TECHNICAL UNIVERSITY OF CRETE

DEPARTMENT OF PRODUCTION ENGINEERING & MANAGEMENT

POST GRADUATE PROGRAMME

SECTOR: ENGINEERING MANAGEMENT

DISSERTATION TITLE

CREDIT RISK MODELING WITH THE USE OF NEURAL NETWORKS AND GENETIC ALGORITHMS

A Thesis submitted to the School of Graduate Studies in partial fulfillment of the requirements
For the degree Master of Science

STUDENT: **KAMPIANAKIS IOANNIS / Register Number: 2008-019-002**

SUPERVISING COMMITTEE: Professor Constantine Zopounidis
Professor Christos Skiadas
Assistant Professor Michael Doumpos

CHANIA - CRETE

AUGUST 2011

TABLE OF CONTENTS

GREETINGS

In this point i would firstly wish to thank Dr. George Atsalakis for his important help, supporting me during the writing of my thesis and for the MatLab algorithm that he had given to me. I should also express my gratitude to the supervising committee and more specifically to Professor Konstantinos Zopounidis, for offering me the opportunity to study in this graduate programme, to Professor Christos Skiadas for the important knowledge that he had transferred to me during his course, 'Survival Data Analysis' and finally to Professor Michael Doumpos, as he had managed to make me assimilate the quantitative way of decision making, through his course 'Multicriteria Decision Making'. My special thanks belong of course to my parents for their emotional support during all these years of study.

INTRODUCTION

Banking in the international level has experienced more than three decades of rapid growth with significant increases in installment credit, family mortgages, business financing and credit card debt. Credit scoring models have been widely used by the financial industry during this time to improve cash flow and credit collections. The advantages of credit scoring include reducing the cost of credit analysis, enabling faster credit decisions, closer monitoring of existing accounts, and prioritizing collections. Today credit scoring is used by the vast majority of banks. In U.S. for example both the Federal Home Loan Mortgage Corporation and Federal National Mortgage Corporation are actively encouraging the use of credit scoring for mortgage origination.With the growth in financial services there have been mounting losses from delinquent loans. For example, in 1991 $1 billion of Chemical Bank's $6.7 billion in real estate loans were delinquent and the bank held $544 million in foreclosed property. In response, many financial organizations are developing new models to support the credit decision. The objective of these credit scoring models is increased accuracy which means that more creditworthy applicants are granted credit thereby increasing profits and non-creditworthy applicants are denied credit thus decreasing losses.

Linear discriminant analysis (LDA), a simple parametric statistical model, was one of the first credit scoring models. The appropriateness of LDA for credit scoring has been questioned because of the categorical nature of the credit data, the fact that the covariance matrices of the good and bad credit classes are often not equal, and due to the commonly occurring deviation from the Gaussian law in the observational data. More sophisticated models are being investigated today to overcome some of the deficiencies of the LDA model. A central concern of these applications is the need to increase the scoring accuracy of the credit decision. An improvement in accuracy of even a fraction of a percent translates into significant future savings. In the pursuit of even small improvements in credit scoring accuracy, researchers and practioniers have explored techniques from the field of Artificial Intelligence and Computer Science, such as the Neural Networks and the Evolutionary Algorithms.

The purpose of this research is to conduct a rigorous study of the dynamics of alternative credit risk modeling methodologies. The central focus of our study lies on testing the performance of a hybrid Neurogenetic approach, based on the use of a direct search nature inspired technique for the optimization of the synaptic weights of an Artificial Feed forward Neural Network, in order to make accurate forecasts of the probability of corporate default. We make use of a number of measures such as the computation of Type I and Type II errors, Forecasting Accuracies and the construction of the

ROC curve, in order to compare the proposed method, with other traditional techniques, such as Discriminant analysis, logistic regression and probit analysis. The validity of the experimental results is enhanced by the use of a 10-fold cross validation technique, in order to ensure the robustness of the derived conclusions.

Summarizing our findings, we should among others, denote the statistically significant superiority of the AI methods compared to the classic statistical techniques. Of great importance, is the observed dominance of the Genetic Optimization over the Backpropagation Algorithm as an alternative but more efficient way to estimate the structural parameters of a Neural Network. As a further step, for future research we would support the development of an even more advanced soft computing technique that would integrate the existing hybrid approach, with the fuzzy set theory. The recent technological leaps in terms of computing and processing power facilitate the use of computationally demanding techniques that are promising for further improvements of the forecasting accuracy of bankruptcy risk.

