implicit, previously unknown, and potentially useful information from large textual datasets (Dorre et al., 1999). Unlike numeric data, textual statements contain not only the factual event but also the explanation for why it happens (Wuthrich et al. 1998).

The individuals are fortunate if the valuable data that they need are already stored in one available database on the web. More often the data are located on a number of different sites. An emerging problem is how to find and collect these data and process them so that they provide additional valuable knowledge. The majority of data mining techniques are meant for extracting meaningful patterns from numeric, well-structured databases. At the same time, ambiguously structured text databases grow large in size and significance, and require effective text mining techniques. A multi-agent software system consisting of a collection of individual software agents, each of which provides a certain task (Lesser 1995) and/or uses different data mining techniques, can be a possible solution for accomplishing this task.

In this paper we create a conceptual model of a knowledge building system based on a society of software agents, and data and text mining methods. Each agent exhibits intelligence by using different data and text mining methods. We believe that software agents, which are able to execute tasks on behalf of a business process, computer application, or an individual, are well suited to dealing with collecting, processing, and compiling vast volumes of dynamic data from distributed sources. The system could monitor new financial updates from a variety of sources, and calculate financial ratios for different companies. These data could be used for various tasks, for example, financial benchmarking and assessing creditworthiness of different companies.

Our model suggests the integration of several computing techniques, namely self-organizing maps for clustering quantitative information, decision trees and/or multinomial logistic regression for classifying new cases into previously obtained clusters, prototype-matching for semantic clustering qualitative information, and various techniques for text summarization. We have previously tested some of the techniques in certain modules of the conceptual model.

The paper is organized as follows: In Section 2 we describe the problem area and the approaches used in financial data analysis for solving the discussed problems and provide an overview of literature and related work in multiagent system design. We describe the conceptual model of our multiagent decision support system in Section 3. We explain the methodological issues of the different computational techniques we propose in Section 4. We discuss the possible limitations and difficulties associated with building and using the proposed system in Section 5. Section 6 contains our conclusions and the directions of future work.

2 DESCRIPTION OF PROBLEM AREA AND RELATED WORK

Financial analysis is very important in today's global economy. Access to more information should be beneficial to any investor or financial stakeholder. Financial benchmarking is an important and valuable tool for assessing the actual financial performance of a company. Financial benchmarking is the process of comparing a number of competitors according to, most commonly, a number of financial ratios, chosen based on the motive for the benchmarking (for example, to compare profitability, efficiency, etc). This type of benchmarking is often external, and does not require the participation of the benchmarked companies. Indeed, financial benchmarking is often performed by consulting companies, or business or industry-specific journals (such as *Pulp and Paper International*). Financial benchmarking can also be used by individual investors seeking to evaluate the actual financial performance or state of an investment object in comparison to competing investment opportunities.

An assessment of the creditworthiness of debt-issuing companies is based on the financial statements of the issuer and on expectations of future economic development using a combination of qualitative and quantitative analysis (Tan et al. 2002). Credit rating agencies (e.g. Moody's Investor Services, Standard & Poor Corp., FLIP) are commercial firms that receive payment for publishing an evaluation of the creditworthiness of their clients. Creditworthiness information is especially useful when borrowing takes place through the issue of securities, rather than by bank loans, since buyers of securities do not know the issuers as well as banks usually know their customers.

The idea of a society of software agents was introduced in Wang et al. (2002) for monitoring and detection of financial risk. In a society of software agents each agent carries out different functions autonomously. We use a multiagent approach for building our knowledge creating system.

There have been a number of attempts to use multiagent systems to support business processes and deal with business environment. Liu (1998) suggested a software agent approach in environmental scanning activities for senior managers. An agent system developed by PriceWaterhouseCoopers, called EdgarScan, scans the financial reports in the Securities and Exchange Commission's database (EDGAR). The agent works by scanning the document for tags that indicate certain financial data. The system also includes a basic graphical benchmarking system, which only allows the user to compare companies by one ratio or value at a time. The agent can be found at http://www.pwcglobal.com/gx/eng/ins-sol/online-sol/edgarscan. Nelson et al. (2000) have proposed an auditing system (FRAANK) based on an agent that retrieves financial information from the EDGAR database.

One popular data mining technique for quantitative data analysis is the *self-organizing map (SOM)* (Kohonen 1997). The SOM has been used for a variety of tasks relating to financial analysis, for example, credit analysis (Martín del-Brío and Serrano-Cinca 1993; Back et al. 1995; Serrano-Cinca 1996; Kiviluoto 1998; Tan et al. 2002), financial benchmarking (Back et al. 1998; Karlsson et al. 2001; Eklund et al. 2002), and macro level economic environment analysis (Kaski and Kohonen 1996). Tan et al. (2002) studied the rating process using Self-organizing maps for clustering and visualizing the financial ratios. Lavrenko et al. (2000), Back et al. (2001), and Kloptchenko et al. (2002) have combined quantitative and qualitative financial data using quantitative and qualitative clustering techniques for knowledge discovery.

3 THE CONCEPTUAL MODEL

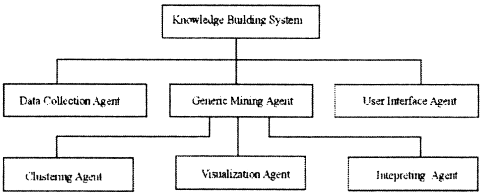


Figure 1: Architecture of the Knowledge Building System.

The proposed conceptual model of the knowledge-building system is depicted in Figure 1. It consists of six agents, i.e. *the Data Collection Agent, the Generic Mining Agent, the User Interface Agent, the Clustering Agent, Visualization Agent* and *the Interpreting Agent*. Each agent carries out its own functions and uses information provided by other agents connected to it. These agents handle three main activities (that are provided by three autonomous agents): data collection and storage (Data Collection Agent), searching for hidden patterns (Generic Mining Agent), and user-interface design (User Interface Agent).

The *Data Collection Agent* is intended to collect, assemble, and sort the quantitative and qualitative data from various Internet resources, such as Bloomberg, Reuters, Wall Street Journal, MSNBC, and individual companies' web sites. These data consist of, for example, market updates, quotes, financial reports, market reports, etc.

The *User Interface Agent* is intended to be responsible for providing the communication channel between the system and the human user that chooses the goal for the system. It should offer the choice of a number of tasks defined by the user in their setup of the system. For example, two possible applications are financial benchmarking and credit rating. These tasks are defined by the data (numeric and textual) included, as well as by the importance placed on each piece of data (for example, the importance of a particular financial ratio). In short, the agent should present the system options, receive the user input commands, and show the final results after it has interacted with the other agents.

The *Generic Mining Agent* is intended to include at least three activities in data processing (see Figure 1.): clustering of the data, visualization of the intermediary results of the previous process, and interpretation of the final results. The clustering techniques are instance dependent, in the sense that we can apply different clustering algorithms when performing data and text mining. We have three agents for the three distinct steps in data processing: the *Clustering Agent, the Visualization Agent*, and the *Interpretation Agent.*

Depending on what mining techniques and data are used, there are two main instances of the Generic Mining Agent: *Data Mining Agent* (Figure 2.) and *Text Mining Agent* (Figure 3.). We see the Generic Mining Agent as a generic class (in programming language understanding), which does not exist physically, but rather is an abstract class that is implemented via its instances. A distinction between the two instances of the Generic Mining Agent is based on the types of data they mine: Data Mining Agent (for processing numeric data) and Text Mining Agent (for processing text data).

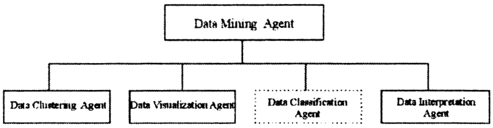


Figure 2: Data Mining Instance of the Generic Mining Agent.

In addition to the activities that are common for both the Data and Text Mining Agents, there are other activities that can be implemented, for example, constructing classification models in the case of the Data Mining Agent and information summarization for the Text Mining Agent. Two new agents can perform these two different activities: the *Data Classification Agent* (see Figure 2, dot-line rectangle) and the *Summarization Agent* (see Figure 3, dot-line rectangle).

The *Knowledge Building System* aims at creating new knowledge by consolidating the obtained new information from the Data Mining and Text Mining Agents. The Knowledge Building System will behave reactively to the goal of the system.

The *Data Mining Agent* would be responsible for numeric data processing and pattern discovery. The Data Mining Agent should provide the Knowledge Building System with the cluster that a company (or other data, depending upon the intended goal) belongs to, as well as the characteristics of the clusters (high profitability, low solvency, etc.), i.e. the results of the entire clustering. The *Data Clustering Agent* should calculate the chosen financial ratios for the chosen companies, standardize the data, and cluster them using self-organizing maps. Finally, the *Data Visualization Agent* visualizes the results.

After we visualize the map clusters provided by the Data Clustering Agent we could use the *Data Classification Agent* that creates a decision tree and/or a multinomial logistic regression model for classifying new financial data (Costea and Eklund 2003). The Data Classification Agent might also use other classifiers. Among these, the agent should use the model that achieves the highest accuracy in training and the best prediction performance.

Then, using all the information from the previous agents, combined with knowledge from other agents in the system, the *Data Interpretation Agent* would attempt to explain the findings. For example, in quantitative clustering, it is important to find explanations for a particular event, such as decreased profitability. This type of information can be found in the textual part of the annual report.

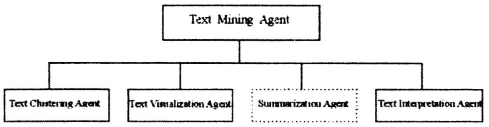


Figure 3: Text Mining Instance of the Generic Mining Agent.

The *Text Mining Agent* is intended to be responsible for processing textual information, and choosing the essential indications in it. It could use the *Summarization Agent* that deals with domain information, creating news summaries for any chosen company, or general market information for any chosen time period, and reports it to the user. Then, the *Text Clustering Agent* would perform financial statement clustering by using the prototype-matching methodology (Visa et al. 2002; Back et al. 2001) and reports which financial reports are close in meaning to each other. The *Text Visualization Agent* would present a visual U-matrix map with cluster representation and labels of the companies, which are clustered according to the similarity of their financial statements. The *Text Interpretation Agent* would have the same functionality as the Data Interpretation Agent, the difference being the type of data that is processed.

The Knowledge Building System combines the information from the Data and Text Mining Agents, i.e. it reports to the user how well the chosen company is performing in light of the chosen task, what level of performance the company displays in comparison with other companies in the analysis (clusters), and explains why (text summaries and clusters). The outputs of the two "instance" agents (Data and Text Mining Agents) can be validated one against the other, and the Knowledge Building System can do this automatically, alarming the user if the results are not convergent.

4 METHODS USED BY THE AGENTS

Our agents use several specific data mining techniques for clustering, visualization, and classification of quantitative and qualitative data.

We have used the SOM for clustering the quantitative data. The SOM is an unsupervised neural network for exploratory data analysis. The SOM takes multidimensional numeric data and clusters them on a two-dimensional topological map. Kiang and Kumar (2001) made a comparison between self-organizing maps and factor analysis and K-means clustering. The authors compared the tool's performances on simulated data, with known underlying factor and cluster structures. The results of the study indicate that self-organizing maps can be a robust alternative to traditional clustering methods.

Once trained SOM models are created, the problem of dealing with new data arises. Instead of time consuming retraining, a different method was proposed in Costea and Eklund (2003). The authors suggest a two-level methodology including initial clustering using SOM, and decision tree or multinomial logistic regression classification models trained on the original SOM model. This way the user is able to deal with new data without retraining maps. We have compared the two classification techniques in terms of their accuracy rates and class predictions and reached the conclusion that choosing among possible classifiers is problem dependent. We can extend the number of variables used for training the SOM maps, since the algorithm does not have restrictions from this point of view. Conversely, this methodology can be used as an alternative way of assessing the creditworthiness of companies as opposed to that provided by, say, Standard & Poor's (Tan et al. 2002).

We have tested the use of the prototype-matching approach for text clustering. This method is based on textual collection processing on word and sentence level processing (Visa et al. 2002; Toivonen et al. 2001). The prototype is a document, or a specific part of it, which is of interest to a particular user. A prototype is matched with an existing text collection to obtain a cluster of semantically similar documents. The methodology is based on text preprocessing, and word and sentence level text encoding and histogram creation.

The text summarization algorithm should extract the most relevant sentences from one or multiple documents with regard to a query. Therefore, we propose the use of a text clustering algorithm (e.g. prototype-matching or bisect k-means) for organizing one or more relevant documents into a tight cluster, and a feature extraction algorithm (e.g. occurrence of cue words, frequent words and proper nouns, position of the sentence with them in the text, sentence length, etc.) and classification algorithm (e.g. Naïve-Bayes classifier, C4.5) for extracting relevant sentences in the relevant documents. The combination of the mentioned techniques requires thorough study for successful summarization. We realize that straightforward word matching is not enough for effective detection of similarity between text pieces.

5 LIMITATIONS AND DIFFICULTIES

There are, of course, a number of problems associated with building a system of this complexity based on data that are freely presented on the Internet. We can divide system limitations in, at least, two categories: limitations that are specific for each individual agent and limitations regarding the integration of different agents. The data collection agent's ability to automatically retrieve financial data from Internet resources is severely hampered by a lack of standard for online financial reporting. A possible future solution to this problem is XBRL (eXtensible Business Reporting Language). XBRL is an XML (eXtensible Markup Language) standard created specifically to address the problem of online business reporting. Currently, there is no way for collection agents to automatically retrieve financial data from diverse web sites without specifically coding the agent for a specific page. (Debreceny and Gray 2001)

Another type of limitation of the system is due to the limitations of the deployed DM and TM techniques (Data and Text Mining Agents). For example, with all its advantages over standard clustering techniques, the SOM has one major drawback: verification of the achieved clustering results. This issue is addressed in Wang (2001), in which the author proposes a number of techniques for verifying clustering results. Similar techniques will have to be used in the system we are proposing.

Text mining techniques have a number of disadvantages due to the highly dimensional structure of text. Two textual pieces can often be nearest neighbors in terms of using similar vocabulary, without actually belonging to the same semantic class. Prototype-matching clustering is an exploratory technique that possesses some difficulties with determination of the clusters, and with their comparison with quantitative clustering. Although, theoretically, text implies richer information about an event than a numerical snapshot of the fact does, this is difficult to verify. Even having excellent text mining techniques on hand that could mine the indications of future financial performances of the company, those indications can be easily concealed by smart word choice and sentence construction.

Also, as was illustrated by the Enron and WorldCom scandals, the financial information presented in annual reports is not always reliable. Of course, if this incorrect information is inserted into our system, the results will also be incorrect. Moreover, there might be unintentional mistakes in the data. Therefore, some kind of error detection and handling capabilities should be built into the system. This is also required by the actual definition of KDD, which includes data cleaning and error detection (Fayyad et al. 1996).

The integration limitations are closely related to the individual agents limitations, e.g.: because of the lack of standard of financial information available on the Internet, the Data Collection Agent might not be able to provide the data that we need to address a specific problem, which makes its integration with the Knowledge Building System extremely difficult.

6 CONCLUSIONS AND FUTURE WORK

In the current research paper we introduced a conceptual model of a system based on different data/text mining methods for knowledge building from freely available data distributed on the web. The system aims to automatically perform different tasks such as data collection, financial benchmarking, assessing creditworthiness of companies, and finding hidden patterns in unordered and unstructured text data. The system uses two types of data (numeric and textual) and data processing techniques (data and text mining techniques) to support and explain the phenomena.

In this paper we discussed the operational facilities of the proposed system that will be accomplished by text and data mining methods. The system knowledge base, system external interface and limitations should be researched further.

As further research problems we could investigate new methods for collecting the input information for the Data and Text Mining Agents (that is improve the Data Collection Agent), extend the conceptual model to include subagents that perform tasks for their "parent" agents: Data Cleaning Agent, Data Aggregator Agent (aggregates information find on different web sources and presents this information further to Data Collection Agent).

REFERENCES

Back, B., G. Oosterom, K. Sere, m. van Wezel, 1995. Intelligent Information Systems within Business:Bankruptcy Predictions Using Neural Networks. In *The 3rd European Conference on Information Systems (ECIS'95)*, Athens, Greece.

Back, B., K. Sere, H. Vanharanta, 1998. Managing complexity in large data bases using self-orginizing maps. In *Accounting Management and Information Technologies* 8(4): 191-210.

Back, B., J. Toivonen, H. Vanharanta, A. Visa, 2001. Comparing numerical data and text information from annual reports using self-orginizing maps. In *International Journal of Accounting Information Systems* 2: 249-269.

Costea, A. and T. Eklund, 2003. A Two-Level Approach to Making Class Predictions. In *The 36th Hawaii International Conference on Systems Sciences (HICSS-36)*, Hawaii, USA, IEEE.

Debreceny, R. and G. L. Gray, 2001. The production and use of semantically rich accounting reports on the Internet: XML and XBRL. In *International Journal of Accounting Information Systems* 2(1): 47-74.

Dorre, J., Gerstl, P., and R. Seiffert, 1999, Text Mining: Finding Nuggets in Mountains of Textual Data, In *Proceedings of th 5th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, San Diego, California, USA

Eklund, T., B. Back, H. Vanharanta, A. Visa, 2002. Assessing the Feasibility of Self-Organizing Maps for Data Mining Financial Information. In *The Xth European Conference on Information Systems (ECIS 2002)*, Gdansk, Poland.

Fayyad, U., G. Piatetsky-Shapiro, P. Smythe, 1996. Knowledge Discovery and Data Mining: Towards a Unifying Framework. In *The Second International Conference on Knowledge Discovery and Data Mining (KDD-96)*, Portland, Oregon, AAAI Press.

Karlsson, J., B. Back, H. Vanharanta, A. Visa, 2001. *Financial Benchmarking of Telecommunications Companies*. TUCS Technical Report No. 395, Turku Centre for Computer Science. Turku.

Kaski, S. and T. Kohonen, 1996. Exploratory Data Analysis by the Self-Organizing Map: Structures of Welfare and Poverty in the World. In *The Third International Conference on Neural Networks in the Capital Markets*, World Scientific.

Kiang, M. and A. Kumar, 2001. An Evaluation of Self-Organizing Map Networks as a Robust Alternative to Factor Analysis in Data Mining Applications. In *Information Systems Research* 12(2): 34-41.

Kiviluoto, K., 1998. Predicting bankruptcies with the self-organizing map. In *Neurocomputing* 21(1-3): 191-201.

Kloptchenko A., T. Eklund., B. Back, J. Karlsson, H. Vanharanta, A. Visa, 2002. Combining Data and Text Mining Techniques for Analyzing Financial Reports. In *The 8th Americas Conference on Information Systems (AMCIS2002)*, Dallas, USA.

Kohonen, T., 1997. *Self-Organizing Maps*, Springer-Verlag. Leipzig, 2nd edition.

Lavrenko, V., M. Schmill, D. Lawrie, P. Ogilvie, 2000. Mining of Concurrent Text and Time Series. In *Text Mining Workshop of the 6th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Boston, MA, USA. ACM.

Lesser, V., 1995. Multiagent Systems: An emerging subdiscipline of AI. In *ACM Computing Surveys* 27(3): 340-342.

Liu, S., 1998. Business Environment Scanner for Senior Managers: Towards Active Executive Support with Intelligent Agents. In *Expert Systems with Applications* 15: 111-121.

Martín-del-Brío, B. and. C. Serrano-Cinca, 1993. Self-organizing Neural Networks for the Analysis and Representation of Data: Some Financial Cases. In *Neural Computing and Applications* 1: 193-206.

Nelson, K. M., A. Kogan, R. P. Srivastava, M. A. Vasarhelyi, H. Lu, 2000. Virtual auditing agents: the EDGAR Agent challenge. In *Decision Support Systems* 28(3): 241-253.

Serrano-Cinca, C., 1996. Self organizing neural networks for financial diagnosis. In *Decision Support Systems* 17(3): 227-238.

Tan, R., J. den Berg, W. den Bergh, 2002. Credit Rating Classification Using Self-Organizing Maps. In *Neural Networks in Business: Techniques and Applications*, ed. by K. Smith and J. Gupta, Idea Group Publishing. Hershey.

Toivonen, J., A. Visa, H. Vanharanta, B. Back, 2001. Validation of Text Clustering Based on Document Contents. In *Machine Learning and Data Mining in Pattern Recognition (MLDM 2001)*, Leipzig, Germany. Springer-Verlag.

Visa, A., J. Toivonen, B. Back, H. Vanharanta, 2002. Contents Matching Defined by Prototypes: Methodology Verification with Books of the Bible. In *Journal of Management Information Systems* 18(4): 87-100.

Wang, S., 2001. Cluster Analysis Using a Validated Self-Organizing Method: Cases of Problem Identification. In *International Journal of Intelligent Systems in Accounting, Finance and Management* 10(2): 127-138.

Wang, H., J. Mylopoulos, S. Liao, 2002. Intelligent Agents and Financial Risk Monitoring Systems. In *Communications of the ACM* 45(3): 83-88.

Wuthrich, B., D. Permunetilleke, S. Leung, V. Cho, J. Zhang, W. Lam, 1998. Daily Prediction of Major Stock Indices from textual WWW data. In *Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining (KDD-98)*, New York, NY, USA, AAAI Press.

Publication 2

Costea A, Eklund T. 2003. A Two-Level Approach to Making Class Predictions. In *Proceedings of 36th Annual Hawaii International Conference on System Sciences (HICSS 2003)*, Sprague Jr RH. (ed.), IEEE Computer Society, Hawaii, USA, January 6-9, 2003, Track: Decision Technologies for Management, Minitrack: Intelligent Systems and Soft Computing. ISBN: 0-7695-1874-5/03.

A Two-Level Approach to Making Class Predictions

Adrian Costea
Turku Centre for Computer Science and IAMSR / Åbo Akademi University, Turku, Finland, Adrian.Costea@abo.fi

Tomas Eklund
Turku Centre for Computer Science and IAMSR / Åbo Akademi University, Turku, Finland, Tomas.Eklund@abo.fi

Abstract

In this paper we propose a new two-level methodology for assessing countries'/companies' economic/financial performance. The methodology is based on two major techniques of grouping data: cluster analysis and predictive classification models. First we use cluster analysis in terms of self-organizing maps to find possible clusters in data in terms of economic/financial performance. We then interpret the maps and define outcome values (classes) for each data row. Lastly we build classifiers using two different predictive models (multinomial logistic regression and decision trees) and compare the accuracy of these models. Our findings claim that the results of the two classification techniques are similar in terms of accuracy rate and class predictions. Furthermore, we focus our efforts on understanding the decision process corresponding to the two predictive models. Moreover, we claim that our methodology, if correctly implemented, extends the applicability of the self-organizing map for clustering of financial data, and thereby, for financial analysis.

1. Introduction

In this study, we are interested in the relationship between a number of macro/microeconomic indicators of countries/companies and different economic/financial performance classifications. We have based our research on two previous studies [2] and [3]. In [2] we compared two different methods of clustering central-east European countries economic data (self-organizing maps and statistical clustering) and presented the advantages and disadvantages of each method. In [3], the self-organizing map (SOM) was used for benchmarking international pulp and paper companies. In both previous studies we were mainly concerned with finding patterns in economic/financial data and presenting this multi-dimensional data in an easy-to-read format (using SOM maps). However, we have not addressed the problem of class prediction as new cases are added to our datasets. From our previous results we cannot directly infer a procedure with which a new data row could be fit into our maps. As we obtain new data, depending upon the standardization technique used, we may be forced to retrain the maps, and repeat the entire clustering process. This is very time consuming, and requires the effort of an experienced SOM user. As Witten & Frank say in their book on data mining: "The success of clustering is measured subjectively in terms of how useful the result appears to be to a human user. It may be followed by a second step of classification learning where rules are learned that give an intelligible description of how new instances should be placed into the clusters." [17, p.39]

Here we propose a methodology that enables us to model the relationship between economic/financial variables and different classifications of countries/companies in terms of their performances. Defining the model permits us to predict the class (cluster) to which a new case belongs. In other words, we insert new data into our model and identify where they fit in the previously constructed map. Choosing the best technique for these two phases of our analysis (clustering/benchmarking/visualization and class prediction) is not a trivial task. In the literature there is a large number of techniques for both clustering and class prediction.

In this study, we use SOM as the clustering technique due to the advantages of good visualization and reduced computational cost. Even with a relatively small number of samples, many clustering algorithms – especially hierarchical ones (for example, Unweighted Pair Group Method with Arithmetic Mean (UPGMA), Ward's, or other bottom-up hierarchical clustering methods) – become intractably heavy [16].

Descriptive techniques, such as clustering, simply summarize data in convenient ways, or in ways that we hope will lead to increased understanding. In contrast, predictive techniques, such as multinomial logistic regression and decision trees, allow us to predict the probability that data rows will be clustered in a specific class in the trained SOM model. In order to find the predictive technique that is most suitable in our particular case, we conduct two experiments using multinomial logistic regression and decision tree techniques. When building real classifiers one can use three different fundamental approaches: the *discriminative approach*, the *regression approach*, and the *class-conditional approach* [6, p.335]. We chose to compare two regression approach methods: *multinomial logistic regression* and *decision trees*.

The rest of the paper is structured as follows. In Section two we present our methodology. In Section three, the datasets are presented and SOM clustering is performed. In Sections four and five, the multinomial regression and decision tree models are built and validated, and in Section six the models are compared. Finally, in Section seven, we present our conclusions.

2. Methodology

In our two-level approach we add another level (class prediction phase) to SOM clustering, as is depicted in Figure 1 (the arrows are the levels):

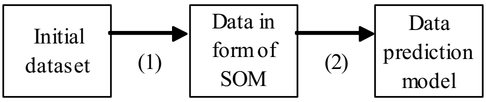


Figure 1. Two-level methodology

(1) – consists of several stages: preprocessing of initial data, training using the SOM algorithm, choosing the best maps, identifying the clusters, and attaching outcome values to each data row; [1]

(2) – depending on the technique that we apply, there can be different stages for this methodology level. When applying statistical techniques, such as multinomial logistic regression, we follow these steps: developing the analysis plan, estimation of logistic regression, assessing model fit (accuracy), interpreting the results, and validating the model. When applying the decision tree algorithm: constructing a decision tree step by step including one attribute at a time in the model, assessing model accuracy, interpreting the results, and validating the model.

After the predictive models for classification were constructed we compared them, based on their accuracy measures. Quinlan [10] states that there are different ways of comparing models besides their accuracy, e.g. the insight provided by the predictive model. However, we will use the accuracy measure since the example above is a subjective measure.

3. Clustering Using SOM

The SOM algorithm stands for self-organizing map algorithm, and is based on a two-layer neural network using the unsupervised learning method. The self-organizing map technique creates a two-dimensional map from n-dimensional input data. This map resembles a landscape in which it is possible to identify borders that define different clusters [8]. These clusters consist of input variables with similar characteristics, i.e. in this report, of countries/companies with similar economic/financial performance. The methodology used when applying the self-organizing map is as follows [1]. First, we choose the data material. It is often advisable to standardize the input data so that the learning task of the network becomes easier [8]. After this, we choose the *network topology*, *learning rate*, and *neighborhood radius*. Then, the network is constructed. The construction process takes place by showing the input data to the network iteratively using the same input vector many times, the so-called *training length*. The process ends when the *average quantization error* is small enough. The best map is chosen for further analysis. Finally, we identify the clusters using the *U-matrix* and interpret the clusters (assign labels to them) using the *feature planes*. From the feature planes we can read per input variable per neuron the value of the variable associated with each neuron.

The network topology refers to the form of the lattice. There are two commonly used lattices, *rectangular* and *hexagonal*. The hexagonal lattice is preferable for visualization purposes as it has six neighbors, as opposed to four for the rectangular lattice [8]. The learning rate refers to how much the winning input data vector affects the surrounding network. The neighborhood radius refers to how much of the surrounding network is affected. The average quantization error indicates the average distance between the best matching units and the input data vectors. Generally speaking, a lower quantization error indicates a better-trained map.

The sample data size is not of a major concern when using SOM algorithm. In [15] the author claims that SOM is easily applicable to small data sets (less than 10000 records) but can also be applied in case of medium sized data sets.

To visualize the final self-organizing map we use the unified distance matrix method (U-matrix). The U-matrix method can be used to discover otherwise invisible relationships in a high-dimensional data space. It also makes it possible to classify data sets into clusters of similar values. The simplest U-matrix method is to calculate the distances between neighboring neurons, and store them in a matrix, i.e. the output map, which then can be interpreted. If there are “walls” between the neurons, the neighboring weights are distant, i.e. the values differ significantly. The distance values can also be displayed in color when the U-matrix is visualized. Hence, dark colors represent great distances while brighter colors indicate similarities amongst the neurons. [14]

3.1. Datasets

In this study we have used two datasets from our previous papers: one dataset on the general economic performance (EconomicPerf) of the central-east-European countries [2] and another (FinancialPerf) on the financial performance of international pulp and paper companies [3]. The variables for the first dataset are:

- Currency Value, or how much money one can buy with 1000 USD, depicts the purchasing power of each country's currency (the greater the better),
- Domestic Prime Rate (Refinancing Rate), which shows financial performance and level of investment opportunities (the smaller the better),
- Industrial Output in percentages to the previous periods, to depict industrial economical development (the greater the better),
- Unemployment Rate, which characterizes the social situation in the country (the smaller the better), and
- Foreign Trade in millions of US dollars, to reveal the deficit/surplus of the trade budget (the greater the better).

In [2] there were two more variables in the dataset: import and export in million USD, as intermediary measures to calculate the foreign trade. We did not take them into account here, since they are strongly correlated with the foreign trade variable. Also, we have replaced the first variable (Foreign Exchange Rate) from the previous study [2] with Currency Value, which is calculated from the Foreign Exchange Rate variable by reversing it and multiplying the result with 1000. We have changed this variable to ensure the comparability among different countries' currencies.

Our dataset contains monthly/annual data for six countries (Russia, Ukraine, Romania, Poland, Slovenia and Latvia) during 1993-2000, in total 225 cases with five variables each. We have in some cases encountered lack of data, which we have completed using means of existing values. However, the self-organizing map algorithm can treat the problem of missing data simply by considering at each learning step only those indicators that are available [7].

The second dataset consisted of financial data on international pulp and paper companies. The dataset covered the period 1995-2000, and consisted of seven financial ratios per year for each company. The ratios were chosen from an empirical study by Lehtinen [9], in which a number of financial ratios were evaluated concerning their validity and reliability in an international context. The ratios chosen were:

- Operating margin, a profitability ratio,
- Return on Equity, a profitability ratio,
- Return on Total Assets, a profitability ratio,
- Quick Ratio, a liquidity ratio,
- Equity to Capital, a solvency ratio,
- Interest Coverage, a solvency ratio, and
- Receivables Turnover, an efficiency ratio.

The ratios were calculated based on information from the companies' annual reports. The dataset consisted of 77 companies and 7 regional averages. The companies were chosen from Pulp and Paper International's annual ranking of pulp and paper companies according to net sales [12]. In total, the dataset consisted of 474 rows of data.