1. THEORETICAL FRAMEWORK

1.1 RISK MANAGEMENT IN BANKING

When discussing the challenges faced by financial institutions in managing risk, it is important to have a consistent definition of the term 'risk'. Risk is defined as the volatility of a corporation's market value. The definition that has been selected is as broad as possible. What is of interest are all decisions that may impact on a change in market value. This is consistent with the view that risk management is about optimizing the risk-reward tradeoff– not about minimizing the absolute level of risk. In practice, banks' exposures are asymmetric. This is particularly true for credit risk, where the upside consists of a small positive yield, and the downside consists of a loss that could range from zero to more than 100 per cent of the exposure. Given the importance of this downside risk, banks tend to focus their energies on understanding and managing the key drivers that determine financial loss. In doing this, they generally distinguish between three main types of risk:

1. Credit risk – the potential financial loss resulting from the failure of customers to honor fully the terms of a loan or contract. Increasingly, this definition is being expanded to include the risk of loss in portfolio value as a result of migration from a higher risk grade to a lower one.

2. Market risk – the risk to earnings arising from changes in interest rates or exchange rates, or from fluctuations in bond, equity or commodity prices. Banks are subject to market risk in both the management of their balance sheets and in their trading operations.

3. Operational risk – the potential financial loss as a result of a breakdown in day-to-day operational processes. Operational risk can arise from failure to comply with policies, laws and regulations, from fraud or forgery, or from a breakdown in the availability or integrity of services, systems or information.

When it comes to risk management, banks' investors are risk minimizers. They will translate any increase in perceived risk or uncertainty almost immediately into sell orders. If risk is measured by the volatility of market value, banks have more work to do. Financial institutions typically distinguish between three principal classes of risk, namely, credit, market and operational risk. For each class, it is necessary to derive a means of estimating expected loss (for credit risk) and unexpected losses (for all three risk groups) such that an appropriate amount of capital can be held

and the performance of each business assessed. The output of the risk measurement process is such that financial institutions usually allocate 55 per cent of their total capital to credit risk, 25 percent to operational risk and the remaining 20 per cent to market risk (both trading risk and balance sheet risk). There are three factors that drive expected and unexpected losses on a credit portfolio:

1. Customer default risk – determined by the risk-grade profile of the portfolio, the tenor of the exposures and the degree of exposure to country risk. Typically, the default rates are calibrated to those of Moody's Investors Service and Standard & Poor's Ratings Service.

2. Exposure – the amount that is likely to be outstanding at the time of default. This includes current drawn amounts as well as an allowance for contingent liabilities and undrawn lines.

3. Loss given default – determined by the level of security cover, the effectiveness of the work-out process and the credit cycle.

Banks are exposed to market risk via their trading activities and their balance sheets. The measurement of trading risk is probably the most advanced of the three main types of risks faced by banks. Development efforts were spearheaded by the publication of JP Morgan'sRiskMetrics™ in the mid 1990s, which created a much more open exchange of risk measurement methodologies.The common measure used to express market risk is 'value-at-risk' (VaR). VaR measures the maximum loss in portfolio value over a one-day holding period with 97.5 per cent confidence. In most commercial banks, the amount of capital allocated to trading risk is relatively small when compared to the other major risks. Greater dependence on technology and centralized operations mean that banks are becoming increasingly exposed to operational risk. Measuring operational risk requires identification of the underlying operational drivers or risk factors. Operational risk must be decomposed to those risks that are closely related to internal processes, people and systems (referred to as 'operational risks') and to those that are more related to the external environment (termed 'business or event risks').

In the area of risk management, corporate failure prediction plays a key role in examining credit loan applications since it enables banks to prevent themselves from insolvency due to bad loans in advance and helps them to sustain profitability from its proper lending practices. In addition, predicting corporate bankruptcy in a proper manner, a bank, can contribute to its community by supplying prospective companies with right fund corresponding to their respective financial

soundness. Moreover, the implementation of BASEL II Accord induces more severe competition among banks since it sets up more rigorous risk and capital management requirements to ensure that a bank holds capital reserves appropriate to the risk the bank is exposed through its lending and investment practices. With these reasons among others, banks now place their huge emphasis on identifying the risks they may face in the future more accurately. Due to rapidly changing corporate environment in recent years, however, a lot of complicated reasons behind corporate bankruptcy are newly emerging. Therefore, despite many existing methodologies for predicting corporate failure, it is worthwhile for academia as well as practitioners to continuously develop state-of-the-art methods reflecting various symptoms of corporate failure that may not be explained by the existing ones.