3.2. Choosing the Best Maps

The two datasets were standardized according to different methods. In [2] the authors used the standard deviations of each variable to standardize the data (Equations 1, 2), while in [3] the data have been scaled using histogram equalization [4]. It is not our intention to describe different methods for the standardization of datasets; however, in the literature there are examples of both standardization techniques used on similar datasets.

$$x_i = \frac{\sum_{j=1}^{n} x_{ij}}{n} \qquad \text{[Eq. 1]}$$

$$\sigma_i = \sqrt{\frac{\sum_{j=1}^{n} (x_{ij} - \overline{x}_i)^2}{n}} \qquad \text{[Eq. 2]}$$

We have trained different maps with different parameters. As is stated in [2] a "good" map is obtained after several different training sessions. Best maps have been chosen based on two measures: one objective measure (the quantization error) and a subjective measure (ease of readability). However, the algorithm quantization error seems to be positively correlated with the dimension of the maps, while ease of readability is negatively correlated. In other words, we can obtain very "good" maps in terms of their quantization error if we use large dimension parameters, while they are poor in terms of readability. Cluster analysis is often a trade-off between accuracy and cluster clarity and manageability, by creating small maps we force the data into larger clusters. Consequently, when we compared the maps we restricted the maps' dimensions to be constant. The chosen maps and their clusters are presented in Figure 1.

3.3. Identifying the Clusters

We identify the clusters on the maps by studying the final U-matrix maps (Figure 1), the feature planes, and at the same time, by looking at the row data. Actually, the title of this paragraph, "identifying the clusters", should be "identifying the clusters of clusters". What we are saying is that we already have the clusters identified by SOM on the map (from now on we will refer to these clusters as row clusters). For example, in case we are using a 7x5 map, we have 35 row clusters. Next we have to identify the "real" clusters by grouping the row clusters. SOM helps us in this respect by drawing darker lines between two clusters that are "far" from each other (in terms of the Euclidean distance). The results for both datasets were very similar in terms of the amount, and characteristics, of clusters (7 in each case).

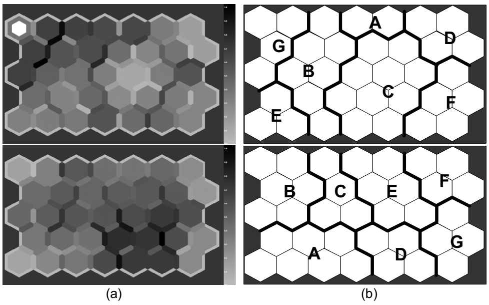


Figure 2. (a) The final U-matrix maps and (b) identified clusters on the maps for the EconomicPerf and FinancialPerf data sets

3.4. Defining the Outcome Values for each Row Data

Roughly speaking, we can state that the outcome values (the classes) in terms of economic/financial performance, were the same in both cases (Figure 1), so the classes are as follows:

A – best performance,
B – slightly below best performance,
C – slightly above average performance,
D – average,
E – slightly below average performance,
F – slightly above poorest performance, and
G – poorest performance.

Defining the outcome values for each data row is a straightforward process. Once we figure out which cluster each row cluster belongs to, the next step is to check which row data vectors are associated with each row cluster, and to associate the class code with those vectors. Consequently, in terms of methodology, we can divide the clustering process into two parts:

- creating the row-clusters – this part is entirely done by the SOM algorithm, the output being the U-matrix;
- creating the “real” clusters – this part is done by the map reader with the help of the SOM algorithm in terms of visualization characteristics.

This kind of multi-level clustering approach is not new. A two-level SOM clustering approach has been suggested before, in [16]. There, the row-clusters are “protoclusters” and our “real” clusters are the “actual” clusters. However, sometimes it is difficult to find good “real” clusters since the second part of the clustering process is highly subjective. Also, the standardization method has an important role, since for different standardization techniques we obtain different maps in terms of quantization error and ease of readability.

4. Applying multinomial logistic regression

In general, when multinomial logistic regression is applied as a predictive modeling technique for classification, there are some steps that have to be followed:

1. Check the requirements regarding the data sample: size, missing data, etc.,
2. Compute the multinomial logistic regression using an available software program (e.g. SPSS),
3. Assess the model fit (accuracy),
4. Interpret the results, and
5. Validate the model.

Below, we follow this methodology when applying logistic regression on our datasets.

4.1. Requirements

In the EconomicPerf dataset, the problem of missing data was overcome by using monthly means for each year. Averages were also used for missing data in the FinancialPerf dataset. The requirement of size, 15-20 cases for each independent variable, was exceeded for each dataset.

4.2. Computing the Multinomial Regression Model

We use SPSS to perform multinomial regression analysis selecting as dependent variables the class variables and as covariates the variables presented in Section 3.1.

4.3. Assessing the Model Fit

From the “Model Fitting information” output table of SPSS we observe that the chi-square value has a significance of < 0.0001, so we state that there is a strong relationship between dependent and independent variables (see Table 2). Next, we study the “Pseudo R-Square” table in SPSS, which also indicates the strength between dependent and independent variables. A good model fit is indicated by higher values. We will base our analysis on the Nagelkerke R^2 indicator (see Table 2). According to this, 74.5% for the EconomicPerf dataset and 97.8% for the FinancialPerf dataset, of the output variation can be explained by variations in input variables. Consequently, we would appreciate the relationships as very strong.

To evaluate the accuracy of the model, we compute the proportional by chance accuracy rate and the maximum by chance accuracy rate. The proportional chance criterion for assessing model fit is calculated by summing the squared proportion of each group in the sample, and the maximum chance criterion is the proportion of cases in the largest group. We obtained the following indicators (Table 1):

Table 1. Evaluate the model's accuracy

	Model	Proportional by chance criterion	Maximum by chance criterion
EconomicPerf	61,3%	29,92%	49,8%
FinancialPerf	88%	15,62%	20,46%

We interpret these numbers as follows: for example, in the case of the EconomicPerf dataset, based on the requirement that the model accuracy should be 25% better than the chance criteria [5, p. 89-90], the standard to use for comparing the model's accuracy is 1.25 x 0.2992 = 0.374. Our model accuracy rate of 61.3% exceeds this standard. The maximum chance criterion accuracy rate is 49.8% for this dataset. Based on the requirement that model accuracy should be 25% better than the chance criteria, the standard to use for comparing the model's accuracy is 1.25 x 49.8% = 62.22%. Our model accuracy rate of 61.3% is slightly below this standard. The FinancialPerf dataset accuracy rate exceeds both standards.

4.4. Interpreting the Results

To interpret the results of our analysis, we study the "Likelihood Ratio Test" and "Parameter Estimates" outputs of SPSS. We find that the independent variables are all significant, in other words they contribute significantly to explaining differences in performance classification (for both datasets). However, not all variables play an important role in all regression equations (e.g. for the first regression equation, "CurrencyValue" is not statistically significant 0,125 > p = 0,05). Next, we can determine the direction of the relationship and the contribution to performance classification of each independent variable by looking at columns "B" and "exp(B)" from the "Parameter Estimates" output of SPSS. For example, a higher industrial output rate increases the likelihood that the country will be classified as a best country (B = +24,027) and decreases the likelihood that the country will be classified among the poorest countries (B = -11,137). It seems that the results for the EconomicPerf dataset are poorer, in the sense that for the FinancialPerf dataset we have more coefficients estimates that are statistically significant. For example, if we study the "Parameter Estimates" outputs of SPSS ("Sig." column), we find that EconomicPerf dataset has 33% significant coefficients, while FinancialPerf dataset has 62.5%.

4.5. Validating the Model

In order to validate the model, we split the datasets in two parts of, approximately, the same length. Our findings are illustrated in Table 2:

Table 2. Datasets' accuracy rates and accuracy rates estimators when applying multinomial logistic regression

		Main dataset	Part1 (split=0)	Part2 (spli=1)
EconomicPerf	Model Chi-Square ($p < 0{,}0001$)	291,420	200,779	136,852
	Nagelkerke R^2	0,745	0,855	0,721
	Learning Sample	61,3%	67%	58,4%
	Test Sample	no test sample	57,6%	67,1%
	Significant coefficients ($p<0{,}05$)	ALL	ALL except: CURRENCY[1]	ALL
FinancialPerf	Model Chi-Square ($p < 0{,}0001$)	1479,72	792,06	752,85
	Nagelkerke R^2	0,978	0,986	0,981
	Learning Sample	88%	89%	89,5%
	Test Sample	no test sample	76,1%	82,4%
	Significant coefficients ($p<0{,}001$)	ALL	ALL	ALL

With one exception, we obtained significant coefficients for the logistic regression equations. In both cases, the accuracy rates of the two split datasets were close to the accuracy rate of the entire dataset. For example, 89% and 89,5% are close to the entire FinacialPerf dataset accuracy rate of 88%. Again, the second dataset outperformed the first one, in the sense that for the FinancialPerf dataset, the accuracy rates for the test samples are closer to the learning sample accuracy rate. However, more investigations should be done to find problems that arise due to insignificant coefficients of each regression equation. Large standard errors for "B" coefficients can be caused by multicollinearity among independent variables, which is not directly handled by SPSS or other statistical packages. Moreover, the problem of outliers and variable selection should be carefully addressed. Also, the discrepancies between learning and test accuracy rates can arise due to the small sizes of the datasets. The larger the dataset is, the better the chance that we have correctly clustered data and, consequently, correct outcome values for each data row. We construct the outcome values based on SOM clustering. There is, of course, a chance that there are misclustered data, which can affect the accuracy of the model.

4.6. Predicting the Classes

The finished model was then used to test the classification of three new data rows for the FinancialPerf

[1] this coefficients is significant for $p < 0{,}153$.

dataset. These consisted of data for three Finnish pulp and paper companies: M-Real (no. 3), Stora Enso (no. 4), and UPM-Kymmene (no. 5), for the year 2001. These were used since they were among the first to publish their financial results. The results are illustrated in Table 3.

Table 3. Predictions using multinomial logistic regression

Operating Margin	ROE	ROTA	Equity to Capital	Quick Ratio	Interest Coverage	Receivables Turnover	Company no.	**Predicted Cluster**
5.621597	17.75955	8.979317	27.02372	0.857129	2.314056	6.8226657	3	**D**
11.0069	15.31568	7.67552	31.23215	0.830754	4.189956	6.2295596	4	**B**
16.27344	22.78149	11.16978	34.59247	0.629825	5.205047	6.0291793	5	**A**

5. Applying the Decision Tree Algorithm

For comparison reasons, a See5 decision tree builder system was applied on both datasets. The system was developed by a research team headed by Quinlan. The algorithm behind the program is based on one of the most popular decision tree algorithms, and was developed in the late 70's, also by Quinlan: ID3 [11]. The main idea is that, at each step, the algorithm tries to select a variable and a value associated with it that discriminate "best" the dataset, and does this recursively for each subset until all the cases from all subsets belong to a certain class. The method is called "Top-Down Induction Of Decision Trees (TDIDT)" and C4.5, C5.0/See5 represent different implementations of this method. The "best" discriminating pair (variable-value) is chosen based on so-called "*gain ratio*" criterion:

gain ratio(X) = gain(X) / split info(X) [Eq. 3]

where gain(X) means the information gained by splitting the data using the test X and:

$$\text{split info (X)} = -\sum_{i=1}^{n} \frac{|S_i|}{|S|} \times \log_2\left(\frac{|S_i|}{|S|}\right) \quad \text{[Eq. 4]}$$

represents the potential information generated by dividing S into n subsets. The See5 system implements these formulas along with some other features that are described in [11] and on the web page http://www.rulequest.com/see5-info.html.

5.1. Computing the Decision Tree

For both datasets, we performed three runs of the See5 software, exactly like we did when applying logistic regression: one for the whole dataset, another using first split dataset ("split=0"), and the other using the second half of data ("split=1"). When validating the entire dataset accuracy rate, we have used cross-validation, while when validating one split dataset accuracy rate we have used the other one as test sample. The results are summarized in Table 4.

The first line, for each dataset, represents the accuracy rates obtained using training datasets. The next two lines show us the validation accuracy rates calculated as follows: for the main dataset a 10-crossvalidation was conducted (64% being the average accuracy rate of 10 decision trees), for the "split=0" dataset we used "split=1" as test dataset (46,9% is the accuracy rate on the second dataset, based on the decision tree built with the first dataset), and the last accuracy rate was calculated by considering "split=1" as the training dataset and "split=0" as the test dataset (changing the roles).

Table 4. Dataset accuracy rates and accuracy rates estimators when applying decision tree algorithm

		Main dataset	Part1	Part2
EconomicPerf	Learning Sample	79,1%	77,7%	78,86%
	Test Sample	no test sample	46,9%	54,5%
	cross-validation	64%	no cross-validation	no cross-validation
FinancialPerf	Learning Sample	84,8%	86,5%	86,5%
	Test Sample	74,6%	71,7%	76,8%
	cross-validation	74,4%	no cross-validation	no cross-validation

When constructing the trees, we kept the two most important parameters constant: m = 5, which measures the minimum number of cases each leaf-node should have, and c = 25% (default value) that is a confidence factor used in pruning the tree.

5.2. Assessing the Model Fit

For the EconomicPerf dataset, it seems that our trees were not consistent due to poor accuracy rates and big discrepancies between learning and test accuracy rates, so further comparison with regression analysis cannot be performed in this case. There is at least a 10% difference between the accuracy rates for each split dataset used.

For the FinancialPerf dataset, the differences between accuracy rates are smaller. Therefore, we used this dataset for further investigation. The chosen decision tree is presented in the Appendix. Reading it we can state that the main attribute used to discriminate the data was ROE. The lower that we go down in the decision tree, the less important the attributes become. At each step the algorithm calculates the information gain for each attribute choosing the split attribute with the largest information gain – we call it *the most important* attribute.

5.3. Interpreting the Results

As we can see from the decision tree (Appendix), the second most important variable depends upon the values of ROE: if our ROE is greater than or equal to 10.71424, it is Equity to Capital, while if ROE is less than or equal to 9.179343, it is Receivables Turnover. We must note that we have used fuzzy thresholds, which allows for a much more flexible decision tree: the algorithm (C5.0) assigns a lower value (*lv*) and an upper value (*uv*) for each attribute chosen to split the data. Then a membership function (trapezoidal) is used to decide which branch of the tree will be followed when a new case has to be classified. If the value of the splitting attribute for the new case is lower than *lv*, the left branch will be followed, and if it is greater than *uv* then we will further use the right branch. If the value lies between *lv* and *uv*, **both** branches of the tree are investigated and the results combined probabilistically – the branch with the highest probability will be followed.

5.4. Validating the Model

Notice the asymmetric threshold values for almost every splitting attribute. In this case (FinancialPerf), the accuracy rate of the test sample is comparable with the accuracy rate of the learning sample. There is no specification on how close these two values should be; consequently, we conclude that the tree is validated. The only way to "really" validate the assumption that the two accuracy rates are "not far" from one another is to consider the two accuracy rates as random variables and then use a statistic test to see if their means differ significantly. This new step in validating the decision tree model would require splitting the dataset in different ways to obtain different training and test datasets, and then, under the assumption that the accuracy rates are random variables that follow normal distribution, which is not always the case, we would test if their means are or are not statistically different.

After training the decision tree, we tested it on the same data rows used in Section 4.

5.5. Predicting the Classes

The results are illustrated in Table 5. As can be seen in the table, the results are somewhat different from those obtained using logistic regression.

Table 5. .Prediction using the decision tree

Operating Margin	ROE	ROTA	Equity to Capital	Quick Ratio	Interest Coverage	Receivables Turnover	Company no.	**Predicted Cluster**
5.621597	17.75955	8.979317	27.02372	0.857129	2.314056	6.8226657	3	**B**
11.0069	15.31568	7.67552	31.23215	0.830754	4.189956	6.2295596	4	**B**
16.27344	22.78149	11.16978	34.59247	0.629825	5.205047	6.0291793	5	**A**

M-Real (no.3) was classified as a D company in Table 3, while it is a B company in table 5. The data rows of Stora Enso and M-real are generally similar, but the decision tree has placed more emphasis on ROE, while logistic regression seems to have emphasized Equity to Capital. Also, we can see from Table 6 that the decision tree has not quite correctly learned the pattern associated with Group D, only being able to correctly classify 58% of the cases in this group. The logistical regression model was much more successful, and we therefore consider its prediction the more reliable of the two. More study will be needed to judge why this happened.

6. Comparing the Classification Models' Accuracy

While this is not the only way to compare two classification techniques, comparing them using accuracy rates is the most used. In [10] the author compared five predictive models from areas of both machine learning and statistics. A comparison similar to ours was made in [13]. The authors compared logistic regression and decision tree induction in the diagnosis of Carpal Tunnel syndrome. Their findings claim that there is no significant difference between the two methods in terms of model accuracy rates. Also, they suggest that the classification accuracy of the bivariate models (two independent variables) is slightly higher than that of multivariate ones. It is not our goal to compare bivariate and multivariate models, while this can be a subject for further investigations using the datasets presented in this paper.

As we stated in section 5, we will consider only the second dataset when comparing the two methods, since for the first dataset the results were very poor in terms of the accuracy rate. In the last section, we will try to explain why we obtained such poor results using the EconomicPerf dataset.

Conversely, in the case of the second dataset (FinancialPerf) both logistic regression and decision tree models were validated against the split datasets. The differences between accuracy rates were smaller in this case, and the learning dataset accuracy rates were very good (88% and 84,8%). Also, both models performed similarly on the test datasets (89%, 89,5% and 86,5%, 86,5%). The bigger difference for the training datasets could be caused by the fact that when applying the decision tree algorithm, we split the data in two parts using 75% of the rows for the learning dataset. The remaining 25% was used as a test dataset. This was due to a number-of-rows restriction in the See5 demo-software (max 400 rows of data). Using logistic regression, changes in accuracy rates can occur when including/excluding some variables in/from the model. In the case of the decision tree, the accuracy rate of the model can be tuned using model parameters, e.g. the minimum number of cases in each leaf (m) or the pruning confidence factor (c). The accuracy rates for the two methods are illustrated in Table 6.

Table 6. The observed accuracy rates of the two methods

Logistic Regression

		Observed						
		a	b	c	d	e	f	g
Predicted	a	88%	6%	2%	4%			
	b	5%	89%	3%	2%			
	c	6%	6%	77%	4%	4%	2%	
	d		6%	2%	84%	8%		
	e			7%	1%	88%	4%	
	f					11%	89%	
	g					3%		97%

Decision Tree

		Observed						
		a	b	c	d	e	f	g
Predicted	a	86%	10%		4%			
	b	4%	87%	5%	1%			3%
	c	3%	8%	76%	5%	8%		
	d	0%	18%	6%	58%	12%		6%
	e			2%		93%	2%	4%
	f					3%	94%	3%
	g				2%	4%	4%	90%

7. Discussion and conclusions

In this study, we have proposed a new two-level approach for making class predictions about countries'/companies' economic/financial performance. We have applied our methodology on two datasets: the EconomicPerf dataset that includes variables describing the economic performance of central-east European countries during 1993-2000, and the FinancialPerf dataset, which includes financial ratios describing the financial performance of international pulp and paper companies during 1995-2000. Firstly, SOM clustering was applied on both datasets in order to identify clusters in terms of economic/financial performance, and the optimal number of clusters to consider. By reading the SOM output (U-matrix maps), we have considered seven to be the most appropriate number of clusters for both datasets. Consequently, we construct the outcome values for each data row based on the SOM maps and the corresponding seven classes: best, slightly below best, slightly above average, average, slightly below average, slightly above poor, and poorest. Secondly, based on the new datasets (updated with the outcome values), we have predicted to which class a new input belongs. We chose and compared two predictive models for classification: logistic regression and decision tree induction.

Why is this approach important? Why combine clustering and classification techniques? Why not directly construct the outcome values and apply the predictive models without performing any clustering? We could perform surveys, asking experts how their company/country performed in different months or years, and then directly apply the classification technique to develop prediction models as new cases are to be classified. First of all, this kind of information (outcome values for each data row) is not easy to get (is costly), and secondly, even if we have it, in order for it to be useful, it has to be "true" and "comparable". What we mean by "true" is that when performing surveys, the respondents can be subjective, giving higher rankings for their country/company (not giving true answers). The outcome values can be un-"comparable" if, for example, one person has different criteria for the term “best performance“ than another. In the best perspective, when answering our questions about their country/company performances the respondents would, most probably, classify their country/company using their knowledge and internal aggregate information. We think our methodology is an objective way of making class predictions about countries'/companies' performances since, using it, we can choose the correct number of clusters, define the outcome values for each data row, and construct the predictive model. Also, the problem of inserting new data into an existing model is solved using this method. The problem is that we normally have to train new maps every time, or standardize the new data according to the variance of the old dataset, in order to add new labels to the maps. Inserting new data into an existing SOM model becomes a problem when the data have been standardized, for example, within an interval like [0,1]. Also, the retraining of maps requires considerable time and expertise. We propose that our methodology solves these problems associated with adding new data to an existing SOM cluster model.

The results show that our methodology can be successful, if it is correctly implemented. Clustering is very important in our methodology, since we define the outcome values for each data row based on it. Our U-matrix maps clearly show seven identifiable clusters. More investigations should be performed on finding the utility of each clustering or, in other words, define "how well" we clustered the data. To evaluate the maps we used two criteria: the average quantization error and the ease-of-readability of each map. As a further research problem, we would try to develop a new measure, or use an existing one, to validate the clustering. When applying logistic regression, we obtained models with acceptable accuracy rates. All the coefficients of all regression equations were statistically significant except one (CURRENCY for the

EconomicPerf dataset). The accuracy rates were evaluated using two criteria: proportional by chance criterion and maximum by chance criterion. The first dataset's accuracy rate didn't satisfy the second criterion. When comparing the two classification techniques, we therefore only took into consideration the results of the second. However, like in [13] our findings claim that the results of the two classification techniques are similar in terms of accuracy rate. Also, when making predictions using the two models, we used data for the FinancialPerf dataset from year 2001. Two out of three new data rows were classified in the same class using both predictive models (Stora Enso and UPM-Kymmene to classes 2 and 1 respectively).

An improvement to our methodology would be to tackle the problem of variable selection for both the clustering and the classification phases, finding a new way to measure clustering utility, and generalizing the methodology. As further research, we will investigate different methods of improving our classification models.

Acknowledgements

The authors would like to thank Professor Barbro Back for her constructive comments on the article.

References

[1] B. Back, K. Sere, and H. Vanharanta, "Managing Complexity in Large Data Bases Using Self-Organizing Maps", *Accounting Management and Information Technologies 8 (4)*, Elsevier Science Ltd, Oxford, 1998, pp. 191-210.

[2] A. Costea, A. Kloptchenko, and B. Back, "Analyzing Economical Performance of Central-East-European Countries Using Neural Networks and Cluster Analysis", in *Proceedings of the Fifth International Symposium on Economic Informatics*, I. Ivan. and I. Rosca (eds), Bucharest, Romania, May, 2001, pp. 1006-1011.

[3] T. Eklund, B. Back, H. Vanharanta, and A. Visa, "Assessing the Feasibility of Self-Organizing Maps for Data Mining Financial Information", in *Proceedings of the Xth European Conference on Information Systems (ECIS 2002)*, Gdansk, Poland, June 6-8, 2002, pp. 528-537.

[4] J. F. Hair, Jr, R. Anderson, and R. L. Tatham, Multivariate Data Analysis with readings. Second Edition. Macmillan Publishing Company, New York, New York, 1987.

[5] D. Hand, H. Mannila, and P. Smyth, *Principles of Data Mining*, MIT Press, Cambridge, 2001.

[6] S. Kaski and T. Kohonen, "Exploratory Data Analysis by the Self-Organizing Map: Structures of Welfare and Poverty in the World", in *Neural Networks in Financial Engineering*, N. Apostolos, N. Refenes, Y. Abu-Mostafa, J. Moody, and A. Weigend. (Eds), World Scientific, Singapore, 1996, pp. 498-507.

[7] J. P. Guiver and C. C. Klimasauskas, "Applying Neural Networks, Part IV: Improving Performance", *PC/AI Magazine 5 (4)*, Phoenix, Arizona, 1991, pp. 34-41.

[8] T. Kohonen, *Self-Organizing Maps*, 2nd edition, Springer-Verlag, Heidelberg, 1997.

[9] J. Lehtinen, *Financial Ratios in an International Comparison*, Acta Wasaensia 49, Vasa, 1996.

[10] J. R. Quinlan, "A Case Study in Machine Learning", in *Proceedings of ACSC-16 Sixteenth Australian Computer Science Conference*, Brisbane, Jan. 1993, pp. 731-737.

[11] J. R. Quinlan, *C4.5 Programs for Machine Learning*, Morgan Kaufmann Series in Machine Learning, Morgan Kaufmann Publishers, San Mateo, 1993.

[12] J. Rhiannon, C. Jewitt, L. Galasso, and G. Fortemps, "Consolidation Changes the Shape of the Top 150", *Pulp and Paper International 43 (9)*, Paperloop, San Francisco, California, 2001, pp. 31-41.

[13] S. 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 32*, Academic Press, 1999, 391-414

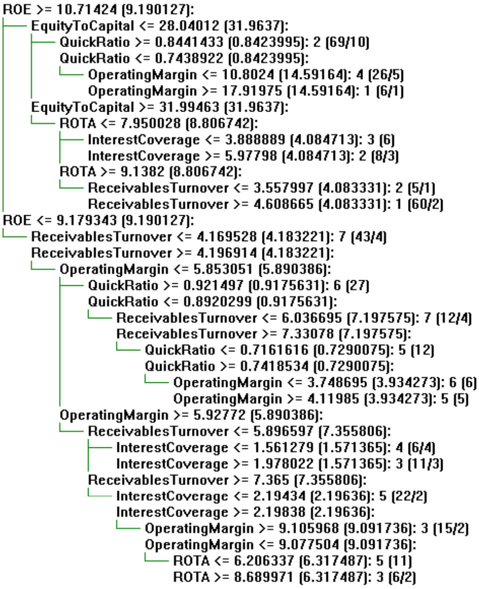
[14] A. Ultsch, "Self organized feature planes for monitoring and knowledge acquisition of a chemical process", in *Proceedings of the International Conference on Artificial Neural Networks*, Springer-Verlag, London, 1993, pp. 84-867.

[15] J. Vesanto "Neural Network Tool for Data Mining: SOM Toolbox", in *Proceedings of Symposium on Tool Environments and Development Methods for Intelligent Systems (TOOLMET2000)*, Oulun yliopistopaino, Oulu, Finland, 2000, pp. 184-196.

[16] J. Vesanto and E. Alhoniemi, "Clustering of the Self-Organizing Map", *IEEE Transactions on Neural Networks 11 (3)*, IEEE Neural Networks Society, Piscataway, New Jersey, 2000, pp. 586-600.

[17] I. Witten and E. Frank, *Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations*, Academic Press, San Diego, 2000.

Appendix: the decision tree



```
ROE >= 10.71424 (9.190127):
├── EquityToCapital <= 28.04012 (31.9637):
│   ├── QuickRatio >= 0.8441433 (0.8423995): 2 (69/10)
│   │   QuickRatio <= 0.7438922 (0.8423995):
│   │   └── OperatingMargin <= 10.8024 (14.59164): 4 (26/5)
│   │       OperatingMargin >= 17.91975 (14.59164): 1 (6/1)
│   EquityToCapital >= 31.99463 (31.9637):
│   └── ROTA <= 7.950028 (8.806742):
│       ├── InterestCoverage <= 3.888889 (4.084713): 3 (6)
│       │   InterestCoverage >= 5.97798 (4.084713): 2 (8/3)
│       ROTA >= 9.1382 (8.806742):
│       └── ReceivablesTurnover <= 3.557997 (4.083331): 2 (5/1)
│           ReceivablesTurnover >= 4.608665 (4.083331): 1 (60/2)
ROE <= 9.179343 (9.190127):
└── ReceivablesTurnover <= 4.169528 (4.183221): 7 (43/4)
    ReceivablesTurnover >= 4.196914 (4.183221):
    └── OperatingMargin <= 5.853051 (5.890386):
        ├── QuickRatio >= 0.921497 (0.9175631): 6 (27)
        │   QuickRatio <= 0.8920299 (0.9175631):
        │   └── ReceivablesTurnover <= 6.036695 (7.197575): 7 (12/4)
        │       ReceivablesTurnover >= 7.33078 (7.197575):
        │       └── QuickRatio <= 0.7161616 (0.7290075): 5 (12)
        │           QuickRatio >= 0.7418534 (0.7290075):
        │           └── OperatingMargin <= 3.748695 (3.934273): 6 (6)
        │               OperatingMargin >= 4.11985 (3.934273): 5 (5)
        OperatingMargin >= 5.92772 (5.890386):
        └── ReceivablesTurnover <= 5.896597 (7.355806):
            ├── InterestCoverage <= 1.561279 (1.571365): 4 (6/4)
            │   InterestCoverage >= 1.978022 (1.571365): 3 (11/3)
            ReceivablesTurnover >= 7.365 (7.355806):
            └── InterestCoverage <= 2.19434 (2.19636): 5 (22/2)
                InterestCoverage >= 2.19838 (2.19636):
                └── OperatingMargin >= 9.105968 (9.091736): 3 (15/2)
                    OperatingMargin <= 9.077504 (9.091736):
                    └── ROTA <= 6.206337 (6.317487): 5 (11)
                        ROTA >= 8.689971 (6.317487): 3 (6/2)
```

Publication 3

Costea A, Eklund T. 2004. Combining Clustering and Classification Techniques for Financial Performance Analysis. In *Proceedings of 8th World Multi-Conference on Systemics, Cybernetics and Informatics (SCI 2004)*, Callaos *et al*. (eds.), Organized by IIIS, Orlando, Florida, USA, July 18-21, 2004, Volume I: Information Systems, Technologies and Applications, Track: Management Information Systems, pp. 389-394. ISBN: 980-6560-13-2.

Combining Clustering and Classification Techniques for Financial Performance Analysis

Adrian COSTEA
Turku Centre for Computer Science, Åbo Akademi University,
Institute for Advanced Management Systems Research,
Lemminkäisenkatu 14B, FIN-20520 Turku, Finland

and

Tomas EKLUND
Turku Centre for Computer Science, Åbo Akademi University,
Institute for Advanced Management Systems Research,
Lemminkäisenkatu 14B, FIN-20520 Turku, Finland

ABSTRACT

The goal of this paper is to analyze the financial performance of world-wide telecommunications companies by building different performance classification models. For characterizing the companies' financial performance, we use different financial measures calculated from the companies' financial statements. The class variable, which for each entrance in our dataset tells us to which class any case belongs, is constructed by applying a clustering technique (the Self-Organizing Map algorithm). We address the issue of map validation using two validation techniques. Then, we address the problem of adding new data, as they become available, into a previously trained SOM map, by building different classification models: multinomial logistic regression, decision tree induction, and a multilayer perceptron neural network. During the experiment, we found that logistic regression and decision tree induction performed similarly in terms of accuracy rates, while the multilayer perceptron did not perform as well. Finally, we propose that, with the correct choice of techniques, our two-level approach provides additional explanatory power over single stage clustering in financial performance analysis.