A classic example of the procedure for the credit risk evaluation of a company is the CAMEL rating system. The rating is based upon five critical elements of a company's operations: Capital Adequacy, Asset Quality, Management, Earnings, and Asset/Liability Management. This rating system is designed to take into account and reflect all significant financial, operational, and management factors examiners assess in their evaluation of a credit union's performance and risk profile. The CAMEL ratings should accurately reflect the condition of a corporation regardless of peer performance. Examiners use the financial ratios and trends displayed on the Financial Reports as well as other calculated ratios to guide them in assigning appropriate ratings. Credit analysts are also expected to use their professional judgment and consider both qualitative and quantitative factors when analyzing corporate performance. Since numbers are often lagging indicators of a credit union's condition, the examiner must also conduct a qualitative analysis of current and projected operations when assigning CAMEL ratings. Analysts distribute the amount and direction of risk exposure in seven categories: Credit, Interest Rate, Liquidity, Transaction experience, Compliance, Reputation, and Strategy.

Capital provides a cushion to fluctuations in earnings so that corporations can continue to operate in periods of loss or negligible earnings. It also provides a measure of reassurance to the members that the organization will continue to provide financial services. Likewise, capital serves to support growth as a free source of funds and provides protection against insolvency. Maintaining an adequate level of capital is a critical element. Under these circumstances, the need for a new legal framework that would define the minimum required capital, was born.

1.2 BASEL ACCORDS & CAPITAL ADEQUACY

The Basel Committee on Banking Supervision, with its revised capital adequacy framework "International Convergence of Capital Measurement and Capital Standards" (Basel Committee on Banking Supervision, 2005) – commonly known as **BASEL II** – proposes a flexible capital adequacy framework to encourage banks to make ongoing improvements in their risk assessment capabilities. Basel II uses a "three pillars" concept.

1. Minimum capital requirements (regulatory capital calculated in order to deal with credit, market and operational risk).

2. Supervisory review.

3. Market discipline.

BASEL I (1988) accord dealt with parts of each of these pillars. More specifically credit risk was dealt in a simpler manner based mainly in the standardized credit ratings, market risk was of minor importance and there was not any option for operational risk inclusion. According to this regulatory structure, virtually all private-sector loans are subject to the same 8% capital ratio with no account of the size of the loan, its maturity and, most importantly, the credit quality of the borrower. Thus, loans to a firm near bankruptcy are treated (in capital requirement terms) in the same fashion as loans to AAA borrowers. It also does not take into account the possible trade off, obtained through the diversification of the credit portfolio, but just defines the capital requirements, in an additive form. By the late 1990's it became clear, that there was a need for a major modification and update in the then, current Accord.

Under **BASEL II** (2004), the credit risk component can be calculated in three different ways of varying degree of sophistication, namely standardized approach, Foundation IRB and Advanced IRB. IRB stands for "Internal Rating-Based Approach". The standardized approach is similar to the current Accord: banks will be expected to allocate capital to their assets based on the risk weights assigned to various exposures. It improved on the original Accord by weighting those exposures based on each borrower's external credit risk rating. The standard risk weight categories are used under **BASEL I** are 0% for short term government bonds, 20% for exposures to OECD Banks, 50% for residential mortgages and 100% weighting on unsecured commercial loans. A 150% rating comes in for borrowers with poor credit ratings. The minimum capital requirement (the percentage of risk weighted assets to be held as capital) remains at 8%.