Keywords: Telecommunications sector, financial performance analysis, SOM, logistic regression, decision trees, perceptron

1. INTRODUCTION

The emergence of the digital era has made a huge amount of financial information freely available. One of the areas greatly affected by the development of the Internet is financial analysis. Managers and stakeholders are increasingly looking to *Knowledge Discovery in Databases* (KDD) for new tools [1]. One KDD tool that could be applied for this type of analysis is the *self-organizing map* (SOM).

The SOM algorithm is a well-known unsupervised-learning algorithm developed by Kohonen in the early 80's [13]. The SOM has been applied in a large number of applications since its conception (see [18]), including a number of financial applications [2], [5], [10], [11], [12], [16], [22].

However, in these articles the problem of class prediction as new cases are added to the datasets have not been addressed. There are two ways to fit the new data, as they become available, into the previously constructed SOM maps. Firstly, new data can be assigned a location on the map using the stored weights between input nodes and output nodes in the SOM model. Thus, appropriately preprocessed new data can be assigned a map location in the existing model. Alternatively, a classification model can be built to model the relationship between the new class variable (obtained in the previous step by the means of one clustering technique, e. g. SOM) and the different financial performance variables.

In this paper, we will explore both methods of class prediction (SOM and classification models). The main reason for using classification models is that they provide more information for explaining how the locations of newly observed data were derived. Quoting Witten & Frank [26] (p.39): "The success of clustering is measured subjectively in terms of how useful the result appears to be to a human user. It may be followed by a second step of classification learning where rules are learned that give an intelligible description of how new instances should be placed into the clusters."

Here we apply our two-level approach which enables us to model the relationship between financial variables and different classifications of companies in terms of their performances. A similar approach, referred to as the Hot Spots Methodology, was suggested in [25], for rule induction from clustering. Williams and Huang tested their approach on insurance and fraud applications.

The rest of the paper is organized as follows. In section 2 we briefly present our two-level approach for modeling the relationship between some financial variables of telecommunications companies and their financial performance classifications. In the following section we present the SOM and the results of the clustering phase, and then in section 4, the three classification models are applied and compared. In section 5 we analyze the class predictions using the data for some companies in 2002, and compare them with the predictions obtained by directly applying the SOM model. In the final section we present our conclusions.

2. METHODOLOGY

Our approach consists of two phases: a clustering phase, in which we obtain several clusters that contain similar data-vectors in terms of Euclidean distances, and a classification phase, in which we construct a class predictive model in order to place new data within the clusters obtained in the first phase.

Among clustering techniques, the SOM (a non-hierarchical clustering technique) has the advantages of providing a visual approximation of inter-similarities in the data and low computational cost. For further information on different clustering techniques and how they work, see [8].

In the classification phase we want to build a model that describes one categorical variable (our performance class) against a vector of dependent variables (in our case: the financial ratios). Here we use three different classifiers: *multinomial logistic regression* (MLR), *decision trees* (DT) and *artificial feed-forward neural networks* (ANN) in the form of multilayer perceptrons (MLP).

The different steps included in our two-level approach are presented below.

Steps for the clustering phase:

- preprocessing of initial data,
- training using the SOM algorithm,
- choosing the best map, and
- identifying the clusters and attaching outcome values to each data row.

Steps for the classification technique (MLR, DT, MLP):

- applying the classifier,
- assessing model accuracy,
- interpreting the results, and
- validating the model.

This approach was applied on Karlsson [10] financial dataset updated with data for 2001. The dataset consists of 630 data rows. The dataset contains 88 companies from five different regions: Asia, Canada, Continental Europe, Northern Europe, and USA, and consists of seven financial ratios per company per year. The ratios used were: *operating margin, return on equity*, and *return on total assets* (profitability ratios); *current ratio* (liquidity ratio); *equity to capital* and *interest coverage* (solvency ratios); and *receivables turnover* (efficiency ratio). The ratios were chosen from Lehtinen's comparison of financial ratios' reliability and validity in international comparisons [14]. The data used to calculate the ratios were collected from companies' annual reports, using the Internet as the primary medium. The time span is 1995-2001. We use data for the year 2002 to test our classification models against the SOM predictions.

3. CLUSTERING USING SOM

Self-Organizing Maps

The SOM algorithm stands for self-organizing map algorithm, and is based on a two-layer neural network using the unsupervised learning method. The self-organizing map technique creates a two-dimensional map from n-dimensional input data. This map resembles a landscape in which it is possible to identify borders that define different clusters [13]. These clusters consist of input variables with similar characteristics, i.e. in this study, of companies with similar financial performance. The methodology used when applying the self-organizing map is as follows [2]. First, we choose the data material. It is often advisable to standardize the input data so that the learning task of the network becomes easier [13]. After this, we choose the *network topology*, *learning rate*, and *neighborhood radius*. Then, the network is constructed. The construction process takes place by showing the input data to the network iteratively using the same input vector many times, the so-called *training length*. The process ends when the *average quantization error* is small enough. The best map is chosen for further analysis. Finally, we identify and interpret the clusters using the *U-matrix map* and *feature planes*. From the feature planes we can read per input variable per neuron the value of the variable associated with each neuron.

The simplest U-matrix method is to calculate the distances between neighboring neurons, and store them in a matrix, i.e. the output map, which can then be interpreted. If there are "walls" between the neurons, the neighboring weights are distant, i.e. the values differ significantly. The distance values can also be displayed in color when the U-matrix is visualized. Hence, dark colors represent great distances while brighter colors indicate similarities amongst the neurons [23].

In [24] the authors propose a two-level clustering approach that involves clustering the data using SOM, and then clustering the SOM using some other clustering method. Their conclusion was that this approach was computationally more effective than applying the clustering methods directly, while the achieved results were similar. We will use the same approach to identify the clusters on our SOM, as this eliminates the need for subjective identification of the clusters. We will use Ward's method to determine the "real" clusters (see next section).

Applying SOM

When constructing the maps (.cod files) we have used a Windows-based program, developed by one of the authors, which is based on SOM_PAK C source files, available at http://www.cis.hut.fi/research/som_pak/. Nenet v1.1a, available for free-demo download at http://koti.mbnet.fi/~phodju/nenet/Nenet/Download.html, was used to visualize the ".cod" files, and Viscovery SOMine (www.eudaptics.com) was used to identify the clusters on the map using Ward's method.

Preprocessing: In order to ease the training of the map and to avoid the algorithm placing too much emphasis on extreme values, we have removed outliers from the data. Once we have detected the outliers of each variable we have two alternatives: to discard the sample which has at least one outlier value or to keep it by removing the peak(s). We chose the second alternative. Finally, we have standardized the input variables to zero mean and unit standard-deviation.

SOM validation: Different values for SOM parameters were tested on the dataset. We have addressed two technical SOM problems: the map dimensionality and the validation of the quantization error. For a comprehensive study of SOM validation, see [3].

In order to perform these validations we added two more functions to our Visual C++ implementation of the SOM algorithm. With the first function we have studied the variations in the quantization error for different dimensions of the map, as was proposed in [3]. For a number of map architectures (4x4, 5x5, 6x6, 7x7, 8x8, and 9x9) we have applied 100 different bootstrap samples, keeping the other SOM parameters constant. For all architectures, the variation coefficients of the quantization errors were calculated. We expected the variation coefficients to increase with the dimensionality of the map. We have obtained very small variation coefficients (around 2%) meaning that we had no empirical evidence to exclude any architecture. Consequently, we have based our choice of map architecture on the visualization capabilities of different maps.

With the second function we have checked whether there is a significant difference between three different quantization error vectors: 100-100, 90-90, and 90-10 vectors. "100-100" vector was obtained in the case where the entire dataset (100%) was used for training and, also, for testing. “90-90” means that 90% of the dataset was used for training and the same for testing, and “90-10” means that 90% was used for training and the remaining for testing. We have used SPSS Matched Pairs Comparison t-test to compare the means of the three vectors finding that there is a difference between the means but the confidence in that result is poor (P-Value for “case1-case2” pair was 0.051).

While being more empirical than theoretical, these validation techniques give us some more confidence in our SOM results.

Clusters: Once the map architecture (9x7) is chosen we have trained our dataset (using the “100-100” case) obtaining an average quantization error of 0.039245. As in [24] we have used a two-step clustering approach: first, after applying SOM, we have obtained 63 clusters (row clusters). Then, we have identified the “real” clusters (7) by grouping the row clusters using Ward’s Method hierarchical clustering (see Figure 1).

In order to define the different clusters, the feature planes (Appendix) were analyzed as well as the row data. By analyzing the feature planes one can easily discover how well the companies have been performing according to each financial ratio. Dark colors of neurons on a feature plane correspond to low values for that particular variable, while light colors correspond to higher values. In our particular case, all of the variables are positively correlated with good company performance. The class variable that we add to the dataset for each data row is in this case measured on an ordinal scale, rather than on an interval one.

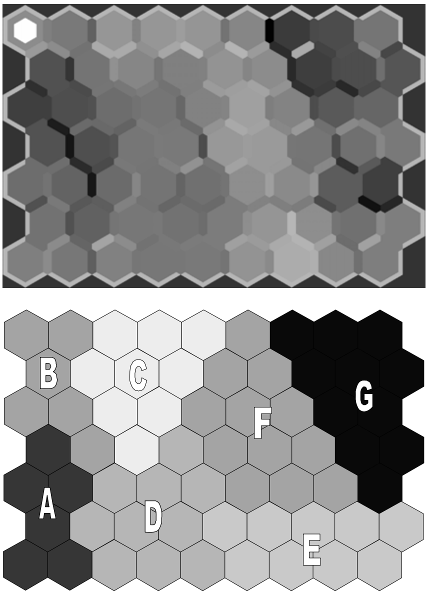


Figure 1. (a) Final 9x7 U-matrix map (1995-01 data) and (b) identified clusters on the map

This means that the classes (A, B, C, D, E, F, and G) are in descending order in terms of companies’ financial performance, but they are not equally distributed (the distances among consecutive classes are different). A short explanation of the characteristics of each group/cluster is presented below:

Group A corresponds to companies performing very well. They exhibit very high profitability, high liquidity and very high solvency. Efficiency varies from high to low.

Group B is very similar to Group A in terms of profitability, but solvency is average and liquidity is poor. Efficiency varies from average to very high. Overall, Group B is also a very good group.

Group C exhibits high profitability, with the exception of return on equity values, which are lower than in groups A and B. Liquidity and efficiency are poor, and solvency is low to average.

Group D consists of fairly average companies. Profitability and liquidity are average, while efficiency is average to poor. Solvency is average in terms of equity to capital values. Interest coverage values, however, are very high in this group.

Group E is very similar to Group D, except for displaying poor liquidity and somewhat higher efficiency.

Group F is a poor group, displaying low values for nearly all ratios, except for operating margin, which is average. This group has particularly poor efficiency.

Group G is the poorest group. This group contains the companies with the poorest values in both profitability and solvency ratios. However, there is a small sub-cluster in the upper right corner showing high liquidity and average solvency. Profitability is very poor for this sub-cluster as well.

After we chose the final map, and identified the clusters, we attached class values to each data row as follows: 1 – A, 2 – B, 3 – C, 4 – D, 5 – E, 6 – F, 7 – G. In the next section we will construct three different classification models to classify a number of companies for the year 2002 into already existing clusters. We will compare the three models in terms of their accuracy rates and class prediction performances.

4. APPLYING THE THREE CLASSIFICATION TECHNIQUES

A more detailed presentation of MLR and DT classification techniques can be found in [20]. ANNs have been extensively used as classifiers. For a review of multi-layer perceptron applications in pattern classification see [4]. Here we will apply our two-level approach presented in section 2 and try to validate it.

Multinomial Logistic Regression

Firstly, multinomial logistic regression was applied on the dataset (updated with values for the class variable, i.e. the associated cluster from the SOM). To see how well the model fits the data we look at the “Model Fitting information” and “Pseudo R-Square” output tables of SPSS (chi-square value has a significance of < 0.0001 and Nagelkerke R^2 = 0.964). This means that there is a strong relationship between the class variables and the financial ratios (96.4% of the class variation is explained by variations in input variables). To evaluate the accuracy rate of 83.3%, we used two criteria (to calculate two new accuracy rates): the proportional by chance and maximum by chance accuracy rates [7] (p. 89-90). These criteria state that the accuracy rate should exceed the expected by chance rates by at least 25% in order to be useful.

Table 1. Evaluating the model's accuracy

	Model	Proportional by chance criterion	Maximum by chance criterion
Telecom	83.3%	16.02%	20.0%

The model accuracy rate is validated against both criteria (it exceeds both standards: 1.25 * 16.02% = 20.02% and 1.25 * 20.0% = 25.0%). To interpret the results of our analysis, we study the "Likelihood Ratio Test" and "Parameter Estimates" output tables of SPSS. All variables are significant ($p < 0.0001$). Not all coefficients for all regression equations are statistically significant. By looking at columns "B" and "exp(B)" from the "Parameter Estimates" output table, we can determine the direction of the relationship and the contribution to performance classification of each independent variable. The findings are as expected e.g.: the likelihood that one data row will be classified into group A is positively correlated with profitability, solvency and efficiency ratios. This corresponds with the characteristics of group A.

We validate our MLR model by splitting the data into two parts of equal size (315 data-rows). When we have used first part ("split" = 0) as the training sample, we have used the second one as test sample, and vice versa. The results are presented in Table 2.

Table 2 shows that we have high accuracy rates, and that the difference between training and validation dataset accuracy rates is relatively small (83.3% - 84.1%). The accuracy rate of the "Part1" and "Part2" datasets (86.0% and 87.0%) validate the main dataset's accuracy rate (83.3%). Also, the differences between learning and validation accuracy rates of the splitting datasets are very small: 86.0% vs. 82.5% for split = 1 and 87.0% vs. 82.6% for split = 0.

Table 2. Datasets' accuracy rates and accuracy rate estimators – MLR

		Main dataset	Part1 (split=1)	Part2 (split=0)
Telecom	Model Chi-Square ($p < 0.0001$)	1802.925	940.485	964.765
	Nagelkerke R2	0.964	0.971	0.975
	Learning Sample	83.3%	86.0%	87.0%
	Test Sample	-	-	-
	Cross Validation	84.1%	82.5%	82.6%
	Significant coefficients ($p<0.0001$)	ALL	ALL	ALL

Decision Tree

Secondly, Quinlan's famous C4.5/C5.0 decision tree algorithm [19] was applied on our dataset. We have used the See5.0 software, which implements a higher-level version of the algorithm. We have performed three runs of the See5 software, exactly like we did when applying logistic regression: one for the whole dataset, another using the first split dataset ("split=1"), and the other using the second half of the data ("split=0").

For comparability reasons, we kept the two most important parameters constant: $m = 5$, which measures the minimum number of cases each leaf-node should have, and $c = 25\%$ (default value), which is a confidence factor used in pruning the tree. The results are presented in Table 3.

Table 3. Dataset accuracy rates and accuracy rate estimators – DT

		Main dataset	Part1 (split=1)	Part2 (split=0)
Telecom	Learning Sample	90.3%	86.0%	86.7%
	Test Sample	-	78.1%	74.6%
	cross-validation	81.3%	75.8%	76.2%

Artificial Feed-Forward Neural Network

Finally, we use a classical artificial feedforward neural network classifier, which is based on the supervised learning algorithm, to perform the second stage of our methodology. The Multilayer Perceptron neural network, trained using the backpropagation algorithm, is currently the most widely used neural network [6]. Multi-layer perceptrons have been used extensively in pattern classification [15], [21], and [9] cited from [4]. Regarding the number of output neurons, we have two alternatives when applying MLPs for pattern classification. The first alternative, which is most commonly used, is to have as many output neurons as the number of classes. The second alternative is to have just one neuron in the output layer, which will take the different classes as values. In this paper we have used one output neuron instead of seven due to the number of cases per weights ratio-restriction. Choosing the number of hidden layers and the number of neurons in each hidden layer is not clear-cut. The choices of these numbers depend on output-input function complexity [17]. It is a well known fact that neural networks are very sensitive regarding the dimensionality of the dataset. In general, a good model is obtained when we have 10 times more training samples than the number of weights.

We have used the sigmoid and linear activation functions for the hidden and output layers respectively. In our experiment (performed using Matlab's Neural Networks toolbox) we have kept all parameters of the MLP constant (the learning algorithm - Scale Conjugate Gradient, the performance goal of the classifier, the maximum number of epochs), except one: the number of neurons in the hidden layer (NH). We repeated the experiment for NH = 7 to 25. The best topology, in terms of test error rate, was for NH = 19. The final topology of the network consists of 7 input neurons, 19 hidden neurons (1 hidden layer) and 1 output neuron (for the class attribute).

The results of applying the MLP classifier are summarized in the following table:

Table 4. Dataset accuracy rates and accuracy rate estimators – MLP

		Main dataset	Part1 (split=1)	Part2 (split=0)
Telecom	Learning Sample	84.13%	85.40%	74.29%
	Test Sample	-	53.97%	63.18%
	cross-validation	84.29%	53.55%	63.55%

As for MLR and DT, we have used 10-folds cross-validation technique.

By looking at Tables 2, 3 and 4 we can compare the three classification models: the accuracy rates for the main dataset were close to each other (83.3%; 90.3% and 84.13%), with the decision tree achieving the best. The classification models were validated against split datasets for the three models: (86.0% and 87.0% for MLR), (86.0% and 86.7% for DT) and (85.4% and 74.29% for MLP). Using our results we can verify the findings of [20] that MLR and DT perform similarly in terms of the accuracy rates of the models. Even though it provides the smallest accuracy rates, MLR seems to be the most robust model out of the three. The logic behind this is that we have the smallest differences between learning and validation accuracy rates when applying MLR technique.

In the next section we will test our classification models against SOM class predictions using data for 2002.

5. PREDICTION ANALYSIS OF ASIAN COMPANIES

In this section we predict the performance classes (2002) for Asian telecommunications companies. Firstly, we standardize the new data using the same preprocessing method and previously achieved normalization. We then label these data into the existing SOM model, in order to obtain the SOM classification. This is possible to do with a small amount of additional data (in our case, sixteen rows of data), but inserting large amounts of data using this method would be problematic. Secondly, we classify the new data using our classification models. Finally, we compare the class predictions obtained with both classification and SOM models.

In Table 5, the class predictions based on financial data for the year 2002 are illustrated. The first column labels each data row. Columns 2, 3, and 4 show the predicted performance class using logistic regression, decision tree induction, and the multilayer perceptron respectively. The final column shows how the original SOM model would have classified the new data.

Table 5. Class predictions

Company	Predicted Cluster			SOM
	MLR	DT	MLP	
73_02	F	F	**E**	F
76_02	G	G	G	G
77_02	**F**	**F**	**E**	G
79_02	D	D	**B**	D
81_02	G	G	**F**	G
82_02	F	**G**	**D**	F
83_02	F	F	**G**	F
84_02	G	G	**F**	G
86_02	F	F	F	F
87_02	F	**E**	**D**	F
88_02	F	F	**E**	F
89_02	C	C	**A**	C
90_02	G	G	**F**	G
91_02	**F**	E	**C**	E
92_02	**D**	**B**	**B**	C
93_02	F	F	**E**	F

We can state that MLR and DT performed very similarly. 12 of 16 companies were classified in the same clusters using these methods. The results of the MLR model slightly better match the SOM classification, with only three cases being incorrectly classified, and these were always in a neighboring class. DT misclassified four cases, and like MLR, these were all placed in neighboring classes. Therefore, we can again conclude that these methods perform quite similarly.

As can be seen, for this data set MLP is more optimistic than the other methods – nearly all companies were placed in higher classes. In our further research we plan to thoroughly investigate the issue of choosing the number of output neurons in order to find the best MLP architecture for our particular problem.

6. CONCLUSIONS

In this study, we were interested in analyzing the financial performance of world-wide telecommunications companies by building different performance classification models. We have trained a SOM model for financial performance analysis, and used three predictive methods (multinomial logistic regression, decision tree induction, and multilayer perceptron) to create classifiers based on our SOM model.

We have addressed the issue of SOM validation. We have applied a bootstrap methodology to validate our quantization error by computing the variation coefficient of the quantization error. We have obtained very small variation coefficients meaning that we had no empirical evidence to exclude any architecture. We also test the variation of the quantization error using different parts of the dataset for training and testing, and while a statistical Matched Pairs Comparison T-means test indicates a difference, the confidence is poor. Therefore, we find no evidence to refute the validity of the trained map.

We have used new financial data (for 2002) to classify a number of Asian companies based on the SOM model. The results show that our multinomial logistic regression and decision tree models were able to capture the patterns in the SOM model better than the artificial neural network. Moreover, as was shown in [20], MLR and DT perform quite similarly. Although the accuracy rates of the three models were fairly similar, the MLP performed differently (more optimistically) on the new data, possibly due to the network architecture choice.

The results of two of our class prediction models also correspond very well with those produced when using, directly, SOM model as classifier. Finally, we propose that, with the correct choice of techniques, our two-level approach provides additional explanatory power over single stage clustering in financial performance analysis.

ACKNOWLEDGEMENTS

The authors would like to thank Professor Barbro Back for her constructive comments on the article.

REFERENCES

[1] P. Adriaans; D. Zantinge, **Data Mining**, Addison-Wesley Longman, Harlow, 1996.

[2] B. Back; K. Sere; H. Vanharanta, "Managing complexity in large data bases using self-organizing maps", **Accounting Management and Information Technologies**, Vol. 8, No. 4, 1998, pp. 191-210.

[3] E. De Bodt; M. Cottrell; M. Verleysen, "Statistical tools to assess the reliability of self-organizing maps", **Neural Networks** 15, Elsevier (ed.), 2002, pp. 967-978.

[4] M. Egmont-Petersen; J.L. Talmona; A. Hasmana; A.W. Ambergenb, "Assessing the importance of features for multi-layer perceptrons", **Neural Networks** 11, Elsevier, 1998, pp. 623–635.

[5] T. Eklund; B. Back; H. Vanharanta; A. Visa "Assessing the Feasibility of Self-Organizing Maps for Data Mining Financial Information", paper presented at **The Xth European Conference on Information Systems (ECIS 2002)**, Gdansk, Poland, 2002.

[6] M.T. Hagan; H.B. Demuth; M. Beale, **Neural Network Design**, PWS Publishing Company, Boston, USA, 1996.

[7] J.F. Hair, Jr.; R.E. Anderson; R.L. Tatham, **Multivariate Data Analysis with Readings**, Macmillan Publishing Company, New York, 1987.

[8] D. Hand; H. Mannila; P. Smythe, **Principles of Data Mining**, The MIT Press, Cambridge, 2001.

[9] L.K. Hansen; C. Liisberg; P. Salamon, "Ensemble methods for handwritten digit recognition", in S. Y. Kung, F. Fallside, J. A. Sorenson, & C. A. Kaufmann (Eds.), **Proceedings of the 1992 IEEE workshop on neural networks for signal processing**, NJ, USA, 1992, pp. 333–342.

[10] J. Karlsson; B. Back; H. Vanharanta; A. Visa "Financial Benchmarking of Telecommunication Companies", **TUCS Technical Report**, No. 395. Turku Centre for Computer Science, Turku, 2001.

[11] S. Kaski; T. Kohonen, "Exploratory Data Analysis by the Self-Organizing Map: Structures of Welfare and Poverty in the World", Apostolos, N.; Refenes, N.; Abu-Mostafa, Y.; Moody, J.; Weigend, A. (eds), **Neural Networks in Financial Engineering**, World Scientific, Singapore, 1996, pp. 498-507.

[12] K. Kiviluoto, "Predicting bankruptcies with the self-organizing map", **Neurocomputing**, Vol. 21, No. 1-3, 1998, pp. 191-201.

[13] T. Kohonen, **Self-Organizing Maps**, 2nd edition. Springer-Verlag, Heidelberg, 1997.

[14] J. Lehtinen, **Financial Ratios in an International Comparison**, Acta Wasaensia. 49, Vasa, 1996.

[15] R.P. Lippmann, "Pattern clasification using neural networks", **IEEE Communications Magazine**, 1989, pp. 47-64.

[16] B. Martín-del-Brío; C. Serrano-Cinca, "Self-organizing Neural Networks for the Analysis and Representation of Data: Some Financial Cases", **Neural Computing and Applications**, Vol. 1, No. 2, 1993, pp. 193-206.

[17] I. Nastac; R. Matei, "An efficient procedure for artificial neural network retraining - Using the a priori knowledge in learning algorithms" **Proceedings of 5th International Conference on Enterprise Information Systems**, Published by Kluwer Academic Publishers, Angers, France, 2003.

[18] M. Oja; S. Kaski; T. Kohonen, "Bibliography of Self-Organizing Map (SOM) Papers: 1998-2001 Addendum", **Neural Computing Surveys**, Vol. 3, 2003, 1-156. http://www.soe.ucsc.edu/NCS/ (read 25.2.2003)

[19] J.R. Quinlan, **C4.5 Programs for Machine Learning**. Morgan Kaufmann Series in Machine Learning, Morgan Kaufmann Publishers, San Mateo, 1993.

[20] S.M. Rudolfer; G. Paliouras; I.S. Peers, "A Comparison of Logistic Regression to Decision Tree Induction in the Diagnosis of Carpal Tunnel Syndrome", **Computers and Biomedical Research**, Vol. 32, No.5, 1999, pp. 391-414.

[21] C.N. Schizas; C.S. Pattchis; I.S. Schofield; P.R. Fawcett, "Artificial neural nets in computer-aided macro motor unit potential classification", **Transactions of IEEE Engineering in Medicine and Biology** 9 (5), 1990, pp. 31–38.

[22] C. Serrano-Cinca, "Self Organizing Neural Networks for Financial Diagnosis", **Decision Support Systems**, Vol.17, No. 3, 1996, pp. 227-238.

[23] A. Ultsch, "Self organized feature planes for monitoring and knowledge acquisition of a chemical process", **The International Conference on Artificial Neural Networks**, Springer-Verlag: London, 1993.

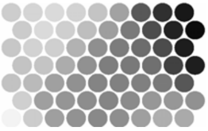
[24] J. Vesanto; E. Alhoniemi, "Clustering of the Self-Organizing Map", **IEEE Transactions on Neural Networks**, Vol. 11, No. 3, 2000, pp. 586-600.

[25] G.J. Williams; Z. Huang, "Mining the Knowledge Mine - The Hot Spots Methodology for Mining Large Real World Databases", Sattar, A. (Ed) **Advanced Topics in Artificial Intelligence**, Lecture Notes in Computer Science, Vol. 1342, Springer-Verlag, 1997, pp. 340-348.

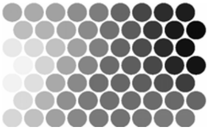
[26] I. Witten; E. Frank, **Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations**, Morgan Kaufmann Publishers, 2000.

APPENDIX

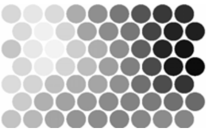
Operating margin



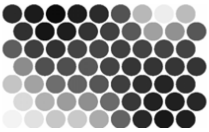
Return on equity



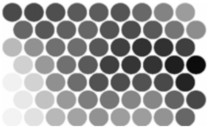
Return on total assets



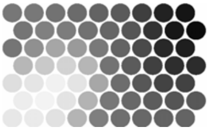
Current ratio



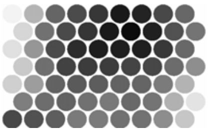
Equity to capital



Interest coverage



Receivables turnover



Publication 4

Alcaraz-Garcia AF, Costea A. 2004. A Weighting FCM Algorithm for Clusterization of Companies as to their Financial Performances. In *Proceedings of the IEEE 4th International Conference on Intelligent Systems Design and Applications (ISDA 2004)*, Rudas I. (ed.), CD-ROM Edition, Budapest, Hungary, August 26-28, 2004, Track: Intelligent Business, pp. 589-594. ISBN: 963-7154-30-2.

A Weighting FCM Algorithm for Clusterization of Companies as to their Financial Performances

Francisco Augusto Alcaraz Garcia
Turku Centre for Computer Science – TUCS
Åbo Akademi University
Lemminkäinengatan 14 B, FIN-20520 Åbo, Finland
E-mail: falcaraz@abo.fi

Adrian Costea
Turku Centre for Computer Science – TUCS
Åbo Akademi University
Lemminkäinengatan 14 B, FIN-20520 Åbo, Finland
E-mail: acostea@abo.fi

***Abstract*— We apply fuzzy logic to group telecommunication companies into different clusters as to their financial performances. The objective is to build an easy-to-use financial assessment tool that can assist decision makers in their investment planning and be applied regardless of the economic sector to be analyzed. We characterize each cluster in terms of profitability, liquidity, solvency and efficiency. We implement a modified fuzzy C-means (FCM) algorithm and compare the results with those of normal FCM and previously reported SOM clustering. The results show an improvement in pattern allocation with respect to normal FCM and SOM. The interpretation of the clusters is done automatically representing each ratio as a linguistic variable.**

I. Introduction

The main target for decision makers is to gain more accurate information about their business market as well as their potential investment opportunities. This is done, in most cases, by analyzing the available historical data (qualitative and quantitative) on the market. The process of obtaining more information about a company's competitors and were it is situated against them is called benchmarking [1] [5]. From the investor's point of view, it is important to see the weaknesses and strengths of a business before the decision to invest or not is taken. Managers and potential investors need financial performances in terms of profitability, liquidity, solvency and efficiency of all economic actors (companies) on that business stage. Analysts have to summarize the high dimensional data to make them interpretable. In this process clustering techniques play a central role.

Clustering is "the organization of a collection of patterns – usually represented as a vector of measurements, or a point in a multidimensional space – into clusters based on similarity" [6]. Traditional clustering methods intend to identify patterns in data and create partitions with different structures. These partitions are called clusters and elements within each cluster should share similar characteristics. In principle, every element belongs to only one partition, but there are observations in the data set that are difficult to position. In many cases subjective decisions have to be made in order to allocate these uncertain observations.

In contrast to these methods, fuzzy logic deals with uncertainty that comes from imprecise information and vagueness. The conventional Boolean logic is substituted by degrees or grades of truth, which allows for intermediate values between true and false. It is common to express the grades of truth by numbers in the closed interval $[0, 1]$, and they can be modeled by membership functions. A membership function assigns a degree of truth (membership degree) for every element subject to the use of that function. The membership function defines a set, called fuzzy set, and degrees of 0 and 1 represent non-membership and full membership respectively to that set, while values in between represent intermediate degrees of set membership.

In this framework, fuzzy clustering methods assign different membership degrees to the elements in the data set indicating in which degree the observation belongs to every cluster. The fuzzy logic approach may also deal with multidimensional data and model nonlinear relationships among variables. It has been applied to financial analysis, for example to evaluate early warning indicators of financial crises [11], or to develop fuzzy rules out of a clustering obtained with self organizing map algorithm [4].

One traditional method in fuzzy clustering is the fuzzy C-means clustering method (FCM) [2]. Every observation gets a vector representing its membership degree in every cluster, which indicates that observations may contain, with different strengths, characteristics of more than one cluster. In this situation we usually assign the elements of the data set to the cluster that has the highest membership degree. In spite of the additional information provided by the methodology, there is a problem with the observations that are difficult to position (uncertain observations) when they obtain similar membership values for two or more clusters.

This paper applies a method to allocate the uncertain observations by introducing weights to the FCM algorithm. The weights indicate the level of importance of each attribute in every cluster so that allocation is done depending on the linguistic classification of the partitions. The data set used corresponds to 7 financial ratios of 88 worldwide telecom companies during the period 1995 to 2001 in an annual basis. The results show that the characterization of the clusters by means of linguistic variables gives an easy to understand, yet formal, classification of the partitions. Also, when weights are extracted from these characteristics, the uncertain observations are allocated. The comparison of the results with other methods is discussed.

II. FCM Algorithm

The FCM algorithm uses as clustering criterion the minimization of an objective function, $J_m(U,v)$, and was developed by Bezdek [2] in 1981. The algorithm partitions a multidimensional data set into a specific number of clusters, giving a membership degree for every observation in every cluster. The objective function to minimize is

$$J_m(U,v) = \sum_{k=1}^{n} \sum_{i=1}^{c} (u_{ik})^m (d_{ik})^2 \qquad (1)$$

where c is the number of clusters, n is the number of observations, $U \in M_{fc}$ is a fuzzy c-partition of the data set X, $u_{ik} \in [0,1]$ is the membership degree of observation x_k in cluster i,

$$d_{ik} = \|x_k - v_i\| = \left[\sum_{j=1}^{p} (x_{kj} - v_{ij})^2 \right]^{1/2} \qquad (2)$$

is the Euclidean distance between the cluster center v_i and observation x_k for p attributes (financial ratios in our case), $m \in [1, \infty)$ is the weighting exponent, and the following constraint holds

$$\sum_{i=1}^{c} u_{ik} = 1. \qquad (3)$$

If m and c are fixed parameters then, by the Lagrange multipliers, $J_m(U,v)$ may be globally minimal for (U,v) only if

$$\underset{\substack{1\le i\le c\\ 1\le k\le n}}{\forall} \; u_{ik} = 1 \Bigg/ \left[\sum_{j=1}^{c} \left(\frac{d_{ik}}{d_{jk}} \right)^{2/(m-1)} \right] \qquad (4)$$

and

$$\underset{1\le i\le c}{\forall} \; v_i = \left[\sum_{k=1}^{n} (u_{ik})^m x_k \right] \Bigg/ \left[\sum_{k=1}^{n} (u_{ik})^m \right] \qquad (5)$$

When $m \rightarrow 1$, the Fuzzy c-Means converges to the Hard c-Means (HCM), and when we increase its value the partition becomes fuzzier. When $m \rightarrow \infty$, then $u_{ik} \rightarrow 1/c$ and the centers tend towards the centroid of the data set (the centers tend to be equal). The exponent m controls the extent of membership sharing between the clusters and there is not theoretical basis for an optimal choice for its value.

The algorithm consists of the following steps:

- Step 1. Fix c, $2 \le c \le n$, and m, $1 \le m \le \infty$. Initialize $U^{(0)} \in M_{fc}$. Then, for s^{th} iteration, $s = 0, 1, 2, \ldots$:
- Step 2. Calculate the c cluster centers $\{v_i^{(s)}\}$ with (5) and $U^{(s)}$.
- Step 3. Calculate $U^{(s+1)}$ using (4) and $\{v_i^{(s)}\}$.
- Step 4. Compare $U^{(s+1)}$ to $U^{(s)}$: if $\|U^{(s+1)} - U^{(s)}\| \le \varepsilon$ stop; otherwise return to Step 2.

Since the iteration is based on minimizing the objective function, when the minimum amount of improvement between two iterations is less than ε the process will stop.

One of the main disadvantages of the FCM is its sensitivity to noise and outliers in data, which may lead to incorrect values for the clusters' centers. Several robust methods to deal with noise and outliers have been presented in [10]. Here, for simplicity, the outliers and far outliers have been leveled in order to minimize their effect in the FCM and the weighting approach suggested.

III. Weighting FCM to Allocate Uncertain Observations

The FCM algorithm gives the membership degree of every observation for every cluster. The usual criterion to assign the data to their clusters is to choose the cluster where the observation has the highest membership value. While that may work for a great number of elements, some other data vectors may be misallocated. This is the case when the two highest membership degrees are very close to each other, for example, one observation with a degree of 0.45 for the first cluster and 0.46 for the third. It is difficult to say in which cluster should we include it and it is possible that, after analyzing the vector components, we realize it does not correspond to the average characteristics of the cluster chosen. We call this data vector as "uncertain" observation. Therefore, it would be useful to introduce in the algorithm some kind of information about the characteristics of every cluster so that the uncertain observations can be better allocated depending on which of these features they fulfil more.

A. Generation of Linguistic Variables

When we analyze a group of companies by their financial performances, we have to be aware of the economic characteristics of the sector they belong to. Levels of ratios showing theoretical bad performances may indicate, for the specific sector, a good or average situation for a company. Conversely, a good theoretical value for the same indicator may indicate a bad evolution of the enterprise in another sector. Usually, financial analysts use expressions like: "high rate of return", "low solvency ratio", etc. to represent the financial situation of the sector or the company. Expressions like that can be easily modeled with the use of linguistic variables and allow the comparison of different financial ratios in a more understandable way regardless of the sector of activity.

Linguistic variables are quantitative fuzzy variables whose states are fuzzy numbers that represent linguistic terms, such as *very small*, *medium*, and so on [7]. In our study we model the seven financial ratios with the help of seven linguistic variables using five linguistic terms: *very low* (VL), *low* (L), *average* (A), *high* (H), *very high* (VH). To each of the basic linguistic terms we assign one of five fuzzy numbers, whose membership functions are defined on the range of the ratios in the data set. It is common to represent linguistic variables with linguistic terms positioned symmetrically [12]. Since there is no reason to assume that the empirical distributions of the ratios in our data set are symmetric, we applied the normal FCM algorithm to each ratio individually in order to obtain the fuzzy numbers, which appeared not to be symmetric. Therefore, the linguistic terms are defined specifically for the sector into consideration. The value of m was set to

1.5 because it gave a good graphical representation of the fuzzy numbers, and these were approximated to fuzzy numbers of the trapezoidal form. The graphical representation of the linguistic variable for operating margin is shown in Figure 1 and its trapezoidal approximation in Figure 2.

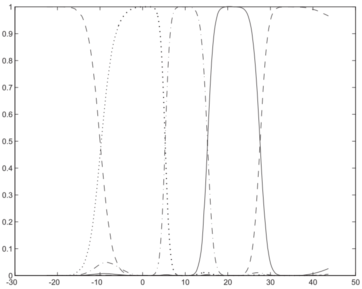


Fig. 1. Linguistic variable representation of operating margin

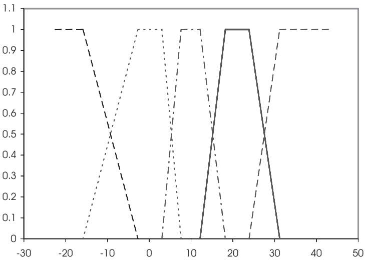


Fig. 2. Trapezoidal approximation for operating margin

Using this approach we can characterize every observation (financial performances of one company in one period), as having *high*, *average*, etc. values in different ratios with respect to the rest of the companies from the same sector. It gives information about the relative situation of the company against its competitors with respect to each individual ratio.

B. Calculation of Weights for the FCM

Once we have the linguistic variables for all financial ratios in our data set, we can obtain an importance coefficient (weight) for every ratio in every cluster and introduce it in the clustering algorithm. The objective is to better allocate uncertain observations taking into consideration the linguistic characterizations of the ratios from the certain observations in every cluster.

In order to separate between certain and uncertain observations the FCM algorithm was applied to the initial data set using $m = 1.5$ and $c = 7$. Other clustering methods like SOM [8] showed the appropriateness of seven clusters for the given data set, therefore seven clusters were chosen to make comparisons possible.

We considered as uncertain observations those for which the difference between the two maximum membership degrees was less than twice the equal membership level for every cluster: $2 * 1/c$, which seems a reasonable assumption in our data set to clearly define linguistic structures in the clusters. By removing the uncertain observations from the clusters we can represent in a better way the true properties of the clusters and, therefore, obtain clearer classification rules.

Once we have the clusters with the certain observations we can apply the linguistic variables obtained in the previous section to determine the linguistic characterizations. In every cluster and for every ratio we can obtain how many times every linguistic term appears and also the percentage with respect to the total number of observations in the cluster. Clearly, a ratio will be important for the cluster if it has a high percentage of occurrences concentrated in few linguistic terms. In the contrary, if one ratio has a number of occurrences evenly distributed among the linguistic terms, it will not be a good definer of the cluster. As a measure of how evenly or unevenly the percentages of the occurrences are distributed we use the standardized variation coefficient (SVC_{ij}). Let us denote with $perc_{ij}$ the vector of percentages of ratio j in cluster i. One element of this vector, $perc_{ij}(k)$, will denote the percentage of occurrences of linguistic term (LT) k for ratio j in cluster i.

$$perc_{ij}(k) = \frac{\text{nr of occurences of LT k for ratio j in cluster i}}{\text{nr of certain observations in cluster i}} \tag{6}$$

where $k = 1(VL), 2(L), 3(A), 4(H), 5(VH)$.

The variation coefficients and the standardized variation coefficients are

$$VC_{ij} = \frac{\text{standard deviation } (perc_{ij})}{\text{mean } (perc_{ij})} \tag{7}$$

and

$$SVC_{ij} = \frac{VC_{ij}}{\sum_{j=1}^{p} VC_{ij}} \tag{8}$$

A high variation coefficient of the percentages indicates that the ratio clearly defines the cluster. After we split the data in certain and uncertain observations, we calculate the weights (SVC_{ij}) using only the certain information. These weights remain constant throughout the iterations of the algorithm. In every iteration, after allocating new uncertain observations, we obtain new clusters' centers and new membership degree values for those observation that remain uncertain.

C. Modified FCM

The previous weights are introduced in the Euclidean distance term of the FCM algorithm in the following form:

$$d_{ik} = \left[\sum_{j=1}^{p} (x_{kj} - v_{ij})^2 SVC_{ij} \right]^{1/2} \tag{9}$$

where SVC_{ij} is the standardized variation coefficient of cluster i for the ratio j, and it fulfils the constraint (10) since they are standardized before introducing them in the objective function.

$$\sum_{j=1}^{p} SVC_{ij} = 1 \tag{10}$$

At each iteration s we should find the membership degrees that minimize the following objective function:

$$J_m(U,v) = \sum_{k\in I}\sum_{i=1}^{c} (u_{ik}^{(s)})^m (d_{ik}^{(s)})^2 \left(1-u_{ik}^{(s-1)}\right) \tag{11}$$

where I is the set of certain observations in iteration s and $u_{ik}^{(s-1)}$ is the membership degrees of the certain observations for cluster i corresponding to the previous iteration. This term is introduced to avoid that lower membership degrees from the uncertain observations become more important in the new allocation. A higher previous membership degree value $u_{ik}^{(s-1)}$ should lead to a lower recalculated distance from that uncertain observation to the center of that cluster. Therefore, $1-u_{ik}^{(s-1)}$ is used when calculating the new distances.

The Lagrange function to minimize the objective function (11)

$$\begin{aligned} J_{m,\lambda}(U,v) \quad = \quad & \sum_{k\in I}\sum_{i=1}^{c} (u_{ik}^{(s)})^m \left(1-u_{ik}^{(s-1)}\right)\cdot \\ & \sum_{j=1}^{p} (x_{kj}-v_{ij}^{(s)})^2 SVC_{ij} - \\ & \sum_{k\in I} \lambda_k \left(\sum_{i=1}^{c} u_{ik}^{(s)} - 1\right) \end{aligned} \tag{12}$$

leads to the partial derivatives

$$\frac{\partial J_{m,\lambda}(U,v)}{\partial u_{ik}^{(s)}} = m(u_{ik}^{(s)})^{(m-1)} (d_{ik}^{(s)})^2 \left(1-u_{ik}^{(s-1)}\right) - \lambda_k \overset{!}{=} 0 \tag{13}$$

and

$$\frac{\partial J_{m,\lambda}(U,v)}{\partial \lambda_k} = \sum_{i=1}^{c} u_{ik}^{(s)} - 1 \overset{!}{=} 0 \tag{14}$$

We obtain from (13)

$$u_{ik}^{(s)} = \left[\frac{\lambda_k}{m(d_{ik}^{(s)})^2 \left(1-u_{ik}^{(s-1)}\right)}\right]^{(1/(m-1))} \tag{15}$$

and with (14) leads to

$$\left(\frac{\lambda_k}{m}\right)^{(1/(m-1))} = 1 \Bigg/ \sum_{i=1}^{c} \left(\frac{1}{(d_{ik}^{(s)})^2 \left(1-u_{ik}^{(s-1)}\right)}\right)^{(1/(m-1))} \tag{16}$$

that together with (15) gives the expression for the membership degrees

$$u_{ik}^{(s)} = 1 \Bigg/ \sum_{r=1}^{c} \left(\frac{(d_{ik}^{(s)})^2 \left(1-u_{ik}^{(s-1)}\right)}{(d_{rk}^{(s)})^2 \left(1-u_{rk}^{(s-1)}\right)}\right)^{(1/(m-1))} \tag{17}$$

The necessary condition for the cluster centers is

$$\begin{aligned} & \frac{\partial J_{m,\lambda}(U,v)}{\partial v_{ij}^{(s)}} = \\ - \quad & 2\sum_{k=1}^{n} (u_{ik}^{(s)})^m \left(1-u_{ik}^{(s-1)}\right) (x_{kj}-v_{ij}^{(s)}) SVC_{ij} \overset{!}{=} 0 \end{aligned} \tag{18}$$

giving

$$\sum_{k\in I} (u_{ik}^{(s)})^m \left(1-u_{ik}^{(s-1)}\right) x_{kj} = v_{ij}^{(s)} \sum_{k\in I} (u_{ik}^{(s)})^m \left(1-u_{ik}^{(s-1)}\right) \tag{19}$$

and the expression for the cluster centers is

$$v_{ij}^{(s)} = \frac{\sum_{k\in I} (u_{ik}^{(s)})^m \left(1-u_{ik}^{(s-1)}\right) x_{kj}}{\sum_{k\in I} (u_{ik}^{(s)})^m \left(1-u_{ik}^{(s-1)}\right)} \tag{20}$$

We use equations (20) and (17) to update the centers and membership degrees in our algorithm. We propose the following algorithm:

- Step 1. Fix c and m. Initialize $U = U^{(1)}$. Apply normal FCM (see Section II) to all dataset and determine the certain (I) and uncertain (I') sets of observations. Determine SVC_{ij} based on the certain observations. We will denote the final U obtained at this step with $U^{(l)}$. Next (steps 2-5 iteratively), allocate the uncertain observations into the certain clusters. Every iteration $s \in \mathbb{N}$ allocating the uncertain elements consists of following steps:
- Step 2. In the iteration s, calculate the centers of the clusters using equation (20) with the membership degrees $u_{ik}^{(s)}$ and $u_{ik}^{(s-1)}$ corresponding to the certain observations of the current and previous iterations, respectively. When $s = 1$, $u_{ik}^{(1)} = U^{(l)}$ and $u_{ik}^{(0)} = 0$, $\forall i = \overline{1,c}$, $\forall k = \overline{1,n}$.
- Step 3. Calculate $u_{ik}^{(s+1)}$ of the uncertain observations using equation (17) with the centers obtained in Step 2, and the previous degrees $u_{ik}^{(s)}$, $k \in I'$ where I' is the set of uncertain data.
- Step 4. Identify the new certain observations from I' (based on $u_{ik}^{(s+1)}$ from the previous step) and attempt to allocate them in the corresponding clusters. Update I with the new certain observations from I'. The remaining uncertain observations will become I' in the next iteration.
- Step 5. If at least one uncertain observation was allocated go to Step 2. If not, exit.

IV. Implementation

We have applied the normal FCM and the modified version presented in Subsection III-C to our dataset trying to find clusters of financial performance. The dataset consists of 630 observations of 88 companies from five different regions (Asia, Canada, Continental Europe, Northern Europe, and USA) during the period 1995 to 2001. Every observation contains seven financial ratios of a company for a year calculated from companies' annual reports, using the Internet as the primary source. The ratios used were: *operating margin*, *return on equity*, and *return on total assets* (profitability ratios); *current ratio* (liquidity ratio); *equity to capital*, and *interest coverage* (solvency ratios); and *receivables turnover* (efficiency ratio). The ratios were chosen from Lehtinen's [9] comparison of financial ratios' reliability and validity in international comparisons. We have used $m = 1.5$ in the implementation of the algorithm as we have done in the calculation of the linguistic variables, and $c = 7$ to make the results comparable with SOM algorithm from our previous work [3]. We have characterized each cluster by using the linguistic variables of the certain observations obtained in Step 1 of the algorithm (Table I).

TABLE I
CHARACTERIZATION OF CLUSTERS

	OM	ROTA	ROE	Current	E to C	IC	Rec. T.	Order
Cluster 1	VL	VL	VL&L	-	A&H	VL&L	-	Bad
Cluster 2	A	A	A&H	-	A	A	-	Average
Cluster 3	VL&L	VL	VL	-	VL&L	L	-	Worst
Cluster 4	H	H	VH	VL	A	A	A	Good
Cluster 5	A	A	A&H	H	H	VH	-	Good
Cluster 6	L	L	A	L	L	L	-	Bad
Cluster 7	VH	VH	H	VH	VH	VH	-	Best

We considered that one linguistic term characterizes one cluster if it represents more than 40 % out of total number of observations for that cluster. For example, for Cluster 1, and ratio ROE, we have two linguistic terms that have more than 40 % of the occurrences (VL&L). When all linguistic terms for one cluster and one ratio are under 40 % we consider that the ratio is not a good definer for that specific cluster. It seems that Receivables Turnover does not have any discriminatory power among data except for one cluster. By simply comparing the clusters we can easily label them as being good, bad, worst, etc. depending on their linguistic terms, as it is shown in Table I.

After Step 1 of the algorithm we obtained 110 uncertain observations, while the remaining 520 certain observations were distributed among different clusters. Our algorithm allocated all uncertain observations except two. We will treat these two observations separately. A total of 19 observations were clustered differently by our algorithm compared to normal FCM. We characterized each one of these observations using our linguistic variables (see Table II). Column X of Table II shows how many ratios of each observation are characterized by the same linguistic term as the characterization of the cluster (shown in Table I) given by the normal FCM, while column Y has the same meaning but for the cluster given by the modified FCM. If we consider that a method clusters better if it gives a higher number of coincidences in the linguistic terms, 9 out of 19 observations (77, 115, 158, 257, 273, 274, 301, 436, 539) were better clustered by our algorithm compared with 6 (42, 213, 233, 265, 391, 619) clustered better by normal FCM. 4 observations (221, 443, 490, and 614) have an equal number of linguistic term coincidences with the clusters. From this point of view, our implementation overcame, overall, normal FCM.

Last column in Table II shows how SOM clustered these uncertain observations in our previous work. We can also see that our method (modified FCM) has an overall better clustering performance than SOM as well. SOM clustered the nine observations for which modified FCM is better than normal FCM in: a) the same clusters as normal FCM for observations (77, 115, 273, 274, 301), b) the same cluster as modified FCM for ($158, 436, 539$), and c) a different cluster for observation 257. This means that in 8 out of 9 cases modified FCM outperformed SOM or they performed similarly. For those cases when normal FCM outperformed modified FCM, only in one case (265) SOM outperforms modified FCM by clustering this observation in the same cluster as normal FCM.

Observations 321 and 442 were not allocated by our algorithm because their two highest membership degree values are too close to each other. Observation 321 has a membership degree of $34,87$ % for Cluster 4 and $31,76$ % for Cluster 5, while observation 442 has $34,9$ % for Cluster 2 and $26,77$ % for Cluster 1.

Observation 321 corresponds to IDT Company for the year 2001, which experienced rather strange financial results: VL operating margin, VH return on total assets, and VH return on equity (all being profitability ratios). It is difficult, therefore, to make an assessment regarding its profitability performance. Subjectively, Clusters 4 and 5 being labeled as good clusters, we can consider IDT financial performance in 2001 as being "good".

Observation 442 corresponds to the average of US companies in the year 1999. Normal FCM clustered this observation in a good cluster, while our approach was more pessimistic by placing the observation in an average cluster (Cluster 2). This observation shows a pattern similar to IDT in 2001 (observation 321) in the sense that has opposite values for different profitability ratios. Moreover, being an average of US telecommunication companies, we would place it (as our modified FCM shows) in an average cluster rather than in a good one as normal FCM recommends.

V. CONCLUSIONS

We have implemented a modified version of the traditional fuzzy C-mean algorithm by introducing some weights measures which better characterize each cluster and each ratio.

Firstly, we have built the clusters using certain information (observations with high differences between the highest two membership degree values). The weights were calculated using seven linguistic variables (one for each ratio) using five linguistic terms: *very low* (VL), *low* (L), *average* (A), *high* (H), *very high* (VH). The remaining uncertain observations were reallocated in the certain clusters by using these weights to calculate new distances between the uncertain observations and the new centers of the certain clusters.

TABLE II

UNCERTAIN OBSERVATIONS

Obs	OM	ROTA	ROE	Current	E to C	IC	Rec. T.	Normal FCM	X	Modif FCM	Y	SOM
42	A	H	H	A	H	VH	L	5 - G	5	2 - A	4	A
77	A	L	VL	A	A	A	L	6 - B	2	3 - W	3	B
115	H	A	A	L	L	L	VH	2 - A	4	6 - B	5	A
158	L	L	L	VL	VH	L	H	5 - G	1	1 - B	4	B
213	A	L	A	VL	VH	L	A	5 - G	3	7 - Be	2	A
221	L	L	L	L	A	L	L	6 - B	5	1 - B	5	B
233	A	A	A	H	VH	VH	L	5 - G	6	7 - Be	3	A
257	L	L	L	VH	A	L	A	6 - B	4	1 - B	5	W
265	H	H	H	VL	H	A	H	2 - A	4	5 - G	3	A
273	H	H	H	L	A	A	L	4 - G	4	2 - A	5	G
274	H	H	H	L	A	A	L	4 - G	4	2 - A	5	G
301	H	H	VH	VH	H	H	L	2 - A	2	4 - G	3	A
321[1]	VL	VH	VH	VH	H	VH	VL	5 - G	3	4 or 5 - G	-	A
391	A	A	A	VH	VH	H	A	5 - G	4	7 - Be	3	A
436	A	VH	H	H	H	L	VH	7 - Be	3	5 - G	5	G
442[1]	A	H	VL	H	A	VH	H	5 - G	4	2 or 1 - A	-	A
443	L	A	L	H	A	VH	A	5 - G	4	2 - A	4	A
490	VL	VL	VL	VL	L	VL	A	3 - W	6	1 - B	6	W
539	L	L	A	A	A	A	A	6 - B	4	2 - A	5	A
614	L	L	L	VH	L	VL	L	3 - W	4	6 - B	4	W
619	L	L	A	A	A	L	A	6 - B	5	2 - A	4	A

[1] Unallocated uncertain observations W – worst, B – bad, A – average, G – good, Be – best

We have compared the results of this approach with normal FCM and SOM using a dataset of 88 worldwide telecommunication companies. Our version outperformed both normal FCM and SOM clustering techniques finding better clusters for the uncertain observations. Also, compared with the other two methods, the use of linguistic variables gave our method a better explanatory power of each cluster. We can now, automatically, characterize each cluster and, also, find those observations that need to be treated carefully due to their specifics.

REFERENCES

[1] T. Bendell, L. Boulter, P. Goodstadt, *Benchmarking for Competitive Advantage*, Pitman Publishing, London, 1998.

[2] J. C. Bezdek, *Pattern Recognition with Fuzzy Objective Function Algorithms*, Plenum Press, New York, 1981.

[3] A. Costea, T. Eklund, J. Karlsson, “A framework for predictive data mining in the telecommunication sector”, *Proceedings of the IADIS International Conference - WWW/Internet*, Published by IADIS Press, Lisbon, Portugal, November 2002.

[4] M. Drobics, W. Winiwarter, U. Bodenhofer, “Interpretation of Self-Organizing Maps with Fuzzy Rules”, *Proceedings of the* ICTA 2000 *– The Twelfth IEEE International Conference on Tools with Artificial Intelligence*, Vancouver, 2000.

[5] J. Ellis, D. Williams, *Comparative Financial Analysis in Corporate Strategy and Financial Analysis: Managerial, Accounting and Stock Market Perspectives*, Financial Times/Pitman Publishing, London, pp. 203–247.

[6] A. K. Jain, M. N. Murty, P. J. Flynn, “Data Clustering: A Review”, *ACM Computing Survey*, vol. 31, no. 3, September 1999.

[7] G. J. Klir, B. Yuan, *Fuzzy Sets and Fuzzy Logic. Theory and Applications*. Prentice Hall PTR, Upper Saddle River, New Jersey, 1995.

[8] T. Kohonen, *Self-Organizing Maps*, 2nd edition. Springer-Verlag, Heidelberg, 1997.

[9] J. Lehtinen, *Financial Ratios in an International Comparison. Validity and Reliability*, Acta Wasaensia, no. 49, Vasa, 1996.

[10] J. Leski, “Towards a robust fuzzy clustering”, *Fuzzy Sets and Systems* 137, pp. 215–233, 2003.

[11] C. K. Lindholm, S. Liu, “Fuzzy Clustering Analysis of the Early Warning Signs of Financial Crisis”, *Proceedings of the FIP2003, an International Conference on Fuzzy Information Processing: Theory and Applications*, Beijing, March 1–4, 2003.

[12] T. Lindström, “A fuzzy design of the willingness to invest in Sweden”, *Journal of Economic Behavior and Organization*, vol. 36, pp. 1–17, 1998.

Publication 5

Costea A, Nastac I. 200x. Assessing the Predictive Performance of ANN-based Classifiers Based on Different Data Preprocessing Methods, Distributions and Training Mechanisms. Submitted to the *International Journal of Intelligent Systems in Accounting, Finance and Management* (in review process).

An earlier version was published as TUCS Technical Report:
Costea A, Nastac I. 2005. Three factors affecting the predictive performance of ANNs: pre-processing method, data distribution and training mechanism. *TUCS Technical Report* No. 679, Turku, Turku Centre for Computer Science, April 2005.

Assesing the predictive performance of ANN-based classifiers based on different data preprocessing methods, distributions and training mechanisms

Adrian Costea* and Iulian Nastac

Turku Centre for Computer Science,
Institute for Advanced Management Systems Research,
Åbo Akademi University, Finland,
{*Adrian.Costea, Iulian.Nastac*}*@abo.fi*

Summary

In this paper we analyze the implications of three different factors (preprocessing method, data distribution and training mechanism) on the classification performance of artificial neural networks (ANNs). We use three preprocessing approaches: no preprocessing, normalization and division by the maximum absolute values. We study the implications of input data distributions by using five datasets with different distributions: the real data, uniform, normal, logistic and Laplace distributions. We test two training mechanisms: one belonging to the gradient-descent techniques, improved by a retraining procedure (RT), and the other is a genetic algorithm (GA), which is based on the principles of natural evolution. The results show statistically significant influences of all individual and combined factors on both training and testing performances. A major difference with other related studies is the fact that for both training mechanisms we train the network using as starting solution the one obtained when constructing the network architecture. In other words we use a hybrid approach by refining a previously obtained solution. We found that when the starting solution has relatively low accuracy rates (80-90%) GA clearly outperformed the retraining procedure, while the difference was smaller to non-existant when the starting solution had relatively high accuracy rates (95-98%). As reported in other studies we found little to no evidence of crossover operator influence on the GA performance.

Keywords: Artificial Neural Networks, Genetic Algorithms, Retraining Procedure, Financial Performance Classifications

1 Introduction

Predictive data mining has two different aims: (1) the uncovering of hidden relationships and patterns in the data, and (2) the construction of usable prediction models (Zupan *et al.* 2001). One type of prediction model is represented by classification models or models for predicting the relative positions of newly observed cases against what is known. Financial performance classification problems concern many business players: from investors to decision makers, from creditors to auditors. They are all interested in the financial performance of the company, what are its strengths and weaknesses, and how the decision process can be influenced so that poor financial performance or, worse, bankruptcy, is avoided. Usually, the classification problem literature emphasizes binary classification, also known as two-group discriminant analysis problems, which is a simpler case of the classification problem. In the case of binary classifications everything is seen in black and white (e. g.: A model which implements a binary classifier would just show a bankruptcy or a

*Correspondence to: Adrian Costea, Turku Centre for Computer Science and Institute for Advanced Management Systems Research, Åbo Akademi University, Lemminkäisenkatu 14 A, FIN-20520, Turku, Finland. Office: +358-2-2153320. Mobile: +358-40-5588513. Fax: +358-2-215-4809. *E-mail address*: Adrian.Costea@abo.fi (A. Costea) non-bankruptcy situation giving no detailed information about the real problems of the company). Greater information would be obtained from the classification models if one particular business sector would be divided into more than two financial performance classes (and it would be easier to analyze the companies placed in these classes). This study introduces an artificial neural network trained using genetic algorithms (GA-based ANN) to help solve the multi-class classification problem. The predictive performance of the GA-based ANN will be compared to a retraining-based ANN which is a new way of training an ANN based on its past training experience and weights reduction. Four different GA-based ANNs will be constructed based on four different crossover operators. At the same time, three different retraining-based ANNs are used depending on when the effective training and validation sets are generated (at which stage of the retraining algorithm). The two different training mechanisms have something in common, which is the ANN architecture. A new empirical method to determine the proper ANN architecture is introduced. In this study the solution (set of weights) determined when constructing the ANN architecture is *refined* by the two training mechanisms. The training mechanisms have an initial solution which is not randomly generated as it is in the majority of the reported related studies (Schaffer *et al.*, 1992; Sexton and Gupta, 2000; Sexton and Sikander, 2001; Pendharkar, 2002; Pendharkar and Rodger, 2004). Moreover, our study investigates the influence of data distributions and preprocessing approach on the predictive performances of the models. Very few authors have studied the implications of data distributions on the predictive performance of ANN, but in combination with other factors such as the size of the ANN (in terms of number of hidden neurons) and input data and weight noise (Pendharkar, 2002). Some studies have focused on the transformation of the input data to help increase the classification accuracy, as well as improve the learning time. For example, Vafaie and DeJong (1998) proposed a system for feature selection and/or construction which can improve the performance of the classification techniques. The authors applied their system on an eye-detection face recognition system, demonstrating substantially better classification rates than competing systems. Zupan *et al.* (1998) proposed function decomposition for feature transformation. Significantly better results were obtained in terms of accuracy rates when the input space was transformed using feature selection and/or construction. Few research papers studied different data preprocessing methods to help improve the ANN training. Koskivaara (2000) investigated the impact of four pre-processing techniques on the forecasting capability of ANNs when auditing financial accounts. The best results were achieved when the data were scaled either linearly or linearly on yearly bases.

However, after we examined Alander's paper (Alander, 1995), which contains 1760 references (from 1987 until 2003) on combining GAs and artificial neural networks, we found out that there is no report in the literature which analyzes the influence of data distribution, preprocessing method, training mechanism and their combinations on the classification performances of ANNs. For example, it is not known (1) what is the preprocessing method that is most suitable for a certain distribution when training ANNs, (2) what is the most suitable *refining* mechanism for training ANNs in terms of prediction accuracy when the data distribution is known, (3) what is the best combination preprocessing method-training technique when we already know the distribution of the input dataset, (4) what is the best crossover operator for learning the connection weights of an ANN when the preprocessing method and input data distribution are known, (5) how important is it for the retraining-based mechanism at which point in the retraining mechanism structure we split the data into effective training and validation sets.

The focus of our study is to address the questions posed above. The paper is organized as follows. In the second section we review the literature on classification models emphasizing ANN-based models for classification. Next, we introduce our model for assessing companies' financial performance. Research questions and derived hypotheses are formulated in section four. The datasets used, with descriptive statistics, are presented in section five. In the sixth section, we show our experiments' results, and finally, the conclusions and directions for future research are discussed.

2 Literature review

The problem of financial performance classification has been tackled in the literature for nearly 40 years. The taxonomy of classification models is based on the algorithm solution being used (Pendharkar, 2002). Firstly, **statistical techniques** have been deployed: univariate statistics for prediction of failures introduced by Beaver (1966), multivariate analysis in Altman (1968), linear discriminant analysis (LDA) introduced by Fisher (1936) who firstly applied it on Anderson's iris data set (Anderson, 1935), multivariate discriminant analysis (MDA) - Edmister (1972), Jones (1987), probit and logit models - Hamer (1983), Zavgren (1985), Rudolpher *et al.* (1999) and recursive partitioning algorithm (RPA) in Frydman *et al.* (1985). The next step in solving the classification problem was the establishment of **induction techniques**. Some of the most popular such techniques are: CART (Breiman *et al.*, 1984), ID3-C4.5-C5.0 (Quinlan, 1993a; Quinlan, 1993b). In (*own ref.*, 2002; *own ref.*, 2003) the authors applied and compared two of the above classifiers: multinomial logistic regression and Quinlan's C5.0 decision tree. The two classifiers performed similarly in terms of accuracy rates and outperformed SOM[1] (Kohonen, 1997) classification. Among the financial application areas of **neural networks** in the early 80s, the financial performance classification problem was not an exception. ANNs were extensively used in financial applications, the emphasis being on bankruptcy prediction. A comprehensive study on ANNs for failure prediction can be found in O'Leary (1998). The author investigates fifteen related papers for a number of characteristics: what data was used, what types of ANN models, what software, what kind of network architecture, etc. Table 1 presents a sample of studies with their results which compared different classification techniques.

Table 1: Sample of pattern classification studies

Authors	Tasks	Techniques	Results
Marais *et al.* (1984)	Modelling commercial bank loan classifications	Probit, RPA	RPA is not significantly better, especially when data do not include nominal variables
Schütze *et al.* (1995)	Document routing problem	Relevance feedback, LDA, Logistic regression, ANN	Complex learning algorithms (LDA, logistic regression, ANN) outperformed a weak learning algorithm (relevance feedback)
Jeng *et al.* (1997)	Prediction of bankruptcy, biomedical	Fuzzy Inductive Learning Algorithm (FILM), ID3, LDA	Induction systems achieve better results than LDA. FILM slightly outperforms ID3
Back *et al.* (1996a, 1997)	Prediction of bankruptcy	LDA, Logit, ANN	ANN outperformed the other 2 methods in terms of accuracy

ANNs in the form of SOMs have been extensively used in financial applications. Martín-del-Brío and Serrano Cinca (1993) propose Self Organizing Feature Maps (SOFM) as a tool for financial analysis. Among the problems associated with the use of traditional statistical models in financial analysis Serrano Cinca (1996) mentions: "the difficulty of working with complex statistical models, the restrictive hypotheses that need to be satisfied and the difficulty of drawing conclusions by non-specialists in the matter". The author proposes the SOM for predicting corporate failure and compares SOM with linear discriminant analysis (LDA) and a multilayer perceptron (MLP) trained with the backpropagation algorithm (BP). The data base contains five financial ratios taken from Moody's Industrial Manual from 1975 through to 1985 for a total of 129 firms, of which 65 are bankrupt and the rest are solvent. Serrano Cinca (1998a, 1998b) extended the scope of the

[1] SOM - Self-Organising Map was introduced by Kohonen in early 80's and is an unsupervised learning technique that creates a two-dimensional topological map from n-dimensional input data. A topological map is a mapping that preserves neighborhood relations. Similar input vectors have close positions on the map.

Decision Support System proposed in the earlier studies by addressing, in addition to corporate failure prediction, problems such as: bond rating, the strategy followed by the company in relation to the sector in which it operates based on its published accounting information, and the comparison of the financial and economic indicators of various countries. Deboek (1998) outlines 12 financial, 4 economic and 5 marketing applications of the SOM. Another major SOM financial application is Back *et al.* (1998) which is an extended version of Back *et al.* (1996b) were the authors analyze and compare 120 pulp-and-paper companies between 1985 and 1989 based on their annual financial statements. Eklund *et al.* (2003) propose SOM as an alternative data mining technique for financial benchmarking of world-wide pulp-and-paper companies. Karlsson (2002) used SOM to analyze and compare the companies from the telecommunication sector.

Koskivaara (2004) summarizes the ANN literature relevant to auditing problems. She concludes that the main auditing application areas of ANNs are as follows: material error, going concern, financial distress, control risk assessment, management fraud, and audit fee which are all, in our opinion, particular cases of classification problems. In other words, in these applications ANNs were used, mainly, as classifiers. Going concern and financial distress can be considered to be particular cases of bankruptcy prediction.

Own ref. (2004) compared three classifiers for financial performance classification of telecom companies and found that the ANN performed similarly in terms of accuracy rates as statistical and induction techniques.

Coakley and Brown (2000) classified ANN applications in finance by the parametric model used, the output type of the model and the research questions.

Another technique to learn the connection weights for an ANN corresponds to the **evolutionary approach** and is represented by genetic algorithms. The literature in this area is relatively rich: Schaffer *et al.* (1992) listed 250 references that combined ANNs and genetic algorithms. GAs are used in the majority of these papers for solving the following problems: to find the proper architecture for the ANN, reduce the input space to the relevant variables, and as an alternative way of learning the connection weights. One paper that uses GAs to solve the last two forementioned problems is Sexton and Sikander (2001). The GA was found to be an appropriate alternative to gradient-descent-like algorithms for training neural networks and, at the same time, the GA could identify relevant input variables in the data set.

Yao (1999) explores the possible benefits of combining ANNs and evolutionary algorithms (EAs). EAs refers to a class of population-based stochastic search algorithms such as evolution strategies (ESs), evolutionary programming (EP) and genetic algorithms (GAs) that are based on principles of natural evolution (Yao, 1999, pp. 1424). Yao presents different combinations between ANNs and EAs such as: evolution of ANN connection weights, evolution of ANN architectures and evolution of ANN learning rules. Through a large literature review, the author shows that the combinations of ANNs and EAs can lead to better models and systems than relying on ANNs or EAs alone. Yao (1999, pp. 1427) presents tens of papers where one of the two training mechanisms (evolutionary algorithms and gradient-descent-like algorithms) was found to achieve better results than the other, and attributes these contradictory results to “whether the comparison is between a classical binary GA and a fast BP algorithm, or between a fast EA and a classical BP algorithm.(...) The best one is always problem dependent”. Yao and Liu (1997) proposed a new evolutionary system - EPNet - for evolving ANNs. The authors use evolutionary programming for evolving simultaneously ANN architecture and connection weights. The negative effect of the permutation problem[2] (Hancock, 1992) was avoided by simply not using crossover operators. EPNet uses 5 different mutations: hybrid training, node deletion, connection deletion, node addition and connection addition. The goal of each mutation is to obtain better offsprings. Firstly, EPNet uses BP to train the network, then the simulated annealing (SA) algorithm is used in training and if the network is improved above some threshold the mutation stops. Otherwise, other mutations are applied gradually (Yao and Liu, 1997, Fig. 5, pp. 6). EPnet was applied on a number of experiments (N-parity problem, the two-spiral problem, four medical diagnosis problems, the Australian credit card assessment problem, and the Mackey-Glass time series prediction problem)

[2] The permutation problem occurs because two ANNs with different architectures can have the same performance. In other words, eventhough the two genetic representations of the networks are different, the networks have the same functionality. The permutation problem makes crossover operator very inefficient and ineffective since with this operator - permutation of hidden nodes - functionally equivalent networks are obtained (Yao, 1999, pp. 1426).

which show that EPNet can discover ANNs that would be difficult to design by human beings. However, EPNet is suitable for applications where the time factor is not crucial, since " ...it searches a much larger space than that searched by most other constructive or pruning algorithms ...." (Yao and Liu, 1997, pp. 20).

Fogel *et al.* (1995, 1998) used ANNs trained with evolutionary algorithms to analyze interpreted radiographic features from film screen mammograms. The results show that even small ANNs (with 2 hidden nodes and small number of important features) can achieve comparable results with much more complex ones. These small networks provide "an even greater chance of explaining the evolved decision rules that are captured by the ANNs, leading to a greater acceptance by physicians" (Fogel *et al.*, 1998). Chellapilla and Fogel (1999) combined ANNs and evolutionary algorithms to learn appropriate and, sometimes (e.g.: checkers) near-expert strategies in zero and nonzero-sum games such as iterated prisoner's dilemma, tic-tac-toe, and checkers.

Many authors (e.g. Schaffer, 1994) found that GA-based ANNs are not as competitive as their gradient-descent-like counterparts. Sexton *et al.* (1998) argued that this difference has nothing to do with the GA's ability to perform the task, but rather with the way it is implemented. The candidate solutions (the ANN weights) were encoded as binary strings which is both unnecessary and unbeneficial (Davis, 1991 and Michalewicz, 1992) when the ANN has a complex structure. The tendency is toward using non-binary (real) values for encoding the weights.

There are few research papers that studied the implications of data distributions on the predictive performance of ANN. Bhattacharyya and Pendharkar (1998) studied the impact of input distribution kurtosis and variance heterogeneity on the classification performance of different machine learning and statistical techniques for classification. Pendharkar (2002) studied the application of a non-binary GA for learning the connection weights of an ANN under various structural design and data distributions, finding that additive noise, size and data distribution characteristics play an important role in the learning, reability and predictive ability of ANNs. Pendharkar and Rodger (2004) studied the implications of data distributions determined through kurtosis and variance-covariance homogeneity (dispersion) on the predictive performance of GA-based and gradient-descent-based ANN for classification. Also, Pendharkar and Rodger (2004) studied the implication of three different type of crossover operator (one-point, aritmetic, and uniform crossover) on the prediction performance of GA-based ANN. No significant difference was found between the different crossover operators. However, GAs based on uniform and aritmetic crossover performed differently at a level of significance of 0.1, suggesting that there might be a statistically significant difference for larger networks (Pendharkar and Rodger, 2004). In Section 4 we present how our study differs from above mentioned ones.

Neural network training can be made more efficient if certain preprocessing steps are performed on the network inputs and targets (Demuth and Beale, 2001). In Zupan *et al.* (1998) the authors proposed a classification technique (HINT) which is based on function decomposition for the transformation of input feature space. The idea is to separate the input space in two less complex disjoint feature spaces that recombined yield the original input feature space. The original input feature space can be reduced if one of the two disjoint feature spaces has redundant features. Zupan *et al.* (1998) used as case studies two well-known machine-learning problems (monk1, monk2) and a housing-loan allocation problem. The authors compared their system (HINT) with Quinlan's C4.5 decision tree algorithm in terms of prediction accuracy, finding out that for all the above problems the system based on function decomposition yielded significantly better results.

In this study we discuss the effect of three factors (data distribution, preprocessing method and training mechanism) and their combinations on the prediction performance of ANN-based classification models. There is no research literature (Alander, 1995) that studied the combined impact of the above mentioned factors on ANN classification performance. This study tries to fill this gap in the literature. We compare two different ANN training mechanisms for pattern classification: one based on traditional gradient-descent training algorithms (RT-based ANN) and another one based on natural selection and evolution (GA-based ANN). We also propose an empirical procedure to determine the ANN architecture which is kept fix for both training mechanisms. The starting solution (initial set of weights) for both training mechanisms is obtained when we determine the ANN architecture. We reveal classes of financial performance for the companies in the telecommunication sector based on profitability, liquidity, solvency and efficiency financial ratios. These ratios are suggested in Lehtinen's (1996) study of the reliability and validity of financial ratios in international comparisons.

3 Financial performance classification models

In this section we present our two approaches for financial performance classifications. We describe two financial ANN classification models which differ based on the training mechanism that they use. The first model is based on gradient-descent-like training mechanisms (RT-based ANN classification model), and the other is based on the principles of natural evolution (GA-based ANN classification model). In Section 6 we apply these two models on 15 different datasets (one for each data distribution-preprocessing method combination). In section 5 we describe how the different datasets with different distributions and preprocessing methods were obtained. In this section, we firstly present for each of the datasets obtained the preliminary steps necessary to build the class variable and to obtain the training and test datasets. Then, we present the empirical procedure for determining the ANN architecture. Finally, we describe the two ANN training mechanisms.

The generic classification model based on neural approaches is depicted in Figure 1.

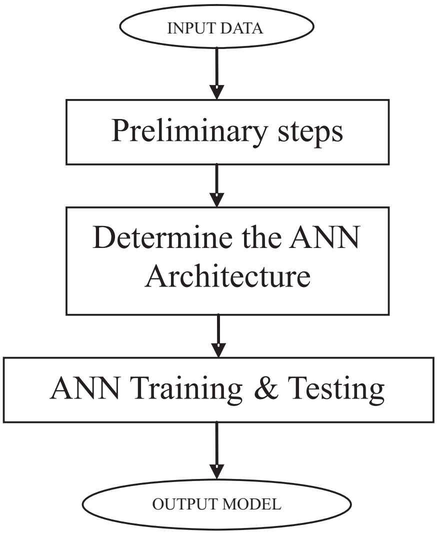


Figure 1: ANN generic classification model

Usually, when constructing classification models, the first step is to separate the data into training (*TR*) and test (*TS*) sets. In case the class variable is missing, as in our case, a clustering method could be applied to build this variable (section 3.1). The second step consists of selecting the proper ANN architecture (section 3.2). This step is concerned with determining the proper number of hidden layers, and the right number of neurons in each hidden layer. Also, here we decide how the class variable should be codified. In other words, how many neurons are necessary on the output layer to represent the class variable? The last step, ANN training, consists of specific tasks depending on the training mechanism used (sections 3.3, 3.4).

3.1 Preliminary steps

Next, we present the steps undertaken to create the training and test sets for the classification models, which we generically call preliminary steps:

1. A clustering technique was applied to build the class variable for each dataset. We have used the fuzzy C-means (FCM) clustering algorithm (Bezdek, 1981) to build the clusters and, consequently, the class variable. The number of clusters is a parameter of our models. It was set to 7 as this was the proper number of classes reported in our previous studies (*own ref.*, 2003).

2. In order to allow the ANN to equally learn the patterns within each cluster we chose an equal number of observations for each cluster.

3. Finally, we split the data into aproximately 90% for training and the remainder for testing.

As was described above, we reduce as much as possible the subjectivity in determining the class variable by applying directly FCM clustering algorithm. When fuzzy clustering algorithms are applied every observation gets a vector representing its membership degree in every cluster, which indicates that observations may contain, with different strengths, characteristics of more than one cluster. Usually, the elements of the data set are assigned to the cluster that has the highest membership degree. In spite of the additional information provided by the methodology, there is a problem with the observations that are difficult to position (uncertain observations) when they obtain similar membership values for two or more clusters. *Own ref.* (2004) introduces a modified version of FCM algorithm to allocate the uncertain observations by introducing weights when calculating distances to the clusters' centers. The authors compare the modified version of FCM algorithm with normal FCM and SOM clustering. The modified FCM algorithm outperformed both the normal FCM and the SOM with respect to pattern classification. In this study, normal FCM was chosen for practical implementation reasons. We created for each financial ratio a linguistic variable that can help us in characterizing the clusters. Linguistic variables are quantitative fuzzy variables whose states are fuzzy numbers that represent linguistic terms (Klir and Yuan, 1995). *Own ref.* (2004) model the seven financial ratios with the help of seven linguistic variables using five linguistic terms: *very low* (VL), *low* (L), *average* (A), *high* (H), *very high* (VH). Table 2 shows the characterization of the seven clusters for the real telecom dataset without preprocessing the data (first preprocessing approach).

Table 2: Characterization of Clusters[a]

	OM	ROTA	ROE	Current	E to C	IC	Rec. T.	Order
Cluster 1	VL	VL	VL&L	-	A&H	VL&L	-	Bad
Cluster 2	A	A	A&H	-	A	A	-	Average
Cluster 3	VL&L	VL	VL	-	VL&L	L	-	Worst
Cluster 4	H	H	VH	VL	A	A	A	Good
Cluster 5	A	A	A&H	H	H	VH	-	Good
Cluster 6	L	L	A	L	L	L	-	Bad
Cluster 7	VH	VH	H	VH	VH	VH	-	Best

[a] *Own ref.* (2004)

We considered that one linguistic term characterizes one cluster if it represents more than 40 % out of total number of observations for that cluster. It seems that one of the ratios - Receivables Turnover (RT) - does not have any discriminatory power among data except for one cluster. By comparing the clusters we can easily label them as being good, bad, worst, etc. depending on their linguistic terms.

3.2 Empirical procedure for determining the ANN architecture

Once the data is ready to be trained, we need to find a suitable architecture for the ANN. Choosing the number of hidden layers and the number of neurons in each hidden layer is not a straight-forward task. The choices of these numbers depend on "input/output vector sizes, size of training and test subsets, and, more importantly, the problem of nonlinearity" (Basheer and Hajmeer, 2000, pp. 22). It is well known that neural networks are very sensitive regarding the dimensionality of the dataset (Hagan *et al.*, 1996; Basheer and Hajmeer, 2000; Demuth and Beale, 2001). Basheer and Hajmeer (2000) cites a number of papers that introduce different rules of tumbs that link the number of hidden neurons (NH) with the number of input (NI) and output (NO) neurons or with the number of training samples (N_{TRN}). One rule of thumb, proposed in Lachtermacher and Fuller (1995) suggests that the number of hidden neurons NH for one output ANN is: $0.11N_{TRN} \leq NH(NI+1) \leq 0.30N_{TRN}$. Upadhyaya and Eryurek (1992) related the total number of weights N_w with the number of training samples: $N_w = N_{TRN}log_2(N_{TRN})$. Masters (1994) proposed that the number of hidden neurons on the hidden layer should take values in the vicinity of the geometric mean of number of inputs (NI) and number of outputs (NO). Taking Basheer and Hajmeer (2000, pp. 23) advice that "the most popular approach to finding the optimal number of hidden nodes is by trial and error with one of the above rules" we chose Masters' rule of tumb as a starting point to develop our ANN architectures. Concerning the number of hidden layers, we performed a number of experiments for ANN architectures with one and two hidden layers to see what the appropriate number of hidden layers is. Almost in every case, an ANN with two hidden layers performed better in terms of training mean square error. Beside our own experiments with the financial dataset, we based our choice of two hidden layers for the ANN architecture on the architecture we found for the prediction of glass manufacturing process variables reported in *Own Ref.* (2004), and on what was previously reported in literature. The two hidden layer ANNs performed better than single hidden layer ANNs in the examples from Hartman and Keeler (1991), Lönnblad *et al.* (1992), and Ohlsson *et al.* (1994). Concerning the problem of choosing one or two hidden layers Chester (1990) argues that "... an MLP with two hidden layers can often yield an accurate approximation with fewer weights than an MLP with one hidden layer" and that "the problem with a single hidden layer is that the neurons interact with each other globally, making it difficult to improve an approximation at one point without worsening it elsewhere". We did not take into consideration three hidden layer cases due to the number of cases per weights ratio-restriction.

We used the sigmoid and linear activation functions for the hidden and output layers, respectively. Regarding the training algorithms, they fall into two main categories: heuristic techniques (momentum, variable learning rate) and numerical optimization techniques (conjugate gradient, Levenberg-Marquardt). Various comparative studies, on different problems, were initiated in order to establish the optimal algorithm (Demuth and Beale, 2001; *own ref.*, 2003). As a general conclusion, it is difficult to know which training algorithm will provide the best (fastest) result for a given problem. A smart choice depends on many parameters of the ANN involved, the data set, the error goal, and whether the network is being used for pattern recognition (classification) or function approximation. Statistically speaking, it seems that numerical optimization techniques present numerous advantages. Analyzing the algorithms that fall into this class, we observed that the Scaled Conjugate Gradient (SCG) algorithm (Moller, 1993) performs well over a wide variety of problems, including the experimental dataset presented in this paper. Even if SCG is not the fastest algorithm (as Levenberg-Marquardt in some situations), the great advantage is that this technique works very efficiently for networks with a large number of weights. The SCG is something of a compromise: it does not require large computational memory, and yet, it still has a good convergence and is very robust. Furthermore, we always apply the early stopping method (*validation stop*) during the training process, in order to avoid the over-fitting phenomenon. And it is well known that for early stopping, one must be careful not to use an algorithm that converges too rapidly (Hagan *et al.*, 1996; Demuth and Beale, 2001). The SCG is well suited for the validation stop method.

In our experiments we have kept all parameters of the ANN constant (the learning algorithm - Scale Conjugate Gradient, the performance goal of the classifier, the maximum number of epochs), except the numbers of neurons in the hidden layers (NH_1, NH_2).

The procedure used to determine the proper values for NH_1 and NH_2 consists of iteratively performing the following experiment:

- Randomly split the training set (TR) into two parts: one for the effective training (TRe) and the other for validation (VAL). In order to avoid the over-fitting phenomenon we have applied the early stopping method (*validation stop*) during the training process.

- Train the network for different values of NH_1 and NH_2. For each combination of NH_1 and NH_2, we performed 4 random initializations of the weights. NH_1 and NH_2 take values in the vicinity of the geometric mean (Masters, 1994) of the number of inputs (NI) and outputs (NO), respectively.

$$\sqrt{NI \cdot NO} - 2 \leq NH_i \leq \sqrt{NI \cdot NO} + 2$$

E.g.: $NI = 7, NO = 7 \Rightarrow NH_1, NH_2 = \overline{5, 9}$. In this case, in total 5*5*4 = 100 trainings are performed for each experiment.

- Save the best ANN architecture in terms of mean square error of the training set (MSE_{TRe}) with the supplementary condition: $MSE_{VAL} \leq (6/5) * MSE_{TRe}$. This supplementary condition was imposed so that the validation error is not too far from the training error, thus, avoiding over-fitting for the test set.

We ran 3 experiments like the one described above (3*100 = 300 trainings) to determine the proper values for NH_1 and NH_2. See the flowchart of the procedure in Appendix A - Figure 3.

Regarding the number of output neurons, we have two alternatives when applying ANNs for pattern classification. The first alternative, which is the most commonly used, is to have as many output neurons as the number of classes. The second alternative is to have just one neuron in the output layer, which will take the different classes as values. We chose the first approach in order to allow the network to better disseminate the input space.

After we performed the 3 experiments we obtained the best ANN architecture and the set of final weights (the solution) that corresponds to this architecture. In the next two sections we will present two training mechanisms used to *refine* this solution.

3.3 RT-based ANN training

Once we determine the ANN architecture (with the corresponding set of weights), the next step is to train the network. The first training mechanism is a retraining-based ANN (*own ref.*, 2004), briefly described next:

- Start with a network with an initial set of weights from the previous step (Determining ANN architecture) as the reference network;
- Perform L runs to improve the ANN classification accuracy. After each experiment we save the best set of weights (the solution) in terms of classification accuracy. Each experiment consists of:
  - Reduction of the weights of the current best network with successive values of scaling factor γ (γ = 0.1, 0.2, ..., 0.9).
    - Retrain the ANN with the new weights and obtain 9 accuracy rates.
  - Choose the best network from the above 9 in terms of classification accuracy.
  - Compare the accuracy rate of the current network with that obtained in the previous step and save the best one for the next run as the current best network.

Depending on the splitting of the training set (*TR*) in the effective training set (*TRe*) and validation set (*VAL*) we have 3 types of retraining mechanisms: one (RT1) where *TRe* and *VAL* are common for all of the L runs, another (RT2) where *TRe* and *VAL* are different for each run, but the same for all 9 reduction weights trainings (second step of the experiment), and finally, RT3 where *TRe* and *VAL* are distinct for each training. We have 4 types of accuracy rates: training accuracy rate (ACR_{TRe}), validation accuracy rate (ACR_{VAL}), total training (effective training + validation) accuracy rate (ACR_{TR}) and test accuracy rate (ACR_{TS}). Correspondingly, we calculate 4 mean square errors: MSE_{TRe}, MSE_{VAL}, MSE_{TR}, and MSE_{TS}. In total 5 runs ($L = 5$) were conducted resulting in 5*9 = 45 new trainings for each type of retraining mechanism. Each retraining mechanism needs aproximatively 30 minutes to complete. Consequently, we need 0.5*3=1.5 hours/experiment to run all 3 retraining mechanisms.

3.4 GA-based ANN training

The second ANN training mechanism used to *refine* the solution is based on the principle of natural evolution. A population of solutions is provided, and by initialization, selection and reproduction mechanisms, potentially good solutions are reached.

Unlike the traditional gradient-descent training mechanisms, GA-based ANN training starts with a population of solutions. A solution is the set of ANN weights after training represented as a vector. All solutions (chromosomes) compete with each other to enter the new population. They are evaluated based on the objective function.

3.4.1 Initialization and fitness evaluation

The population size is a parameter of our models. It was set to $PS = 20$ for three reasons: firstly, Dorsey and Mayer (1995) suggests that this value is good enough for any grade of problem complexity, secondly, the population size increases by adding new chromosomes with both crossover - $PS' > PS$ - and mutation operators - $PS'' > PS' > PS$ - (after the new population is evaluated we resize the population to the initial size by keeping the best PS chromosomes in terms of ACR_{TR} and discarding the others), and thirdly, due to the fact that as the population size increases, the running time of our GA-based algorithms becomes unfeasibly high. Even with a small initial population of 20 chromosomes, one running of GA-based refining mechanism (1000 generations) takes up to 2 hours. If we multiply this with 600 we get a total of 1200 hours for training all GA-based ANNs[3]. For details, see section 6.

The first chromosome of the population is the set of weights obtained when determining the ANN architecture. The other 19 chromosomes are generated by training the ANN with the previously obtained architecture. Afterwards, the first generation of the algorithm may begin. The number of generations is related with the empirical formula suggested in Ankenbrandt (1991). The number of generations for a non-binary GA without mutation is given by the formula: $N_{gen} = \ln[(n-1)^2]/\ln(r)$ where n is the population size and r is the average fitness of candidates with a particular gene value over the average fitness of all other candidates.

Each chromosome is evaluated using the accuracy rate for the training set (ACR_{TR}).

3.4.2 Selection

Firstly, the elitism technique is applied in the sense that the best N_{elite} chromosomes in terms of ACR_{TR} are inserted into the new population. The rest of the chromosomes ($20 - N_{elite}$) are selected based on the probability of selection (*roulette wheel* procedure) for each chromosome: $P_i = ACR_{TR_i} / \sum_{i=1}^{20} ACR_{TR_i}$

The higher the probability P_i for a chromosome is, the higher its chance of being drawn into the new population. This procedure tries to simulate the process of natural selection or survival of the fittest. We decided to employ elitist selection in our algorithms as a consequence of what was reported in the literature. For example, Rudolph (1994) proves by means of homogeneous finite Markov chains that GAs converge probabilistically to the global optimum only when elitist selection is used (the best individual survives with probability one). Miller and Thomson (1998) uses GAs to evolve digital circuits with a new chromosome representation and finds out that “without elitism the GA struggled to find any fully correct solutions for what is essentially a very simple circuit, but with elitism the results were markedly improved”. Shimodaira (1996) develops a GA with large mutation rates (controlled by a decreasing function of generation) and elitist selection - GALME - and finds out that the performance of GALME is remarkably superior to that of traditional GA. Fogel *et al.* (2004) applies evolutionary algorithms for similar RNA structure discovery and focuses on the optimization of population and selection parameters. The study compares elitist selection with three different tournament selections (tournament size 5, 10, and 20) and finds out that the increased tournament size increases the variance in the mean convergence and that tournament size 5 achieved slightly better mean variance than elitist selection. However, the number of clients (workstations) that arrived at corect solutions was roughly similar when elitist and 5 size tournament selection were employed.

Next, 80% (probability of crossover: $P_c = 0.8$) of the chromosomes obtained previously are randomly selected for mating. The choice of crossover probability as well as the other GA parameters (mutation probability, population size) is more art than science. Tuson and Ross (1998)

[3] 600 = 10 (number of runs for each GA experiment) * 4 (number of crossover operators) * 3 (number of preprocessing methods) * 5 (number of different distributional data sets).

compared the performance of non-adaptive GAs (GAs with fixed crossover and mutation probabilities) with the performance of adaptive GAs (GAs that use operators' adaptation) and founds out that "...at least for the problems used here, adaptation by the co-evolution of operator settings is of limited practical use, except in situations where the tuning process is sufficiently time limited." The authors suggested that the proper choice of the crossover in the case of non-adaptive GAs depend upon the population model, the problem to be solved, its representation and the performance criterion being used. For example, for "Deceptive" problem "a low crossover probability gives high quality results, whereas a high crossover probability exchanges solution quality for a higher speed of search" (Tuson and Ross, 1998). Rogero (2002) finds out that an increase on the crossover probability above 0.3 does not improve the convergence speed of the GA. However, the author mentions that this value is problem dependent. The problem with choosing high crossover probabilities is that potentially very performant parents would be removed from the population. This is not the case for our GA implementation, since after reproduction we increase the population to include both the parents and their offsprings. In other words, the probability of crossover is not essential for the performance of our algorithm as long as it has a high value.

3.4.3 Reproduction

The selected chromosomes are randomly paired and recombined to produce new solutions. There are two reproduction operators: *crossover* and *mutation*. With the first the mates are recombined and new born solutions inherit information from both parents. With the second operator new parts of the search space are explored and, consequently, we expect that new information is introduced into the population. In this study we have applied four types of crossover: arithmetic, one-point, multi-point and uniform crossover. Let us denote with L the length of the chromosomes and with P_1 and P_2 two parent-chromosomes:

$$\begin{aligned} P_1 &= g_{11}, g_{12}, \ldots, g_{1L} \\ P_2 &= g_{21}, g_{22}, \ldots, g_{2L} \end{aligned}$$

One-point crossover
For each pair of chromosomes we generate a random integer $X, X \in \{1, L\}$. The two new born children are constructed as follows:

$$\begin{aligned} C_1 &= g_{11}, g_{12}, \ldots, g_{1X}, g_{2,X+1}, \ldots, g_{2L} \\ C_2 &= g_{21}, g_{22}, \ldots, g_{2X}, g_{1,X+1}, \ldots, g_{1L} \end{aligned}$$

Multi-point crossover
We split the chromosomes in n parts ($n \leq 5$). We generate randomly the number of splitting points n. Then, n distinct random numbers $(X_1, X_2, \ldots, X_n)$ are generated with $X_i \in \{1, L\}$ and $X_1 < X_2 < \ldots < X_n$. The two children are:

$$\begin{aligned} C_1 &= g_{11}, g_{12}, \ldots, g_{1X_1}, g_{2,X_1+1}, \ldots, g_{2X_2}, g_{1,X_2+1}, \ldots, g_{1X_3}, g_{2,X_3+1}, \cdots \\ C_2 &= g_{21}, g_{22}, \ldots, g_{2X_1}, g_{1,X_1+1}, \ldots, g_{1X_2}, g_{2,x_2+1}, \ldots, g_{2X_3}, g_{1,x_3+1}, \cdots \end{aligned}$$

Arithmetic crossover
Firstly, we split the parent-chromosomes in n parts as we did for multi-point crossover. The children' genes are convex combinations of the parents' genes.

$$C_1 = \begin{cases} \alpha * g_{1i} + (1-\alpha) * g_{2i}, & i = \overline{1, X_1} \\ (1-\alpha) * g_{1i} + \alpha * g_{2i}, & i = \overline{X_1 + 1, X_2} \\ \alpha * g_{1i} + (1-\alpha) * g_{2i}, & i = \overline{X_2 + 1, X_3} \\ \ldots \end{cases}$$

$$C_2 = \begin{cases} (1-\alpha) * g_{1i} + \alpha * g_{2i}, & i = \overline{1, X_1} \\ \alpha * g_{1i} + (1-\alpha) * g_{2i}, & i = \overline{X_1 + 1, X_2} \\ (1-\alpha) * g_{1i} + \alpha * g_{2i}, & i = \overline{X_2 + 1, X_3} \\ \ldots \end{cases}$$

where $\alpha \in [0, 1]$ is a random number and is generated for each chromosome-pair.

Uniform crossover
For each pair of genes of the parent-chromosomes we generate a random number $\alpha \in [0, 1]$. If $\alpha < 0.5$ the gene of the first parent goes to the first child and the gene of the second parent goes to the second child. Otherwise, the genes are inversed.

The children-chromosomes are *added* to the population. The size of the population becomes $PS' > PS$. Next we apply the mutation operator for all the chromosomes in PS'. We used only uniform mutation.

Uniform mutation
The probability of mutation is set to $P_m = 0.01$ which means that approximately 1% of the genes will mutate for each chromosome. An $\alpha \in [0, 1]$ is generated for each gene of each chromosome and if $\alpha \leq P_m$, the new gene is randomly generated within the variable domain. Otherwise, the gene remains the same. If at least one gene is changed then the new chromosome is added to the population, obtaining $PS'' > PS' > PS$. As in the case of crossover probability, the proper setting of mutation probability depends on the population model, the problem to be solved, fitness function (Tuson and Ross, 1998). Tuson and Ross (1998) founds no difference between fixed and adapted mutation rates: when the initial mutation rate was "close to theoretically optimal value - 3/chrom length" then "the speed to solution was improved". DeJong (1975) considers mutation probability to be inversely proportional to the population size. Hesser and Männer (1990) includes in the calculation of mutation probability both population size and chromosome length. Hoehn (1998) introduced mutation at both parental and offspring levels and implemented four GAs based on the mutation probabilities for the two levels: standard GA (no mutation for parental level and .001 for offspring level), low GA (.001 for both parental and offspring levels), high GA (.1 for parental and .001 for offspring levels), and variable GA (from .001 to .1 for parental and .001 for the offspring levels). The four GAs were compared in terms of their performances in optimizing De Jong's (DeJong, 1975) functions F1-F5 (Hoehn, 1998, pp. 222). The author finds out that introducing parental mutation is generally advantageous when compared to the standard GA with only offspring mutation. In our experiments we used both parental and offspring mutation by applying mutation on both parents and their offsprings. This operation was possible since after we apply crossover operation we *add* the new chroms (offsprings) to the population and keep their parents. Consequently, the mutation is applied at both levels: parental and offspring levels. Hoehn (1998) gives us an idea of what constitutes a very low mutation probability (.001) and a very high one (.1), but his results do not help in choosing between low and high mutation probabilities. For some of DeJong's functions (F3, F4) GA with low mutation rate performed better than GA with high mutation rate, while for others (F2) it was the opposite. Correspondingly, throughout our experiments we used a "moderate" mutation probability (.01), the choice of which was, also, based upon "theoretically optimal value - 3/chrom length" (Tuson and Ross, 1998) since our chroms' lenghts vary (depending on the dataset used) around value 200 ($3/200 \approxeq 0.1$).

The final step in constructing the new population is to reduce it in size to 20 chromosomes. We select from PS'' the best 20 chromosomes in terms of ACR_{TR} satisfying the condition that one chromosome can have no more than max_lim duplicates. We use the mutation operator to generate more chromosomes in the case that the number of best chromosomes which satisfy the above condition is less than 20.

As a summary, excluding the crossover, the parameters of our GA models are as follows: number of generations (N_{gen}), population size (PS), number of elite chromosomes (N_{elite}), maximum number of splitting points (max_split) in the case of multi-point crossover, probability of crossover (P_c), probability of mutation (P_m), and maximum number of duplicates for the chromosomes (max_lim). There were around 1000 generations ($N_{gen} = 1000$) which took aproximately 2 hours to complete for each GA-based refining mechanism. As we had different retraining mechanisms, we had different GAs (4) but, this time, based on the type of crossover operator used. Consequently, we need 2*4 = 8 hours/experiment to run all 4 GAs.

4 Research questions and derived hypotheses

The main advantages of neural approaches for classification over the traditional ones are: ANNs are free of any distributional assumptions, are universal approximators, no problems with intercorrelated data, and they provide a mapping function from the input to the outputs without any a priori knowledge about the function form (function approximation capability). The most popular ANN learning technique in the literature is back-propagation (BP), which is "an approximate steepest descendent algorithm"(Hagan *et al.*, 1996) for feedforward neural networks. BP has several limitations, the most important one being its scalability: as the size of the training problem increases, the training time increases non-linearly (Pendharkar and Roger, 2004). When the basic BP is applied to a practical problem, the training may take a relatively long time (Hagan *et al.*, 1996). Among other limitations: the difficulty of the training data itself, handling the outliers, and reduced power of generalization due to large solution space. The cause for the last limitation could be the fact that the BP algorithm is likely to quickly get stuck in a local optimum, which means that the algorithm depends strongly on the initial starting values. As we described in section 3.2 many techniques have been proposed to decrease the learning time of BP and to ignore shallow local minimum. SCG was used for ANN training throughout this study.

The difference between BP/BP-variants and GA-based ANN training techniques is that BPs start from one solution and try to improve it based on some error minimization technique, while GAs start with a population of solutions and through some initialization, reproduction and recombination methods tries to reach a solution. GAs are known as hill climbing techniques, a capability that arises from the convex combination (*arithmetic crossover operator*) of two parents on the opposite sides of a hill. Moreover, the possible risk of reaching a local optimum is avoided by the GA since it creates new solutions by altering some elements of the existing ones (*mutation operator*), hence, widening the search space.

We test two training mechanisms: one based on a traditional gradient-descent technique improved by a retraining procedure (RT), and the other on genetic algorithms (GA). Moreover, we analyze the influence of the crossover operator on the predictive performance of genetic algorithms.

A crucial step in ANN training is the pre-processing of the input data. Pre-processing can be performed in two ways: one way is to apply the pre-processing technique for each individual input variable obtaining the same dimensionality of the input dataset, and the other is to apply a transformation on the whole input dataset, at once, possibly obtaining a different dataset dimensionality. The second way of pre-processing is applied when the dimension of the input vector is large, there are intercorrelations between variables and we want to reduce the dimensionality of the data and uncorelate the input. The former way of pre-processing deals with two comparability issues regarding the input variables. Firstly, each variable has to have the same importance in the training process. For that we could scale all variables so that they always fall within a specified range. Secondly, the dispersion of the variables should be the same for all variables, so that the impact of variables' dispersion on ANN training is the same for all variables. In our study we use three preprocessing approaches: no preprocessing which does not take into consideration any of the comparability concerns, division with the maximum absolute values which handles the first comparability issue and normalization which addresses both comparability issues. In this study we test whether the choice of the pre-processing approach for individual variables has any impact on the predictive performance of the ANN.

One of the goals of this study is to find out whether the combination of pre-processing approach and input data distribution has an impact on the ANN classification performance. At the same time, we are interested whether the data distribution has any influence on the choice of training technique when ANNs are applied in financial classification problems. In other words, does the data distribution - training mechanism combination have any impact on the ANN classification performance? Consequently, data with different distributions has to be generated. Some authors (e.g. Bhattacharyya and Pendharkar, 1998; Pendharkhar, 2002; Pendharkar and Rodger, 2004) studied the impact of the data distributions through kurtosis and variance-covariance homogeneity on the classification performance of ANNs. In aforementioned studies, the authors used fictive datasets drawn from uniform, normal, logistic and Laplace distributions arguing that they roughly correspond to the following kurtosis values: -1, 0, 1, 3. In line with above research we study the implications of input data distributions by using five datasets with different distributions: the real data, uniform, normal, logistic and Laplace distributions. We show in section 5 the descriptive statistics including kurtosis and skewness values for the financial ratios of real telecom dataset. We used the characteristics of the real data to derive four fictive datasets with uniform, normal, logistic and Laplace distributed data.

In this study we analyze the implications of three different factors (preprocessing method, data distribution and training mechanism) and their combinations on the classification performance of neural networks.

We compared our research questions with what was previously reported in the literature (e.g.: Pendharkar and Rodger, 2004). However there are some important differences in the assumptions in our study compared with the others:

- The main difference is that here GA and gradient descent methods are used to *refine* the classification accuracy of an already obtained ANN-based solution for the classification problem. Both the GA and the RT-based ANNs start from a solution provided when determining the ANN architecture and they try to *refine* it. All other studies compared GA and gradient-descent methods starting from random solutions. We expect that the GA-based ANN will outperform the RT-based ANN in refining what the ANN already learned due to the GA's better searching capabilities.

- The second main difference is the type of the classification problem itself. Here we are interested in separating the input space into more than 2 parts (e.g. 7 financial performance classes) providing more insights in the data.

- We are interested if the combination of preprocessing approach, distribution of the data, and training technique has any impact on the classifiers' predictive performances.

- Here non-parametric statistical tests are used to validate the hypotheses. Only t-tests or ANOVA were used in the other studies, but no evidence of satisfaction of the assumptions was provided. We performed a 3-way ANOVA to strengthen the results of the non-parametric tests.

- Also four different crossover operators are used in order to find whether this operator has an influence on the GA's predictive performance. We introduce one crossover operator - multi-point crossover - in addition to the three crossover operators presented in Pendharkar and Rodger (2004).

The first difference has an impact on all the hypotheses that we formulate in this study since, here, there is a different problem. The GA and RT-based ANNs improve an already existing solution and do not construct it from scratch. Their behavior depends on how that solution was obtained (using what kind of method).

Based on the above discussion we formulated the following hypotheses:

H1. The training mechanism used to refine the solution obtained when determining the ANN architecture will have an influence on the classification performance of ANNs. The GA-based ANN will outperform the RT-based ANN both in training and testing in the refining process.

H2. Data preprocessing will have an influence on both RT and GA-based ANN training and testing performances.

H3. Data distribution will have an influence on both RT and GA-based ANN training and testing performances.

Additional hypotheses:

H4. The crossover operator will have an influence on GA-based ANN training and testing performances.

H5. The stage at which we generate the effective training and validation sets will have an influence on RT-based ANN training and testing performances.

The main hypothesis of our paper is formulated as follows:

H6. All binary and ternary combinations of the above three factors (training mechanism, pre-processing method and data distribution) will have an influence on both RT and GA-based ANN training and testing performances.

5 Datasets and descriptive statistics

Telecommunications sector dataset.

We used financial data about worldwide telecom companies. There are 88 companies structured in five groups: US (32), Europe except Scandinavian companies (20), Asia (20), Scandinavia (10), and Canada (6). The time span is 1995-2001. For each company and for each year seven financial ratios were collected with the Internet as the primary source. These ratios are suggested in Lehtinen's (1996) study of financial ratios' reliability and validity in international comparisons. The ratios measure four different aspects of companies' financial performance: profitability - 3 ratios (operating margin - OM, return on total assets - ROTA, and return on equity - ROE), liquidity - 1 ratio (current ratio = current assets / current liabilities), solvency - 2 ratios (equity to capital - EC, interest coverage - IC), and efficiency - 1 ratio (receivables turnover - RT) (Karlsson, 2002). In total the dataset consists of 651 rows taken from companies' financial statements in their annual reports: 88 companies * 7 years = 616 rows. 35 more rows were obtained with the averages for the five groups (5 groups * 7 years = 35 rows). Out of 651 rows 21 were discarded due to lack of data for calculating some ratios resulting in a final dataset of 630 rows.

In order to ease the training and to avoid the algorithms placing too much emphasis on extreme values, we removed far outliers and outliers from the data. An outlier is sometimes more technically defined as "a value whose distance from the nearest quartile is greater than 1.5 times the interquartile range" (SPSS for Windows, 2002). To remove the outliers we calculated the quartiles for each variable. If we denote with l the lower quartile, with m the median and with u the upper quartile of variable x, then the far outliers (fo), outliers (o) and anomalies (a) for that variable belong to the following intervals:

$$\begin{aligned}
&fo \in (-\infty, l-3d) \cup (u+3d, +\infty)\\
&o \in [l-3d, l-1.5d) \cup (u+1.5d, u+3d]\\
&a \in [l-1.5d, l-d) \cup (u+d, u+1.5d]
\end{aligned}$$

where $d = u$ - l is the distance from the upper quartile to the lower. For example, Figure 2 shows the frequencies of far outliers, outliers, and anomalies for Operating Margin (OM) ratio. There are 30 far outliers (green), 30 outliers (red), and 17 anomalies (blue).

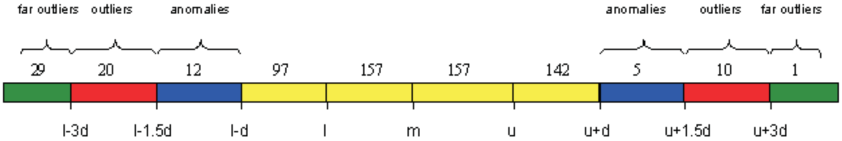


Figure 2: The structure of Operating Margin ratio

Once we have detected the far outliers and the outliers of each variable we have two alternatives: to discard a sample that has at least one far outlier or outlier value or to keep it by taking the peak(s) off. We chose the later alternative. For example, in the case of OM ratio, we "leveled" 49 left outliers values (29 far outliers + 20 outliers) with $l - 1.5d$ (= -22.48 for OM ratio) and 11 right outliers values (1 far outlier + 10 outliers) with $u + 1.5d$ (= 43.62 for OM ratio). We proceed likewise with all ratios.

In Table 3 we present the descriptive statistics including skewness, kurtosis and Kolmogorov-Smirnov normality test for the financial ratios of telecom companies. When the data is normally distributed the values for both skewness and kurtosis are 0. A positive value for skewness indicates that the distribution has more values less than the mean and a long right tail, while a negative value for skewness indicates that the distribution has more values greater than the mean and a long left tail. A negative value for kurtosis indicates flatness (flat center, thin tails), while a positive kurtosis indicates peakedness (spiky center, fat tails) (SPSS for Windows, 2002). A skewness or kurtosis value greater than ±2.0 indicates that the distribution differs significantly from a normal distribution (SPSS for Windows, 2002).

The skewness and kurtosis values fall into the range (-2, 2) of approximately normal distributions for all the variables. However, we encountered no 0 values. 5 financial ratios have positive

Table 3: Descriptive statistics for the financial ratios

Financial Ratio	unit	Min	Max	Mean	Std. Dev.	Skewness	Kurtosis	Kolmogorov-Smirnov Z	Sig.
Operating Margin	%	-22.48	43.62	8.8774	14.6053	-.363	.206	2.142	.000
Return on Total Assets	%	-15.48	32.67	8.6762	11.1545	.126	.116	1.435	.033
Return on Equity	%	-30.74	46.93	6.9287	20.2061	-.139	-.217	2.399	.000
Current Ratio	ratio	.11	4.23	1.7185	1.1004	1.006	.022	3.271	.000
Equity to Capital	%	-13.13	105.31	47.3223	26.5947	.311	.017	1.982	.001
Interest Coverage	ratio	-12.76	24.83	6.9007	9.8797	.474	-.251	3.650	.000
Receivables Turnover	ratio	.97	9.94	5.6167	2.0093	.502	-.045	2.115	.000

skewness which indicates slight asymmetric distribution with tails extending more towards positive values (there are more companies performing below the sector average). 4 financial ratios are somewhat “peaked” (positive values for kurtosis), operating margin being the most “peaked” and the other 3 financial ratios have flatter centers and thiner tails when compared to the normal distribution. Results of the Kolmogorov-Smirnov test show that the normality assumption is rejected for all financial ratios at a significance level of $\alpha = 0.05$. These results support the use of ANNs for financial analysis (e.g. financial classification models) over the traditional statistical methods since neural networks are free of any distributional assumptions.

We used the real dataset with “leveled” outliers and far-outliers to generate the fictive datasets.

Generating the fictive datasets. In order to test the impact of data distribution and preprocessing method on the predictive performances of the classifiers we generated in addition to the real dataset (“REAL”) four new datasets with different distributions: uniform (“UNIF”), normal(“NORM”), logistic(“LOG”) and Laplace (“LAP”) distributions. We chose these four distributions as they roughly correspond to four kurtosis values: -1, 0, 1, 3. We estimated the distributions’ parameters using the means and variances of the telecom dataset ratios. Regarding standardization three approaches were undertaken: one was to keep the data un-standardized (“no preprocessing” - PR1), the second was to normalize data to zero mean and unit standard deviation (“normalization” - PR2) and the third was to divide the data by the maximum of absolute values (“maximum of absolute values” PR3). We used these three preprocessing approaches to gradually cope with the comparability issues of input variables raised in section 4. In total we obtained 15 datasets, one for each distribution-preprocessing method combination: (REAL, PR1), (REAL, PR2), (REAL, PR3), (UNIF, PR1), ..., (LAP, PR3).

6 Experiments

For each one of the 15 datasets obtained we applied the following methodological steps:

1. For the RT-based ANN we repeated the procedure (described in subsection 3.3) 30 times, obtaining 4 vectors (30 elements in size) of different accuracy rates for each retraining mechanism type (RT1, RT2, RT3): a vector of effective training accuracy rates ($RT\_VEC\_ACR_{TRe}$), a vector of validation accuracy rates ($RT\_VEC\_ACR_{VAL}$), a vector of total training (effective training + validation) accuracy rate ($RT\_VEC\_ACR_{TR}$) and a vector of test accuracy rates ($RT\_VEC\_ACR_{TS}$). Correspondingly, we obtained 4 vectors with the mean square errors: $RT\_VEC\_MSE_{TRe}$, $RT\_VEC\_MSE_{VAL}$, $RT\_VEC\_MSE_{TR}$, and $RT\_VEC\_MSE_{TS}$. The total time needed for RT-based training was aproximatively 675 hours = 1.5 (hours/experiment) * 30 (experiments) * 15 (input datasets).

2. For the GA-based ANN we applied the procedure (described in subsection 3.4) 10 times

for each type of crossover (one-point - GAO, multi-point - GAM, arithmetic - GAA, and uniform - GAU). The other GA parameters used were as follows: $N_{gen} = 1000$, $PS = 20$, $N_{elite} = 3$, $max\_split = 5$, $P_c = 0.8$, $P_m = 0.01$ and $max\_lim = 1$. We obtained 2 vectors (10 elements in size) for each type of crossover operator: a vector of training accuracy rates ($GA\_VEC\_ACR_{TR}$) and a vector of test accuracy rates ($GA\_VEC\_ACR_{TS}$) and, correspondingly, 2 vectors with mean square errors: $GA\_VEC\_MSE_{TR}$, and $GA\_VEC\_MSE_{TS}$. The total time needed for GA-based training was aproximatively 1200 hours = 8 (hours/experiment) * 10 (experiments) * 15 (input datasets).

3. We used statistical tests to compare the vectors of the two training mechanisms in order to validate our hypotheses.

The following experiments differ in two perspectives: the hypothesis that they try to validate and/or the type of statistical test used (non-parametric vs. parametric).

Experiment 1. In the first experiment we try to validate the first hypothesis using non-parametric tests (Siegel and Castellan, 1988). We used the real dataset (the original telecom data) without preprocessing the data (first preprocessing approach). After we separated the data in training (90%) and test (10%) sets, we generated the ANN architecture. Then, in order to refine our solution, we applied the two training mechanisms (RT-based ANN and GA-based ANN). We applied the methodological steps described above and we compared statistically the results' vectors of both training mechanisms in order to validate our first hypothesis (Tables 4 and 5). We used Mann-Whitney-Wilcoxon and Kolmogorov-Smirnov non-parametric tests to avoid the assumptions of the parametric tests.

Table 4: Technique influence on training

	GAO RT1	GAO RT2	GAO RT3	GAM RT1	GAM RT2	GAM RT3	GAA RT1	GAA RT2	GAA RT3	GAU RT1	GAU RT2	GAU RT3
Mann-Whitney U	10.000	30.000	20.000	10.000	30.000	20.000	10.000	29.500	20.000	10.000	28.500	20.000
Wilcoxon W	475.000	495.000	485.000	475.000	495.000	485.000	475.000	494.500	485.000	475.000	493.500	485.000
Z	(5.628)	(4.555)	(5.069)	(5.582)	(4.521)	(5.029)	(5.573)	(4.534)	(5.022)	(5.581)	(4.578)	(5.029)
Asymp. Sig. (2-tailed)	.000	.000	.000	.000	.000	.000	.000	.000	.000	.000	.000	.000
Exact Sig. [2*(1-tailed Sig.)]	.000	.000	.000	.000	.000	.000	.000	.000	.000	.000	.000	.000
Kolmogorov-Smirnov Z	2.647	2.465	2.556	2.647	2.465	2.556	2.647	2.465	2.556	2.647	2.465	2.556
Asymp. Sig. (2-tailed)	.000	.000	.000	.000	.000	.000	.000	.000	.000	.000	.000	.000

Table 5: Technique influence on testing

	GAO RT1	GAO RT2	GAO RT3	GAM RT1	GAM RT2	GAM RT3	GAA RT1	GAA RT2	GAA RT3	GAU RT1	GAU RT2	GAU RT3
Mann-Whitney U	39.000	57.000	48.000	24.500	43.500	34.000	52.500	69.500	62.000	23.500	41.500	33.000
Wilcoxon W	504.000	522.000	513.000	489.500	508.500	499.000	517.500	534.500	527.000	488.500	506.500	498.000
Z	(4.777)	(3.716)	(4.219)	(5.205)	(4.139)	(4.648)	(4.369)	(3.313)	(3.766)	(5.231)	(4.207)	(4.676)
Asymp. Sig. (2-tailed)	.000	.000	.000	.000	.000	.000	.000	.001	.000	.000	.000	.000
Exact Sig. [2*(1-tailed Sig.)]	.000	.003	.001	.000	.000	.000	.001	.010	.005	.000	.000	.000
Kolmogorov-Smirnov Z	2.100	1.917	2.008	2.373	2.191	2.282	1.826	1.643	1.734	2.373	2.191	2.282
Asymp. Sig. (2-tailed)	.000	.001	.001	.000	.000	.000	.003	.009	.005	.000	.000	.000

As Table 5 shows (all significance coefficients = .000) all the pairs of accuracy rates vectors are statistically different. The direction of the difference is given by the statistics calculated. Mann-Whitney U statistic corresponds to the better group in the sense that it represents the smaller number of cases with higher ranks between groups. The Wilcoxon W statistic is simply the smaller of the two rank sums displayed for each group in the rank table. The Kolmogorov-Smirnov Z test statistic is a function of the combined sample size and the largest absolute difference between the two cumulative distribution functions of the two groups. Consequently, by analyzing both the calculated statistics and rank table we can determine the direction of the difference between the groups. For this particular experiment the rank table shows that the accuracy rates are always higher in the case of GA-based ANN training than for RT-based ANN, thus, validating first hypothesis.

As for training, GA-based ANN training models performed better than gradient-descent-like models in testing for all possible GA-RT technique-technique combinations.

Experiment 2. Our second experiment validates the second hypothesis using non-parametric tests. We preprocessed the real data using normalization and compared the results with those obtained for un-preprocessed data (Table 6). For each combination of the 2 preprocessing approaches and the 7 training techniques (4 GA-based ANN and 3 RT-based ANN) we calculated means for training and testing accuracy rates.

Table 6: Preprocessing method influence

	PR1-PR2 (TR)	PR1-PR2 (TS)
Mann-Whitney U	.000	6.000
Wilcoxon W	28.000	34.000
Z	(3.130)	(2.380)
Asymp. Sig. (2-tailed)	.002	.017
Exact Sig. [2*(1-tailed Sig.)]	.001	.017
Kolmogorov-Smirnov Z	1.871	1.604
Asymp. Sig. (2-tailed)	.002	.012

PR1 – "no preprocessing" PR2 – "normalization"

The preprocessing method had an impact on the both training mechanisms' performances. However, we found greater impact on the performance for training (U statistic = 0.000) than for testing (U = 6.000). Also, there is greater confidence on the results obtained for training (level of significance = 0.002) than for testing (level of significance = 0.02). Nevertheless, according to the rank tables we obtained higher accuracy rates when we preprocessed the data using normalization than the case when we used no preprocessing for both training and testing.

Experiment 3. To test our third hypothesis we applied the methodology on the fictive datasets and compare the results with those for the real data. In Table 7 we present the accuracy rates for training and testing samples. For this experiment we used no preprocessing of data. We calculated the means of accuracy rates vectors for each technique-distribution combination.

We applied the non-parametric tests to check the validity of our third hypothesis (Tables 8 and 9). The hypothesis is strongly supported both for training and testing cases. There is a statistical difference in performance between all distribution pairs, except three: real-logistic and uniform-normal pairs in the case of training and logistic-Laplace pair in the case of testing. The performance order of the distributions fit our expectations; the best accuracy rates were obtained for normally distributed data, followed by data distributed uniformly. The third best performances were achieved for the real dataset which overcame logistic and Laplace distributions in this order.

Experiment 4. Here we use non-parametric tests to validate hypotheses $H4$ that crossover operator has an influence on both GA-based ANN training and testing performances (Table 10) and $H5$ that the generation of effective training and validation sets has an influence on both RT-based ANN training and testing performances. As for the first experiment we used the real dataset

Table 7: Accuracy rates for distribution pairs' comparison (no preprocessing)

	REAL TR	UNIF TR	NORM TR	LOG TR	LAP TR	REAL TS	UNIF TS	NORM TS	LOG TS	LAP TS
GAO	93.02	95.84	96.39	94.82	90.09	85.24	88.57	89.46	81.19	80.41
GAM	92.86	95.84	96.60	94.99	90.00	85.48	87.86	89.64	81.43	81.02
GAA	92.92	95.78	96.49	94.80	90.16	85.48	88.93	90.00	81.43	81.22
GAU	93.30	95.76	96.43	94.82	90.19	85.95	88.75	89.82	81.19	81.02
RT1	92.22	94.92	95.48	92.70	88.12	83.49	89.11	89.46	79.92	78.10
RT2	92.43	95.04	95.45	92.70	88.06	83.97	88.57	89.52	79.92	77.76
RT3	92.41	94.84	95.52	92.53	88.06	83.81	89.05	89.52	79.29	77.89

Table 8: Distribution influence on training

	REAL UNIF	REAL NORM	REAL LOG	REAL LAP	UNIF NORM	UNIF LOG	UNIF LAP	NORM LOG	NORM LAP	LOG LAP
Mann-Whitney U	.000	.000	12.000	.000	12.000	2.000	.000	.000	.000	.000
Wilcoxon W	28.000	28.000	40.000	28.000	40.000	30.000	28.000	28.000	28.000	28.000
Z	(3.134)	(3.130)	(1.599)	(3.134)	(1.599)	(2.881)	(3.137)	(3.134)	(3.134)	(3.137)
Asymp. Sig. (2-tailed)	.002	.002	.110	.002	.110	.004	.002	.002	.002	.002
Exact Sig. [2*(1-tailed Sig.)]	.001	.001	.128	.001	.128	.002	.001	.001	.001	.001
Kolmogorov-Smirnov Z	1.871	1.871	1.069	1.871	1.069	1.604	1.871	1.871	1.871	1.871
Asymp. Sig. (2-tailed)	.002	.002	.203	.002	.203	.012	.002	.002	.002	.002

Table 9: Distribution influence on testing

	REAL UNIF	REAL NORM	REAL LOG	REAL LAP	UNIF NORM	UNIF LOG	UNIF LAP	NORM LOG	NORM LAP	LOG LAP
Mann-Whitney U	.000	.000	.000	.000	.000	.000	.000	.000	.000	14.000
Wilcoxon W	28.000	28.000	28.000	28.000	28.000	28.000	28.000	28.000	28.000	42.000
Z	(3.134)	(3.137)	(3.144)	(3.134)	(3.134)	(3.141)	(3.130)	(3.144)	(3.134)	(1.346)
Asymp. Sig. (2-tailed)	.002	.002	.002	.002	.002	.002	.002	.002	.002	.178
Exact Sig. [2*(1-tailed Sig.)]	.001	.001	.001	.001	.001	.001	.001	.001	.001	.209
Kolmogorov-Smirnov Z	1.871	1.871	1.871	1.871	1.871	1.871	1.871	1.871	1.871	.802
Asymp. Sig. (2-tailed)	.002	.002	.002	.002	.002	.002	.002	.002	.002	.541

(the original telecom data) without preprocessing the data (first preprocessing approach).

Table 10: The influence of crossover operator on training and testing

	GAO GAM TR	GAO GAA TR	GAO GAU TR	GAM GAA TR	GAM GAU TR	GAA GAU TR	GAO GAM TS	GAO GAA TS	GAO GAU TS	GAM GAA TS	GAM GAU TS	GAA GAU TS
Mann-Whitney U	35.000	40.000	45.000	45.000	32.000	36.500	45.000	47.000	41.000	49.000	45.500	44.500
Wilcoxon W	90.000	95.000	100.000	100.000	87.000	91.500	100.000	102.000	96.000	104.000	100.500	99.500
Z	(1.826)	(1.082)	(.608)	(.445)	(1.679)	(1.201)	(.610)	(.269)	(.976)	(.094)	(.548)	(.491)
Asymp. Sig. (2-tailed)	**.068**	.279	.543	.656	**.093**	.230	.542	.788	.329	.925	.584	.624
Exact Sig. [2*(1-tailed Sig.)]	.280	.481	.739	.739	.190	.315	.739	.853	.529	.971	.739	.684
Kolmogorov-Smirnov Z	.671	.671	.447	.224	.447	.447	.224	.447	.224	.447	.224	.447
Asymp. Sig. (2-tailed)	.759	.759	.998	1.000	.988	.988	1.000	.988	1.000	.988	1.000	.988

We found a very weak support: two pair-vectors differ significantly at a level of significance of 0.1: GAO vs. GAM and GAM vs. GAU, both in the case of training phase. Also, we found no evidence to differentiate between the three retraining mechanisms.

Experiment 5. In the first 4 experiments, when we validate our hypotheses, we relied exclusively on non-parametric tests. We argued that the parametric tests (like t-test, univariate $ANOVA$ etc.) require the analyzed vectors to satisfy different assumptions. For instance when applying $ANOVA$ analysis one should check the following assumptions: observations are independent, the sample data have a normal distribution, and scores in different groups have homogeneous variances. The first assumption is satisfied since all other factors besides preprocessing, distribution and training mechanism that could influence the classifiers' performances are fixed. For the second assumption we argue that $ANOVA$ is robust against normality assumptions if the sample size is large. Regarding the third assumption, $SPSS$ (the software that we used) incorporates the case when the variances between groups are assumed to be non-equal.

In order to give more strength to our results from the previous experiments and, at the same time, to validate our main hypothesis, we finally performed a 3-way $ANOVA$ analysis having as grouping variables: the technique used (GAO, GAM, GAA, GAU, RT_1, RT_2, and RT_3), the preprocessing method (PR_1 - "no preprocessing", PR_2 - "normalization", PR_3 - "dividing the variables by the maximum absolute values"), the data distribution ($REAL$, $UNIF$, $NORM$, LOG, and LAP). With the third preprocessing method we obtained values between -1 and +1. We used the vectors' means to fill in our accuracy rates data. Tables 11 and 12 include the data we used to perform 3-way $ANOVA$.

Next, the results of 3-way $ANOVA$ for both training and test accuracy rates are shown in Tables 13 and 14.

As the tables show all the factors are statistically significant. In other words they have an individual and combined influence on both training and testing performances. The last column (*partial eta squared*) reports the "practical" significance of each term, based upon the ratio of the variation (sum of squares) accounted for by the term, to the sum of the variation accounted for by the term and the variation left to error. Larger values of *partial eta squared* indicate a greater amount of variation accounted for by the model term, to a maximum of 1. Here the individual factors and their combinations, while statistically significant, have great effect on classifier accuracy. Consequently, the main hypothesis (**H6**) is validated.

In the next 3 tables we present the pairs' comparison for the training performances. The second hypothesis (*H2*) is validated (Table 15) and "normalization" is the best preprocessing approach, followed by "maximum absolute values" and "no preprocessing" in this order. Concerning the third hypothesis (*H3*) the best performance was obtained when data were *normally* distributed (Table 16). The next best distribution was that of the real data, followed by uniform, logistic and Laplace. Our first hypothesis (*H1*) is satisfied (Table 17), GA performing better than RT in

Table 11: Accuracy rates for training

PREPROC	DISTRIB	TECHNIQUE						
		GA				RT		
		GAO	GAM	GAA	GAU	RT1	RT2	RT3
Un-preprocessed	REAL	93.02	92.86	92.92	93.30	92.22	92.43	92.41
	UNIF	95.84	95.84	95.78	95.76	94.92	95.04	94.84
	NORM	96.39	96.60	96.49	96.43	95.48	95.45	95.52
	LOG	94.82	94.99	94.80	94.82	92.70	92.70	92.53
	LAP	90.09	90.00	90.16	90.19	88.12	88.06	88.06
Normalization	REAL	99.43	99.49	99.46	99.33	99.11	99.10	99.08
	UNIF	99.79	99.81	99.79	99.79	99.80	99.80	99.80
	NORM	97.90	97.90	97.90	97.90	98.07	98.03	97.97
	LOG	99.11	99.06	98.98	98.98	98.95	98.98	98.96
	LAP	98.08	98.13	98.01	98.10	98.02	98.01	98.06
Max of Absolute Values	REAL	99.68	99.68	99.68	99.68	99.69	99.69	99.69
	UNIF	97.79	97.77	97.73	97.84	97.77	97.77	97.89
	NORM	96.91	97.00	97.02	97.02	96.93	96.90	96.91
	LOG	96.50	96.52	96.52	96.52	96.68	96.59	96.60
	LAP	95.64	95.83	95.76	95.81	95.26	95.47	95.23

Table 12: Accuracy rates for testing

PREPROC	DISTRIB	TECHNIQUE						
		GA				RT		
		GAO	GAM	GAA	GAU	RT1	RT2	RT3
Un-preprocessed	REAL	85.24	85.48	85.48	85.95	83.49	83.97	93.81
	UNIF	88.57	87.86	88.93	88.75	89.11	88.57	89.05
	NORM	89.46	89.64	90.00	89.82	89.46	89.52	89.52
	LOG	81.19	81.43	81.43	81.19	79.92	79.92	79.29
	LAP	80.41	81.02	81.22	81.02	78.10	77.76	77.89
Normalization	REAL	85.71	86.19	85.71	85.71	85.79	85.63	85.79
	UNIF	92.86	93.04	92.86	92.86	92.86	92.86	92.92
	NORM	96.43	96.43	96.43	96.43	96.49	96.19	96.43
	LOG	88.10	88.10	88.10	88.10	88.10	88.25	88.25
	LAP	92.25	92.45	91.84	92.25	91.36	91.50	91.56
Max of Absolute Values	REAL	97.62	97.62	97.62	97.62	97.54	97.70	97.62
	UNIF	95.00	95.36	95.71	95.36	96.31	96.43	96.07
	NORM	93.21	93.57	93.93	93.57	92.86	93.15	93.27
	LOG	88.10	88.10	87.86	88.10	88.25	88.10	88.25
	LAP	88.37	89.18	88.98	89.59	89.86	89.93	89.86

Table 13: 3-way ANOVA for training

Source	Type III Sum of Squares	df	Mean Square	F	Sig.	Partial Eta Squared
Model	979340.512	30	32644.684	6322228.263	.000	1.000
PREPROC	540.706	2	270.353	52358.680	.000	.999
DISTRIB	148.280	4	37.070	7179.276	.000	.997
TECHNIQ	6.396	1	6.396	1238.708	.000	.943
PREPROC * DISTRIB	138.900	8	17.362	3362.559	.000	.997
PREPROC * TECHNIQ	9.486	2	4.743	918.554	.000	.961
DISTRIB * TECHNIQ	1.426	4	.356	69.036	.000	.786
PREPROC * DISTRIB * TECHNIQ	2.574	8	.322	62.310	.000	.869
Error	.387	75	.005			
Total	979340.899	105				

Table 14: 3-way ANOVA for testing

Source	Type III Sum of Squares	df	Mean Square	F	Sig.	Partial Eta Squared
Model	844172.942	30	28139.09	506683.910	.000	1.000
PREPROC	1296.821	2	648.411	11675.541	.000	.997
DISTRIB	904.441	4	226.110	4071.431	.000	.995
TECHNIQ	3.537	1	3.537	63.695	.000	.459
PREPROC * DISTRIB	605.016	8	75.627	1361.771	.000	.993
PREPROC * TECHNIQ	10.714	2	5.357	96.461	.000	.720
DISTRIB * TECHNIQ	5.432	4	1.358	24.454	.000	.566
PREPROC * DISTRIB * TECHNIQ	10.002	8	1.250	22.511	.000	.706
Error	4.165	75	.056			
Total	844177.107	105				

refining the solution. However, the difference between accuracy rates is not as obvious as it was for the "real" data from experiment 1. This is explainable since in later case (only "real" data) the starting solution has relatively low accuracy rates (80-90%) and it could have been easily improved while in this experiment (centralized data) we have some starting solutions with high accuracy rates (95-98%) that would be hard to improve whatever would be the training mechanism used to *refine* them. We find no evidence for the fourth and fifth hypotheses (*H4, H5*), all crossover operators and retrainig mechanisms achieving comparable results.

Table 15: Pairs' comparison for "preprocessing" factor

(I) PREPROC	(J) PREPROC	Mean Difference (I-J)	Std. Error	Sig.	95% Confidence Interval for Difference	
					Lower Bound	Upper Bound
1	2	(5.438)(*)	.017	.000	(5.473)	(5.404)
	3	(3.933)(*)	.017	.000	(3.967)	(3.898)
2	1	5.438(*)	.017	.000	5.404	5.473
	3	1.506(*)	.017	.000	1.471	1.540
3	1	3.933(*)	.017	.000	3.898	3.967
	2	(1.506)(*)	.017	.000	(1.540)	(1.471)

1-"no preprocessing", 2-"normalization", 3-"maximum of absolute values"
(*) The mean difference is significant at the .05 level.

In the case of pairs' comparisons for testing performances we encounter a similar result. All the mean differences are statistically significant. Only the order of best performers has slightly changed: "maximum of absolute values" - "normalization" - "no preprocessing" for the "preprocessing" factor, uniform - normal - real - Laplace - logistic for the "distribution" factor. Once again, our first hypothesis is satisfied, GA performing better on test data as well.

7 Conclusions

In this study, we investigate the influence of three different factors and their combinations on the prediction performance of ANN classification models. The three factors are: pre-processing method (none, division by absolute maximum values and normalization), data distribution (the real data, uniform, normal, logistic and Laplace distributions) and training mechanism (a gradient-descent-like mechanism improved by a retraining procedure - RT - and a natural-evolution-based mechanism known as genetic algorithm - GA).

Few studies have shown the *individual* influence of preprocessing method and data distribution on the prediction performance of ANNs. Koskivaara (2000) investigates the impact of four pre-processing techniques on the forecast capability of ANNs. Other studies (Pendharkar, 2002; Pendharkar and Rodger, 2004) investigate the combined influence of other factors such as distribution kurtosis, variance heterogeneity, network size and input and weights noise on the ANN classification performance. After examining Alander's paper (Alander, 1995) we could not find any report in literature which analysis the influence of data distribution, preprocessing method, training mechanism and their combinations on the classification performance of ANNs. In this study we are concerned with questions regarding the choice of different factor-factor pairs when the third factor is fixed. For example which combination preprocessing method-training mechanism would be the most suitable given the distribution of the data is known.

As we have shown (section 4), this study has a different perspective than other studies which use genetic algorithms to train neural networks. A major difference with related studies (Schaffer *et al.*, 1992; Sexton and Gupta, 2000; Sexton and Sikander, 2001; Pendharkar, 2002; Pendharkar and Rodger, 2004) is that the two ANN training mechanisms are used to *refine* the initial solution (the

Table 16: Pairs' comparison for "distribution" factor

(I) DISTRIB	(J) DISTRIB	Mean Difference (I-J)	Std. Error	Sig.	95% Confidence Interval for Difference	
					Lower Bound	Upper Bound
1	2	(.439)(*)	.022	.000	(.483)	(.394)
	3	.251(*)	.022	.000	.206	.296
	4	.775(*)	.022	.000	.730	.819
	5	2.984(*)	.022	.000	2.939	3.3028
2	1	.439(*)	.022	.000	.394	.483
	3	.690(*)	.022	.000	.645	.734
	4	1.213(*)	.022	.000	1.169	1.258
	5	3.422(*)	.022	.000	3.378	3.467
3	1	(.251)(*)	.022	.000	(.296)	(.206)
	2	(.690)(*)	.022	.000	(.734)	(.645)
	4	.524(*)	.022	.000	.479	.568
	5	2.733(*)	.022	.000	2.688	2.777
4	1	(.775)(*)	.022	.000	(.819)	(.730)
	2	(1.213)(*)	.022	.000	(1.258)	(1.169)
	3	(.524)(*)	.022	.000	(.568)	(.479)
	5	2.209(*)	.022	.000	2.164	2.254
5	1	(2.984)(*)	.022	.000	(3.028)	(2.939)
	2	(3.422)(*)	.022	.000	(3.467)	(3.378)
	3	(2.733)(*)	.022	.000	(2.777)	(2.688)
	4	(2.209)(*)	.022	.000	(2.254)	(2.164)

1-REAL, 2-NORM, 3-UNIF, 4-LOG, 5-LAP
(*) The mean difference is significant at the .05 level.

Table 17: Pairs' comparison for "technique" factor

(I) TECHNIQ	(J) TECHNIQ	Mean Difference (I-J)	Std. Error	Sig.	95% Confidence Interval for Difference	
					Lower Bound	Upper Bound
1	2	.499(*)	.014	.000	.471	.527
2	1	(.499)(*)	.014	.000	(.527)	(.471)

1-GA, 2-RT
(*) The mean difference is significant at the .05 level.

ANN set of weights). Rather than randomly generating it, the initial solution is obtained when determining the ANN architecture which is kept fixed in the refining process for both training mechanisms. An empirical procedure to determine the proper ANN architecture is introduced. Problem complexity (the number of variables and output neurons) is another difference with related studies which usually consider the two-class discrimination problem. In our prediction models the number of financial performance classes is set to 7. We can easily change this parameter to simulate the binary classification problem allowing us precise and detailed comparisons with other related studies. Another distinction with related studies comes from the type of the tests used to validate hypotheses. In this study, we rely on non-parametric tests to validate individual influence of the factors. However, we finally performed a 3-way ANOVA to validate the main hypothesis, but without violating its constraints.

Depending on where the training and validation sets are generated we have three RT-based training mechanisms and depending on the crossover operator used we have four GA-based training mechanisms. RT-based training mechanism is a new way of training an ANN based on its past training experience and weights reduction (*Own ref.*, 2004). In addition to what was reported in literature (e.g. Pendharkar, 2004) we introduce a new crossover operator (multi-point crossover) and test its performance against classical one-point, aritmetic and uniform crossovers.

We define six hypotheses. Hypotheses $H1$, $H2$, and $H3$ are concerned with the individual influence of each of the three factors on the prediction performance of the ANN. The results show a very strong support for all three hypotheses. Concerning $H1$ we found that when the starting solution has relatively low accuracy rates (80-90%) GA outperformed the RT mechanism, while the difference was smaller to zero when the starting solution had relatively high accuracy rates (95-98%). This can be considered a normal result since we do not expect great improvements starting from an already very good solution. It is interesting to check in the future studies whether these hybrid approaches overcome the classical ones (the ones were the weights of the ANN are randomly initialized). The validation of $H2$ show that preprocessing method has an influence on the ANN performance, normalization achieving the best results. In line with Pendharkar (2002), the validation of $H3$ shows that increasing kurtotic data distributions hurt the performance of ANN during both training and testing phases.

Hypothesis $H4$ tests the influence of crossover operator on the prediction performance of GA-based ANN. As it was reported (Yao, 1999; Pendharkar and Rodger, 2004) the crossover operator seems to have no impact on the classification performance of GA-based ANNs. The 5^{th} hypothesis ($H5$) tests whether the point at which we split the data into effective training and validation sets has any impact on the prediction performance of RT-based ANN. We found no difference in RT-based ANN training performance for all three RT-based mechanisms.

The main hypothesis (**H6**) concerns the individual and combined influence of the three factors on the prediction performance of ANNs. In experiment 5 we tested $H6$ performing a 3-way ANOVA, and again, all individual factors have a statistically significant influence on both ANN training and testing performances. At the same time, the influence of any combination of the three factors was found to be statistically significant. The results of pairs' comparisons for each factor validate once again the first three hypotheses.

In our experiments RT was much faster than GA. Therefore, when the time is a critical factor, RT can be taken into consideration as long as there is no major difference between the performances of these two approaches. The GA-based ANN needs around 1000 generations for each training. Other stopping criteria may be employed to make the GA training faster. Further research may be focused on tunning GA parameters making the GA training more efficient.

Acknowledgements

We wish to thank Jonas Karlsson for providing us the telecom dataset and Barbro Back for her valuable comments. We would like to thank two anonymous reviewers for very useful comments and criticisms.

Appendix A

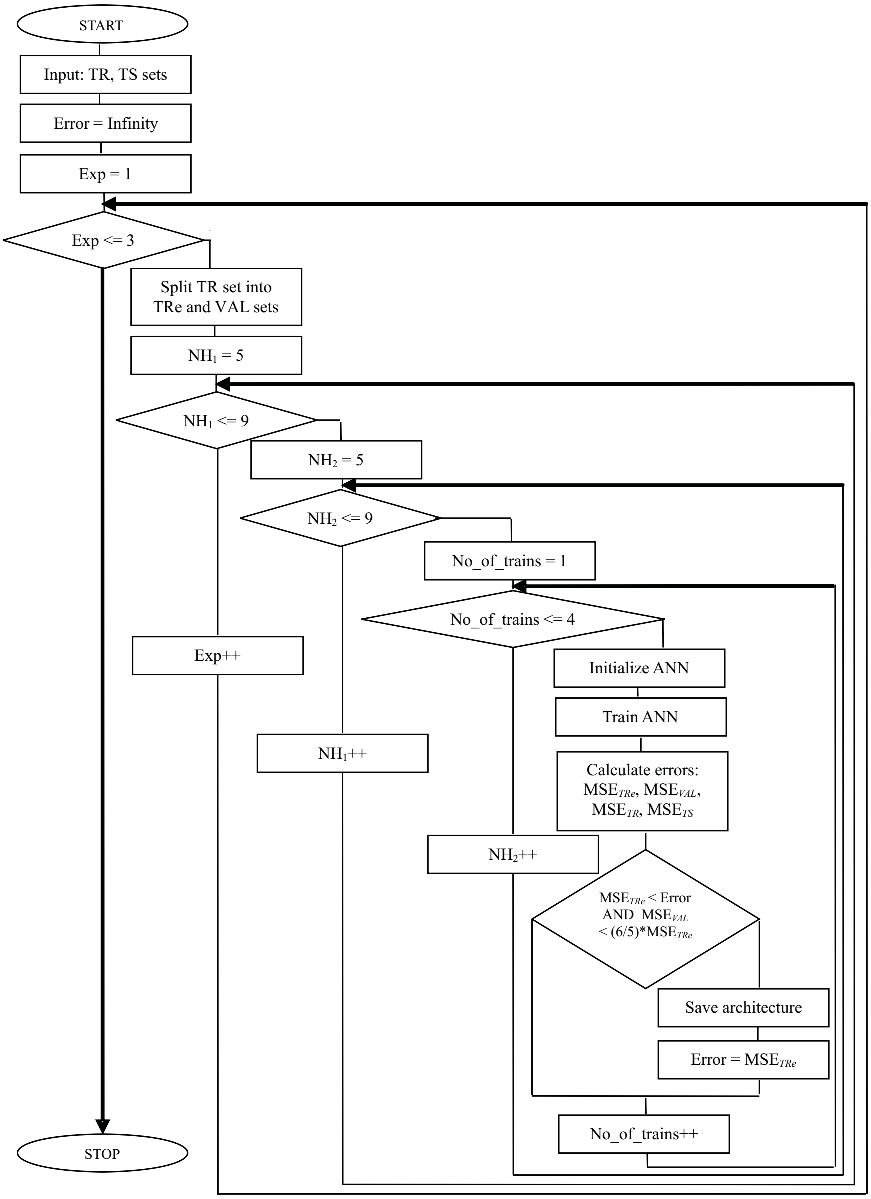


Figure 3: Flowchart of the empirical procedure to determine ANN architecture

References

[1] Alander JT. 1995. Indexed bibliography of genetic algorithms and neural networks. Report 94-1-NN, University of Vaasa, Department of Information Technology and Production Economics, 1995. (ftp://ftp.uwasa.fi/cs/report94-1/gaNNbib.ps.Z) *Key:* gaNNbib.

[2] Altman EI. 1968. Financial Ratios, Discriminant Analysis and the Prediction of Corporate Bankruptcy. In *The Journal of Finance* **23**: 589-609.

[3] Anderson E. 1935. The irises of the Gaspe' peninsula. In *Bull Am Iris Soc* **59**: 2-5.

[4] Ankenbrandt CA. 1991. An extension to the theory of convergence and a proof of the time complexity of genetic algorithms. In *Proceedings of 4th International Conference on Genetic Algorithm*; 53-68.

[5] Back B, Laitinen T, Sere K, van Wezel, M. 1996a. Choosing Bankruptcy Predictors Using Discriminant Analysis, Logit Analysis, and Genetic Algorithms. In *TUCS Technical Report* **40**, September 1996.

[6] Back B, Sere K, Vanharanta H. 1996b. Data Mining Accounting Numbers Using Self Organising Maps. In Proceedings of Finnish Artificial Intelligence Conference, Vaasa, Finland, 20-23 August 1996.

[7] Back B, Laitinen T, Hekanaho J, Sere K. 1997. The Effect of Sample Size on Different Failure Prediction Methods. In *TUCS Technical Report* **155**, December 1997.

[8] Back B, Sere K, Vanharanta H. 1998. Managing Complexity in Large Data Bases Using Self-Organizing Maps. In *Accounting Management and Information Technologies* **8**(4):191-210.

[9] Basheer IA, Hajmeer M. 2000. Artificial neural networks: fundamentals, computing, design, and application. In *Journal of Microbiological Methods* **43**: 3-31.

[10] Bhattacharyya S, Pendharkar PC. 1998. Inductive, evolutionary and neural techniques for discrimination: An empirical study. In *Decision Sciences* **29**(4): 871-899.

[11] Beaver WH. 1966. Financial Ratios as Predictors of Failure, Empirical Research in Accounting: Selected Studies. In *Supplement to Journal of Accounting Research* **4**: 71-111.

[12] Bezdek JC. 1981. *Pattern Recognition with Fuzzy Objective Function Algorithms.* Plenum Press, New York.

[13] Breiman L, Friedman JH, Olshen R, Stone C. 1984. *Classification and Regression Trees.* Wadsworth and Brooks, Monterey, CA.

[14] Chellapilla K, Fogel DB. 1999. Evolution, Neural Networks, Games, and Intelligence. In *Proceedings of the IEEE*, September: 1471-1496.

[15] Chester DL. 1990. Why Two Hidden Layers are Better than One. In Proceedings of International Joint Conference on Neural Networks IJCNN-1990 **1**, Lawrence Erlbaum, pp. 265-268.

[16] Coakley JR, Brown CE. 2000. Artificial Neural Networks in Accounting and Finance: Modeling Issues. In *International Journal of Intelligent Systems in Accounting, Finance & Management* **9**: 119-144.

[17] Davis L (ed.). 1991. *Handbook of Genetic Algorithms.* Van Nostrand Reinhold: New York.

[18] Deboeck G. 1998. Financial Applications of Self-Organizing Maps. In *Neural Network World* **8**(2): 213-241.

[19] DeJong KA. 1975. *An Analysis of the Behavior of a Class of Genetic Adaptive Systems.* Ph.D. dissertation, University of Michigan, Ann Arbor, MI.

[20] Demuth H, Beale M. 2001. *Neural Network Toolbox.* The MathWorks Inc, Natick Press: MA, USA.

[21] Dorsey RE, Mayer WJ. 1995. Genetic Algorithms for Estimation Problems with Multiple Optima, Non-differentiability, and other Irregular Features. In *Journal of Business and Economic Statistics* **13**(1): 53-66.

[22] Edmister RO. 1972. An Empirical Test Of Financial Ratio Analysis For Small Business Failure Prediction. In *Journal of Financial and Quantitative Analysis* **7**: 1477-1493.

[23] Eklund T, Back B, Vanharanta H, Visa A. 2003. Financial Benchmarking Using Self-Organizing Maps - Studying the International Pulp and Paper Industry. In *Data Mining - Opportunities and Challenges.* J. Wang, Ed. Hershey, PA, Idea Group Publishing: 323-349.

[24] Fisher RA. 1936. The use of multiple measurements in taxonomic problems. In *Ann. Eugenics* **7**: 179-188.

[25] Fogel DB, Wasson EC, Boughton EM. 1995. Evolving Neural Networks for Detecting Breast Cancer. In *Cancer Letters* **96**: 49-53.

[26] Fogel DB, Wasson EC, Boughton EM, Porto VW. 1998. Evolving artificial neural networks for screening features from mammograms. In *Artificial Intelligence in Medicine* **14**: 317-326.

[27] Fogel GB, Weekes DG, Sampath R, Ecker DJ. 2004. Parameter Optimization of an Evolutionary Algorithm for RNA Structure Discovery. In *Proceedings of 2004 Congress on Evolutionary Computation*, IEEE Press, Piscataway, NJ, pp. 607-613.

[28] Frydman H, Altman EI, Kao DL. 1985. Introducing Recursive Partitioning for Financial Classification: The Case of Financial Distress. In *The Journal of Finance* **40**: 269-291.

[29] Hagan MT, Demuth HB, Beale M. 1996. *Neural Networks Design.* MA: PWS Publishing, Boston.

[30] Hamer M. 1983. Failure Prediction: Sensitivity of classification accuracy to alternative statistical method and variable sets. In *Journal of Accounting and Public Policy* **2**(Winter): 289-307.

[31] Hancock P. 1992. *Coding strategies for genetic algorithms and neural nets.* PhD Thesis, University of Stirling, Department of Computer Science, 1992.

[32] Hartman E, Keeler JD. 1991. Predicting the Future: Advantages of Semilocal Units. In *Neural Computation* **3**(4): 566-578.

[33] Hesser J, Männer R. 1990. Towards an Optimal Mutation Probability for Genetic Algorithms. In *Proceedings of the* 1st *Workshop on Parallel Problem Solving from Nature*, Sringer-Verlag, October 01-03, pp.23-32.

[34] Hoehn PT. 1998. Wolves in Sheeps Clothing? The Effects of "Hidden" Parental Mutation on Genetic Algorithm Performances. In *Proceedings of ACM 36th annual Southeast regional conference*, pp. 221-227.

[35] Jeng B, Jeng YM, Liang TP. 1997. FILM: A Fuzzy Inductive Learning Method for Automatic Knowledge Acquisition. In *Decision Support Systems* **21**(2): 61-73.

[36] Jones F. 1987. Current techniques in bankruptcy prediction. In *Journal of Accounting Literature* **6**: 131-164.

[37] Karlsson J. 2002. *Data-Mining, Benchmarking and Analysing Telecommunications Companies.* Licentiate Thesis at the Department of Information Systems at Åbo Akademi University, Turku.

[38] Klir GJ, Yuan B. (1995). *Fuzzy Sets and Fuzzy Logic. Theory and Applications.* Prentice Hall PTR, Upper Saddle River, New Jersey.

[39] Kohonen T. 1997. *Self-Organizing Maps.* 2nd edition, Springer-Verlag, Heidelberg.

[40] Koskivaara E. 2004. Artificial Neural Networks in Analytical Review Procedures. In *Managerial Auditing Journal* **19**(2): 191-223.

[41] Koskivaara E. 2000. Different Pre-Processing Models for Financial Accounts when Using Neural Networks for Auditing. In *Proceedings of the European Conference on Information Systems*, Hans Robert Hansen - Martin Bichler - Harald Mahrer (eds.), Viena, Austria, July 3-5, 2000, pp. 326-332.

[42] Lachtermacher G, Fuller JD. 1995. Backpropagation in time series forecasting. In *Journal of Forecasting* **14**: 381393.

[43] Lehtinen J. 1996. *Financial Ratios in an International Comparison.* Acta Wasaensia. 49, Vasa.

[44] Lönnblad L, Peterson C, Rögnvaldsson T. 1992. Mass Reconstruction with a Neural Network. In *Physics Letters* **B278**: 181-186.

[45] Marais ML, Patell JM, Wolfson MA. 1984. The experimental design of classification models: An application of recursive partitioning and bootstrapping to commercial bank loan classification. In *Journal of Accounting Research* **22**: 87-114.

[46] Martín-del-Brío B, Serrano Cinca C. 1993. Self Organizing Neural Networks for the Analysis and Representation of Data: some Financial Cases. In *Neural Computing & Applications.* Springer Verlag (ed.) **1**(2): 193-206.

[47] Masters T. 1994. *Practical Neural Network Recipes in C++.* Academic Press, Boston, MA.

[48] Michalewicz Z. 1992. *Genetic Algorithms + Data Structures = Evolution Programs.* Springer: Berlin.

[49] Miller JF, Thomson P. 1998. Aspects of Digital Evolution: Geometry and Learning. In *Proceedings of the* 2^{nd} *International Conference on Evolvable Systems - ICES98*, 23-25 September, 1998, EPFL, Lausanne, Switzerland.

[50] Moller MF. 1993. A scaled conjugate gradient algorithm for fast supervised learning. In *Neural Networks* **6**: 525-533.

[51] Ohlsson M, Peterson C, Pi H, Rögnvaldsson T, Söderberg B. 1994. Predicting Utility Loads with Artificial Neural Networks – Methods and Results from the Great Energy Predictor Shootout. In *1994 Annual Proceedings of ASHRAE*, Inc, ASHRAE Transactions: Symposia.

[52] O'Leary DE. 1998. Using Neural Networks to Predict Corporate Failure. In *International Journal of Intelligent Systems in Accounting, Finance & Management* **7**: 187-197.

[53] Pendharkar PC. 2002. A computational study on the performance of artificial neural networks under changing structural design and data distribution. In *European Journal of Operational Research* **138**: 155-177.

[54] Pendharkar PC, Rodger JA. 2004. An empirical study of impact of crossover operators on the performance of non-binary genetic algorithm based neural approaches for classification. In *Computers & Operations Research* **31**: 481-498.

[55] Quinlan JR. 1993. A Case Study in Machine Learning. In *Proceedings of ACSC-16 Sixteenth Australian Computer Science Conference*, January, Brisbane; 731-737.

[56] Quinlan JR. 1993. *C4.5 Programs for Machine Learning.* Morgan Kaufmann Series in Machine Learning. Morgan Kaufmann Publishers, San Mateo.

[57] Rogero JM. 2002. A Genetic Algorithms Based Optimisation Tool for the Preliminary Design of Gas Turbine Combustors. *PhD Thesis*, Cranfield University, November 2002.

[58] Rudolfer S, Paliouras G, Peers I. 1999. A Comparison of Logistic Regression to Decision Tree Induction in the Diagnosis of Carpal Tunnel Syndrome. In *Computers and Biomedical Research* **32**: 391-414.

[59] Rudolph G. 1994. Convergence analysis of canonical genetic algorithms. In *IEEE Transactions on Neural Networks* **5**: 96-101.

[60] Schaffer JD, Whitley D, Eshelman LJ. 1992. Combinations of Genetic Algorithms and Neural Networks: A survey of the state of the art. In *COGANN-92 Combinations of Genetic Algorithms and Neural Networks.* IEEE Computer Society Press: Los Alamitos, CA; 1-37.

[61] Schaffer JD. 1994. Combinations of genetic algorithms with neural networks or fuzzy systems. In *Computational Intelligence: Imitating Life*, Zurada JM, Marks RJ, Robinson CJ (eds). IEEE Press: New York; 371-382.

[62] Schütze H, Hull D, Pedersen J. 1995. A comparison of classifiers and document representations for the routing problem. In *Proceedings of the 18th annual international ACM SIGIR conference on Research and development in information retrieval*, Seattle, Washington, United States. ACM Press: New York, NY, USA; 229-237.

[63] Serrano Cinca C. 1996. Self Organizing Neural Networks for Financial Diagnosis. In *Decision Support Systems*. Elsevier Science **17**: 227-238.

[64] Serrano Cinca C. 1998a. Self-organizing Maps for Initial Data Analysis: Let Financial Data Speak for Themselves. In *Visual Intelligence in Finance using Self-organizing Maps*. Guido Deboeck and Teuvo Kohonen (eds.). Springer Verlag, July 1998.

[65] Serrano Cinca C. 1998b. From Financial Information to Strategic Groups - a Self Organizing Neural Network Approach. In *Journal of Forecasting*. John Wiley and Sons (ed.), September 1998, **17**: 415-428.

[66] Sexton RS, Dorsey Re, Johnson JD. 1998. Toward a global optimum for neural networks: A comparison of the genetic algorithm and backpropagation. In *Decision Support Systems* **22**(2): 171-186.

[67] Sexton RS, Gupta JND. 2000. Comparative evaluation of genetic algorithm and backpropagation for training neural networks. In *Information Sciences* **129**: 45-49.

[68] Sexton RS, Sikander NA. 2001. Data Mining Using a Genetic Algorithm-Trained Neural Network. In *International Journal of Intelligent Systems in Accounting, Finance & Management* **10**: 201-210.

[69] Shimodaira H. 1996. A New Genetic Algorithm Using Large Mutation Rates and Population-Elitist Selection (GALME). In *Proceedings of the 8th International Conference on Tools with Artificial Intelligence (ICTAI '96)*, pp. 25-32.

[70] Siegel S, Castellan-Jr. NJ. 1988. *Nonparametric Statistics for the Behavioral Sciences*, 2nd edition. McGraw-Hill International Editions.

[71] SPSS for Windows. Release 11.5.1 (16 Nov 2002). Chicago: SPSS Inc.

[72] tuson) Tuson A, Ross P. 1998. Adapting Operator Settings in Genetic Algorithms. In *Evolutionary Computation* **6**(2): 161-184.

[73] Upadhyaya BR, Eryurek E. 1992. Application of neural network for sensory validation and plant monitoring. In *Neural Technology* **97**: 170176.

[74] Vafaie H, DeJong K. 1998. Feature Space Transformation Using Genetic Algorithms. In *IEEE Intelligent Systems* **13**(2): 57-65.

[75] Yao X. 1999. Evolving Artificial Neural Networks. In *Proceedings of the IEEE*, **87**(9):1423-1447.

[76] Yao X, Liu Y. 1997. A new evolutionary system for evolving artificial neural networks. In *IEEE Transactions on Neural Networks*, **8**(3):694-713.

[77] Zavgren C. 1985. Assessing the vulnerability to failure of American industrial firms: A logistics analysis. In *Journal of Business Finance and Accounting* (Spring): 19-45.

[78] Zupan B, Bohanec M, Demšar J, Bratko I. 1998. Feature Transformation by Function Decomposition. In *IEEE Intelligent Systems* **13**(2): 38-43.

[79] Zupan B, Demšar J, Kattan MW, Ohori M, Graefen M, Bohanec M, Beck JR. 2001. Orange and Decisions-At-Hand: Bridging predictive data mining and decision support. In *IDDM-2001: ECML/PKDD-2001 Workshop Integrating Aspects of Data Mining, Decision Support and Meta-Learning*, Freiburg, Giraud-Carrier C, Lavrac N, Moyle S, Kavšek B (eds); 151-162.

Publication 6

Nastac I, Costea A. 2004. A Retraining Neural Network Technique for Glass Manufacturing Data Forecasting. In *Proceedings of 2004 IEEE International Joint Conference on Neural Networks (IJCNN 2004)*, IEEE, Budapest, Hungary, July 25-29, 2004, Volume 4, Track: Time Series Analysis, pp. 2753-2758. ISBN: 0-7803-8359-1.

A Retraining Neural Network Technique for Glass Manufacturing Data Forecasting

Iulian Nastac
Turku Centre for Computer Science and
Institute for Advanced Management Systems Research
Lemminkäisenkatu 14B
FIN-20520 Turku, Finland
E-mail: inastac@abo.fi

Adrian Costea
Turku Centre for Computer Science and
Institute for Advanced Management Systems Research
Lemminkäisenkatu 14B
FIN-20520 Turku, Finland
E-mail: acostea@abo.fi

***Abstract*— This paper advances a retraining-neural-network-based forecasting mechanism that can be applied to complex prediction problems, such as the estimation of relevant process variables for glass manufacturing. The main purpose is to obtain a good accuracy of the predicted data by using an optimal feedforward neural architecture and well-suited delay vectors. The artificial neural network's (ANNs) ability to extract significant information provides a valuable framework for the representation of relationships present in the structure of the data. The evaluation of the output error after the retraining of an ANN shows that the retraining technique can substantially improve the achieved results.**

I. Introduction

The quality of the melting process in glass manufacturing is intricately dependent on many input variables [3]. There are influences that can be changed by the operator or by automatic controllers, but there are also disturbances that cannot be controlled. Artificial Neural Networks (ANNs) are modelling tools that have the ability to adapt to and learn complex topologies of inter-correlated multidimensional data [1], [4], [5]. Constructing reliable time series models for data forecasting is challenging, due to nonstationarities and nonlinear effects [2], [7], [8], [12], [13]. In this paper, we present our ANN model, which is useful when there is a huge amount of data that implies the presence of correlations across time.

The goal of our research was to find a practical mathematical model [3] that describes the relationship between 29 input variables (all of which are measurable influences), and 5 output variables that model a glass manufacturing system (Fig. 1).

All inputs and outputs vary dynamically, and large time-delays might occur. Changing an input variable may result in an output change that starts only a couple of hours later and goes on for up to several days [3].

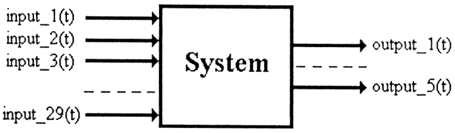


Fig. 1. Multi-input-multi-output system for glass quality.

The raw data consist of 9408 rows (timesteps) - one data row every 15 minutes during 14 weeks. For the first 13 weeks (8736 rows), both input and output data were used during the training process, and during the last week (14, which was the test week) we predicted the corresponding 672 rows of outputs.

The previous result [9], which we obtained at the EUNITE Competition 2003, has been further improved and extended.

The structure of this paper is as follows. Section II presents the problem that concerns model structure and data preprocessing. In the next section we introduce the retraining technique and explain our approach. The main features of our experimental results are given in Section IV, where we also present the actual improvement of the model architecture. Conclusions are given in the last section of the paper.

II. Model Structure and Data Preprocessing

The structural design of an ANN is a crucial factor that determines its performance [11]. A suitable choice for the global architecture of the model is not a trivial task, if one wants to make a good prediction. In Fig. 2 we present our idea of training a feedforward ANN such that the latter becomes a predictor. We use delayed rows of input data to simulate the current states of the outputs. For learning purposes, the network inputs consist of many blocks with several time-delayed values of the glass manufacturing system inputs and fewer blocks with system delayed outputs. The ANN target outputs consist of the current values of the glass manufacturing system outputs. Therefore, the network tries to match the current values of the outputs by adjusting a function of the past values of the inputs and outputs.

At moment t, each output is affected by the inputs at different past timesteps ($t\text{-}id_1$, ..., $t\text{-}id_n$), and also by the outputs at other past timesteps ($t\text{-}od_1$, ..., $t\text{-}od_m$). We denote by so called In_Del and Out_Del, two delay vectors that include the delays that we take into account:

$$In\_Del = [id_1, id_2, ..., id_n]$$

$$Out\_Del = [od_1, od_2, ..., od_m]$$

0-7803-8359-1/04/$20.00 ©2004 IEEE

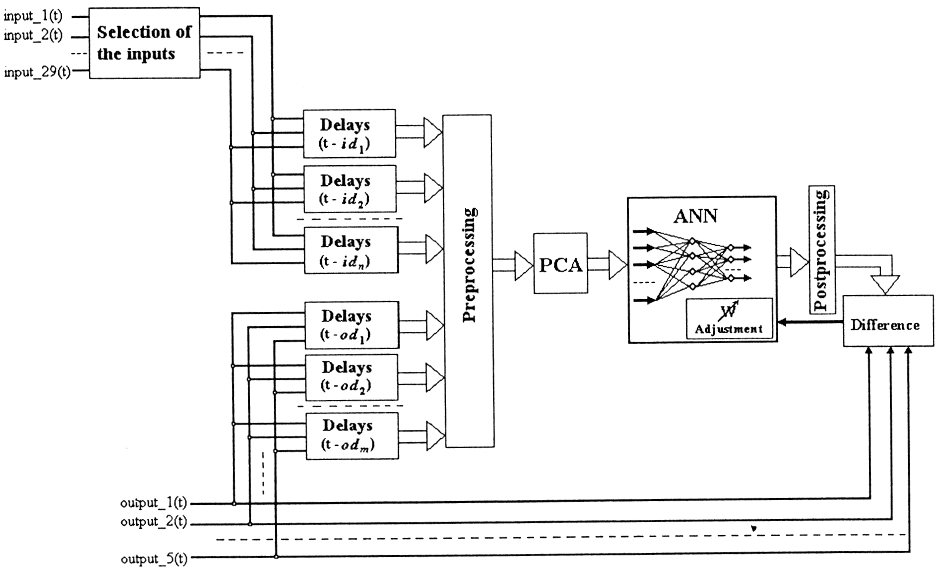


Fig. 2. Training process of the forecasting architecture.

For *In_Del*, we used various delay vectors with n = 7, 8 or 9 elements, whose values are in intervals that can cover one to three days. Regarding *Out_Del*, we employed different combinations with m = 3, 4 or 5 elements, whose values cover 2 to 5 hours. The distribution of the vector elements was similar to the Gamma distribution. The elements of each vector were ascendingly ordered. Consequently the maximum value of any vector is in the last position of the vector (id_n or od_m, respectively).

The recurrent relation performed by our model is as follows:

$$Y(t+1) = F(X(t+1-In\_Del(i)), Y(t-Out\_Del(j))) \quad (1)$$

where X and Y are vectors of input and output, respectively; $i = \overline{1,n}$; $j = \overline{1,m}$.

The block for input selection permits the choice of the most relevant inputs. According to the glass manufacturer's suggestion, a good option was to use the following inputs: 1, 2, 4, 10, 19, 20, 23, 29, as well as ($input\_1$ + $input\_2$ + ... + $input\_10$) and ($input\_20$ + $input\_21$ + ... + $input\_28$). The sums represent the heating energies (E_1 and E_2, respectively) of the melting glass furnace.

We used feedforward ANNs with one or two hidden layers. The ANN model depicted in Fig. 2 restricts the total possible training set (for model adaptation) to 8736 – id_n input-output pairs.

Once we decided all the influences on the output, at moment t, we applied Principal Component Analysis (PCA) [6] to reduce the dimensionality of the input space and to uncorrelate the inputs. Before applying PCA, we had preprocessed the inputs and outputs by using normalization. Data preprocessing prepares raw data for the forecasting model and transforms it into a format that will be more easily and effectively processed. We have applied the reverse process of normalization, in order to denormalize the simulated outputs. Data preprocessing and data postprocessing are essential steps of the knowledge discovery process in real world applications, and they greatly improve the network's ability to capture valuable information when they are correctly carried out [1], [4].

III. Training Procedure

The feature of "universal functional approximator" [5] brings the power and flexibility of neural networks to the process of learning complex patterns and relationships. However, the potential risk of using the universal approximator is the overfitting problem, since it is often easy to train a large network model to learn the peculiarities of the data, as well as the underlying relationship. Therefore, the balance between learning capability and generalization power is very important in neural network forecasting applications.

As the basic training algorithm, we used the Scale Conjugate Gradient (SCG) algorithm [8]. In order to avoid the overfitting phenomenon, we applied the early stopping method (*validation stop*) during the training process.

Next, the accuracy of the results was improved by applying the *retraining technique* [10] in a special way. This technique is a mechanism for extracting practical information directly from the weights of a reference ANN that had been already trained. This retraining procedure reduces the reference network weights (and biases) by a scaling factor γ, $0 < \gamma < 1$. The reduced weights are further used as the initial weights of a new training sequence, with the expectation of a higher accuracy. Briefly, the entire technique can be summarized by the following phases:

- Training an Artificial Neural Network in a natural way with *validation stop*, and with the weights initialized to small, uniformly distributed values;
- Reducing the first network weights and biases by a *scaling factor* γ $(0 < \gamma < 1)$;
- Retraining the network with the new initial weights;
- Comparing the *validation error* (or training error) in both cases.

Another advantage of this technique is reflected by a significant decrease in the number of training cycles, if compared to the classical training methods [10].

The data that we used for our model consisted of $8736 - id_n$ input-output pairs. As splitting criterion, we randomly chose approximately 85% of the data for training set, and the rest for validation.

Next we describe the *three steps* that we took to refine our model:

1) Firstly, we decided the proper number of hidden neurons (N_h). Each of the training sessions started with the weights initialized to small, uniformly distributed values [4], [10]. We chose the best model with respect to the smallest error between the desired and the simulated outputs. This error was calculated for $8736 - id_n$ data that included both training and validation sets. We tested several ANN architectures, with N_h having values in the vicinity of the geometric mean of the input number (N_i) and the output number (N_o) [1]:
$$\sqrt{N_i * N_o} - 10 \leq N_h \leq \sqrt{N_i * N_o} + 10 \qquad (2)$$
2) Secondly, we applied the retraining technique using the ANN architecture (with its associate training and validation sets) achieved during the previous step. We applied this technique for each value of γ $(\gamma = 0.1, 0.2, \ldots, 0.9)$, keeping the neural network that achieved the minimum error as the reference network. We repeated this step three times.
3) Thirdly, we applied the retraining technique again, with the only difference being that we randomly reconstructed the training and validation sets before each retraining sequence.

Decisive in choosing the best model during each step was the mean square error of the differences between the real and the simulated outputs of $8736 - id_n$ data rows, which included both training and validation sets.

After each of these steps, we obtained a new model. Consequently, for a single combination of delay vectors, we had three models. The above-mentioned steps were applied iteratively for different delay vectors. Moreover, for some combinations, which provided promising results, we repeated the experiments. We obtained a total of 69 different models. The criterion for choosing one of these 69 models was the minimum value of the error ERR [3] (see next section), but used for eight evaluation intervals.

IV. Forecasting Model

In the forecasting architecture, the outputs at moment t are affected by the outputs at the previous timesteps. During the training phases, we used the real data as the input.

In the next part, we tried to predict the 672 values of the outputs (week 14), in a sequential mode. Therefore (see Fig. 3), in order to produce the outputs at timestep t, the neural network used as input (in addition to the real inputs of the glass manufacturing system from different past timesteps) the estimated outputs that were calculated at previous steps, using other simulated outputs, and so on. Applying this iterative process, a forecast may be extended as many steps as required, yet running the risk that each step increases the forecasting error.

In order to have a "selection tool" for the best model, we split the output data between timestep $(id_n + 1)$ and timestep 8736 in eight distinct intervals of one week each. The "evaluation intervals" were as follows: $1000 - 1672, 2000 - 2672, \ldots, 8000 - 8672$.

For each interval, we computed the error ERR [3] that represents the accuracy of the approximation of all output data during the forecasting horizon of N timesteps:
$$ERR = \frac{1}{5}\sum_{i=1}^{5}\frac{100}{N}\sum_{k=1}^{N}\frac{|O_{Rki} - O_{Fki}|}{|O_{Rki}|}\cdot f(k) \qquad (3)$$
where

- N = 1344 (number of timesteps)
- O_{Rki} = real *output_i* at timestep k
- O_{Fki} = forecasted *output_i* at timestep k

and
$$f(k) = \frac{500}{500 + k} \qquad (4)$$
a weight function decreasing with the number of timestep k.

To measure the model quality, we calculated ERR_A as the average of eight ERR errors computed for the "evaluation intervals". Then, for each model we got an ERR_A value. All of the 69 models were tested against each other in terms of the ERR_A value that we used as the "selection tool". After that, we computed ERR_T (the error of the test interval) for the week 14, in order to see if the model with the minimum ERR_A is the same as the model with the minimum ERR_T. We noted a certain correlation between ERR_A and ERR_T (see Fig. 4), which means that ERR_A could be used as "selection tool" for the models, in case the test set is not available. The experimental results (for four models) are depicted in Table I, which also includes the significant parameters of the models.

The main improvement of the forecasting architecture, if we compare it to the one that we used for the EUNITE Competition [9], resulted from the application of the delay vector *Out_Del*, instead of the timestep $t - 1$.

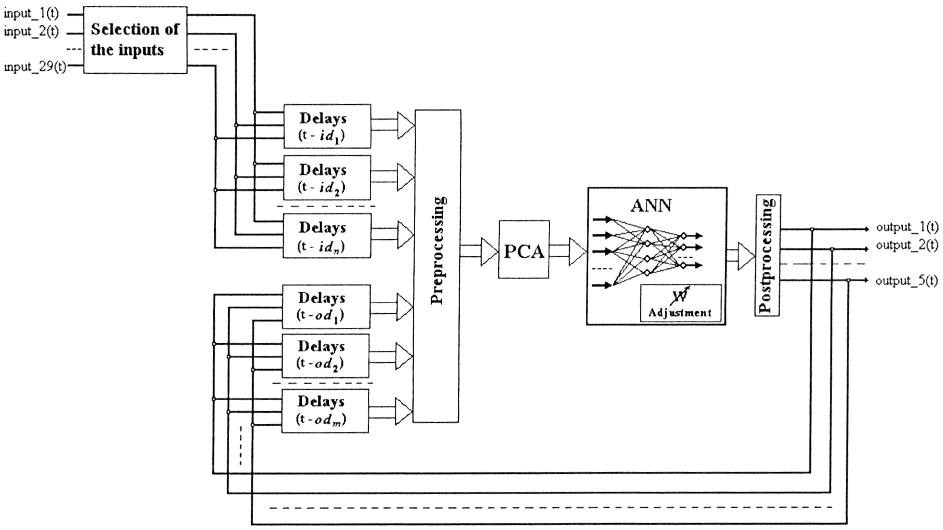


Fig. 3. Simulation process of the forecasting.

TABLE I
PARAMETERS OF FOUR SELECTED MODELS

Models	A	B	C	D
Inputs	All	All	All	1, 2, 4, 10, 19, 20, 23, 29, E_1, E_2
Prediction horizon	weeks 13&14	week 14	week 14	week 14
Size of PCA trans. matrix	132 × 237	139 × 281	113 × 257	48 × 105
ANN	132:35:5	139:35:5	113:45:5	48:30:19:5
In_Del	[10 20 35 55 80 120 185 290]	[5 10 20 30 45 65 95 145 210]	[1 3 6 10 15 22 31 45]	[1 3 6 10 15 22 31 45]
Out_Del	-	[0 4 10 20]	[0 2 6 12 20]	[0 2 6 12 20]
Step	2	3	2	2
ERR_A	0.3602	0.2680	**0.2151**	0.2514
ERR_T	0.4095	0.2877	**0.2702**	0.2832

Furthermore, it was possible to reduce the number of inputs using the block of input selection. This implied further that, after PCA preprocessing, the number of weights between the first and second layers of the ANN (where most of the weights were concentrated) decreased significantly. Under these circumstances, a new hidden layer can be added to the neural network architecture (e.g. model D in Table I), maintaining a similar ratio between the number of training samples and the total number of weights. Unfortunately, two hidden layers models did not show a better accuracy of the outputs compared to one hidden layer models.

The parameters of model A that we submitted to the competition are shown in Table I. This model with $ERR\_T = 0.4095$ got the third position according to the defined error criteria (first winner had $ERR\_T = 0.3418$).

Note that the last three columns of Table I, which correspond to the three new models, respectively, show the improved results. Moreover, the shapes of the neural network outputs (Fig. 7) are smoothed in comparison to the ones of the old forecasting architecture [9].

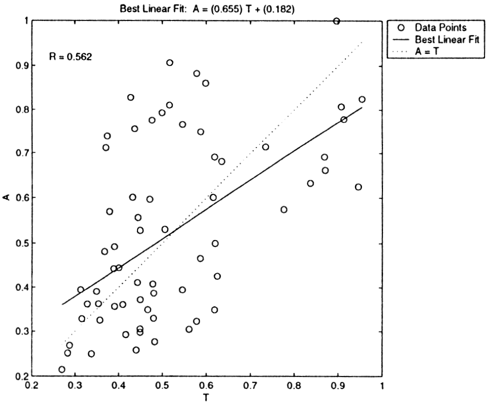


Fig. 4. Correlation between ERR_A and ERR_T.

Model C (Table I) was the best with respect to the value of ERR_T. After the training process (Fig. 5), the ANN outputs approximated the real data very well.

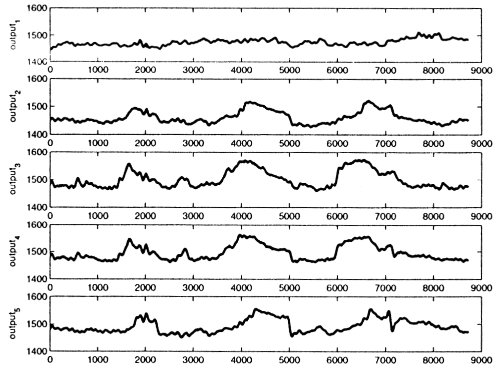


Fig. 5. Training result. The real outputs (thin lines) are completely covered by the simulated outputs (thick lines).

For this best model, we computed the outputs by using the iterative forecasting procedure, for the eight evaluation intervals (Fig. 6). Observe that, at the beginning of each interval, the forecasting is well performed. There are only few regions where the error of the prediction is visibly increased, at the end of some intervals.

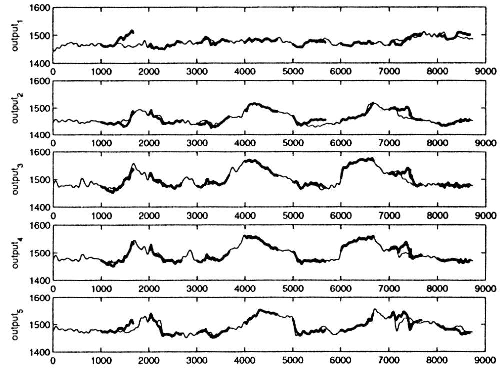


Fig. 6. Real outputs (thin lines) and simulated outputs (thick lines) after iterative forecasting process in eight intervals.

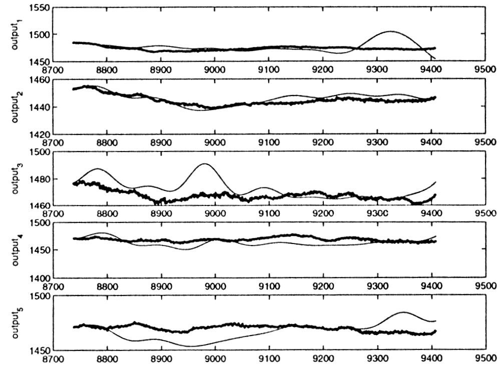


Fig. 7. Real outputs (thin lines) and simulated outputs (thick lines) that correspond to week 14.

The performance of the model C for the test interval (week 14) is shown in Fig. 7. Clearly, the accuracy of the outputs decreases in time, as each new forecasting step subsumes the errors of the previous predictions. Furthermore, the test interval (week 14) may exhibit new patterns that were not taken into consideration during the training process. In spite of these problems, the stability of the forecasting process can easily be noticed.

V. CONCLUSIONS

In this paper, we have designed a neural network tool for data prediction. Our method exploits the input-output dependence across time, and uses two delay vectors. We employed the PCA procedure in order to reduce the dimensionality of the input space and to un-correlate the inputs. The learning process was refined by applying the retraining procedure.

Choosing the best model is not an easy task, but it can be done if one uses ERR_A as the "selection tool", since there exists a certain correlation with ERR_T. The accuracy should be improved by using, as much as possible, training data that cover all the potential situations and, at the same time, more evaluation intervals.

We were limited by the memory and speed of our computers. We believe that if we use vectors with more than 9 elements, we can increase the performance of our tool.

At the same time, there are other efficient algorithms like Levenberg-Marquardt or Bayesian regularization, which also need a powerful machine to solve the problem. It is quite easy to replace the SCG algorithm used by our tool with one of the previously mentioned algorithms, as the basic level architecture and retraining procedure are independent of the training algorithm.

During the experiments, we noticed that the retraining technique exposed significant improvements in the achieved result. Current research targets the implementation of an adaptive system, which will be periodically retrained, in order to continuously learn the latest evolution of the glass melting process.

ACKNOWLEDGMENT

The authors would like to thank Dr. Katharina Lankers, Schott Glas Company, Germany, for the data and useful comments.

REFERENCES

[1] I. A. Basheer and M. Hajmeer, "Artificial neural networks: fundamentals, computing, design, and application", *Journal of Microbiological Methods*, Elsevier Science, vol. 43, pp. 3-31, 2000.

[2] V. L. Berardi and G. P. Zhang, "An Empirical Investigation of Bias and Variance in Time Series Forecasting: Model Considerations and Error Evaluation", *IEEE Transactions on Neural Networks*, vol. 14, pp. 668-680, 2003.

[3] (2003) Eunite Competition 2003: Prediction of product quality in glass manufacturing. [Online]. Available: *http://www.eunite.org/eunite/events/eunite2003/competition2003.pdf.*

[4] M. T. Hagan, H. B. Demuth, and M. Beale, *Neural Networks Design*, MA: PWS Publishing, Boston, 1996.

[5] K. Hornik, M. Stinchcombe, and H. White, "Multilayer feedforward networks are universal approximators", *Neural Networks*, vol. 2, pp. 359-366, 1989.

[6] Jackson, J.E., *A user guide to principal components*, John Wiley, New York, 1991.

[7] R. Lacroix, F. Salehi, X. Z. Yang, andK. M. Wade, "Effects of data preprocessing on the performance of artificial neural networks for dairy yield prediction and cow culling classification", *Transactions of the ASAE*, vol. 40(3), pp. 839-846, 1997.

[8] M. F. Moller, "A scaled conjugate gradient algorithm for fast supervised learning", *Neural Networks*, vol. 6, pp. 525-533, 1993.

[9] I. Nastac and A. Costea, "Advanced Data Forecasting Using Retraining Neural Network Technique", *TUCS Technical Report*, No. 542 (2003), Turku, Finland (Report of the result of the participation to the EUNITE Competition 2003). [Online]. Available: http://www.tucs.fi/Research/Series/serie.php?year=2003&type=techreport.

[10] I. Nastac and R. Matei, "Fast retraining of artificial neural networks", in *Rough Sets, Fuzzy Sets, Data Mining and Granular Computing*, Wang et al. (Eds.), Springer-Verlag in the series of Lecture Notes in Artificial Intelligence (LNAI 2639), pp. 458-462, 2003.

[11] P. C. Pendharkar, "A computational study on the performance of artificial neural networks under changing structural design and data distribution", *European Journal of Operational Research*, vol. 138, pp. 155-177, 2002.

[12] A. Weigend, D. Rumelhart, and B. Huberman, "Predicting the future: A connectionist approach", *Int. J. Neural Syst.*, vol. 3, pp. 193-209, 1990.

[13] G. Zhang, B.E. Patuwo, and M.Y. Hu, "Forecasting with artificial neural networks: The state of the art", *Int. J. Forecasting*, vol. 14, pp. 35-62, 1998.

Turku Centre for Computer Science
TUCS Dissertations

28. **Martin Büchi**, Safe Language Mechanisms for Modularization and Concurrency
29. **Elena Troubitsyna**, Stepwise Development of Dependable Systems
30. **Janne Näppi**, Computer-Assisted Diagnosis of Breast Calcifications
31. **Jianming Liang**, Dynamic Chest Images Analysis
32. **Tiberiu Seceleanu**, Systematic Design of Synchronous Digital Circuits
33. **Tero Aittokallio**, Characterization and Modelling of the Cardiorespiratory System in Sleep-disordered Breathing
34. **Ivan Porres**, Modeling and Analyzing Software Behavior in UML
35. **Mauno Rönkkö**, Stepwise Development of Hybrid Systems
36. **Jouni Smed**, Production Planning in Printed Circuit Board Assembly
37. **Vesa Halava**, The Post Correspondence Problem for Marked Morphisms
38. **Ion Petre**, Commutation Problems on Sets of Words and Formal Power Series
39. **Vladimir Kvassov**, Information Technology and the Productivity of Managerial Work
40. **Franck Tétard**, Managers, Fragmentation of Working Time, and Information Systems
41. **Jan Manuch**, Defect Theorems and Infinite Words
42. **Kalle Ranto**, Z_4-Goethals Codes, Decoding and Designs
43. **Arto Lepistö**, On Relations between Local and Global Periodicity
44. **Mika Hirvensalo**, Studies on Boolean Functions Related to Quantum Computing
45. **Pentti Virtanen**, Measuring and Improving Component-Based Software Development
46. **Adekunle Okunoye**, Knowledge Management and Global Diversity - A Framework to Support Organisations in Developing Countries
47. **Antonina Kloptchenko**, Text Mining Based on the Prototype Matching Method
48. **Juha Kivijärvi**, Optimization Methods for Clustering
49. **Rimvydas Rukšėnas**, Formal Development of Concurrent Components
50. **Dirk Nowotka**, Periodicity and Unbordered Factors of Words
51. **Attila Gyenesei**, Discovering Frequent Fuzzy Patterns in Relations of Quantitative Attributes
52. **Petteri Kaitovaara**, Packaging of IT Services – Conceptual and Empirical Studies
53. **Petri Rosendahl**, Niho Type Cross-Correlation Functions and Related Equations
54. **Péter Majlender**, A Normative Approach to Possibility Theory and Soft Decision Support
55. **Seppo Virtanen**, A Framework for Rapid Design and Evaluation of Protocol Processors
56. **Tomas Eklund**, The Self-Organizing Map in Financial Benchmarking
57. **Mikael Collan**, Giga-Investments: Modelling the Valuation of Very Large Industrial Real Investments
58. **Dag Björklund**, A Kernel Language for Unified Code Synthesis
59. **Shengnan Han**, Understanding User Adoption of Mobile Technology: Focusing on Physicians in Finland
60. **Irina Georgescu**, Rational Choice and Revealed Preference: A Fuzzy Approach
61. **Ping Yan**, Limit Cycles for Generalized Liénard-type and Lotka-Volterra Systems
62. **Joonas Lehtinen**, Coding of Wavelet-Transformed Images
63. **Tommi Meskanen**, On the NTRU Cryptosystem
64. **Saeed Salehi**, Varieties of Tree Languages
65. **Jukka Arvo**, Efficient Algorithms for Hardware-Accelerated Shadow Computation
66. **Mika Hirvikorpi**, On the Tactical Level Production Planning in Flexible Manufacturing Systems
67. **Adrian Costea**, Computational Intelligence Methods for Quantitative Data Mining

Turku
Centre *for*
Computer
Science

Lemminkäisenkatu 14 A, 20520 Turku, Finland | www.tucs.fi

University of Turku
- Department of Information Technology
- Department of Mathematics

Åbo Akademi University
- Department of Computer Science
- Institute for Advanced Management Systems Research

Turku School of Economics and Business Administration
- Institute of Information Systems Sciences

ISBN 951-29-4013-2
ISSN 1239-1883