The IRB approach is a major innovation of the new accord as for the first time bank internal assessments of key risk drivers are primary inputs to the capital requirements. The close relationship between the inputs to the regulatory capital calculations and banks' internal risk assessments will facilitate a more risk sensitive approach to minimum capital. Changes in a client's credit quality will be directly reflected in the amount of capital held by banks. Thus, banks with better measure of their economic risks will be able to allocate capital more efficiently and more closely in line with their actual sensitivity to the underlying risks. In the IRB Approach to credit risk there are two variants: a foundation version and an advanced version. In the first version, banks must provide internal estimates of probability of default (PD)—which measures the likelihood that the borrower will default over a given time horizon. In addition, in the advanced approach, banks, subject to certain minimum conditions and disclosure requirements, can determine other elements needed to calculate their own capital requirements. They are the following:

1. Probability of Default (PD) - measures the likelihood that the borrower will default over a given time horizon. Probability of default (PD) per rating grade, gives the average percentage of obligors that default in this rating grade in the course of one year exposure at default.

2. Loss given default (LGD), which measures the proportion of the exposure that will be lost if a default occurs.

3. Exposure at default (EAD), which for loan commitments measures the amount of the facility that is likely to be drawn if a default occurs. (EAD) gives an estimate of the amount outstanding (drawn amounts plus likely future drawdown's of yet undrawn lines) in case the borrower defaults.

4. Maturity (M), which measures the remaining economic maturity of the exposure.

The two approaches differ primarily in terms of the inputs that are provided by the bank based on its own estimates and those that have been specified by the Committee. The risk weights and thus capital charges are determined through the combination of quantitative inputs provided by banks and formulas specified by the supervisor. Risk Weighted Assets (RWA) for credit risk are calculated for on- and off-balance-sheet exposures that are not captured in our market risk RWAs with the exception of OTC derivatives for which both market risk and credit risk RWAs are calculated. The calculations are consistent with the Advanced Internal Ratings Based (AIRB) approach and the

Internal Models Method (IMM) of Basel II, and were based on Exposure at Default (EAD), which is an estimate of the amount that would be owed to the bank at the time of a default, multiplied by each counterparty's risk weight. Under the Basel II AIRB approach, counterparty's risk weight is generally derived from a combination of the Probability of Default (PD), the Loss Given Default (LGD) and the maturity of the trade or portfolio of trades. Loss Given Default is the magnitude of likely loss on the exposure and is expressed as a percentage of the exposure. Loss Given Default is facility-specific because such losses are generally understood to be influenced by key transaction characteristics such as the presence of collateral and the degree of subordination. Loss Given Default is determined in one of two ways. Under the foundation methodology, LGD is estimated through the application of standard supervisory rules, which differentiate the level of Loss Given Default based upon the characteristics of the underlying transaction, including the presence and type of collateral. In the advanced methodology, the bank itself determines the appropriate Loss Given Default to be applied to each exposure, on the basis of robust data and analysis which is capable of being validated both internally and by supervisors.

In the following lines we provide a summary of the most commonly used formulas for the credit risk variables.

- Expected Loss: E (L, t) = PD×LGD×EaD
- Risk Weighted Assets: RWA= K × 12.5 × EAD
- LGD = {EAD – [NCV*(1-VDF)]} / EAD
- TCV = NCV – (NCV * VDF)
- EAD = NCV (1-VDF) & NCV = EAD / (1- VDF)

Where

- TCV: True Collateral Value
- NCV: Nominal Collateral Value
- VDF: Value Depreciation Factor

The derivation of risk-weighted assets is dependent on estimates of the PD, LGD, and EAD and, in some cases, on the effective maturity (M), for a given exposure. For exposures not in default, the formula for calculating risk-weighted assets is:

- Correlation (R) = 0.12 × (1 – EXP (-50 × PD)) / (1 – EXP (-50)) +
  0.24 × [1 – (1 – EXP (-50 × PD)) / (1 – EXP (-50))]

- Maturity adjustment (b) = (0.11852 – 0.05478 × ln(PD))^2
- Capital requirement (K) = [LGD × N[(1 – R)^-0.5 × G(PD) + (R / (1 – R))^0.5 × G(0.999)] – PD x LGD] x (1 – 1.5 x b)^-1 × (1 + (M – 2.5) × b)
- Risk-weighted assets (RWA) = K x 12.5 x EAD

Where

- N(x): The cumulative distribution function for a standard normal random variable
- G(z): The inverse cumulative distribution function for a standard normal random variable

PD estimates must be a long run average of one year realized default rates for borrowers in the grade. For corporate and bank exposures, the PD is the greater of the one year PD associated with the internal borrower grade to which that exposure is assigned, or 0.03%. Banks may use one or more of the three specific methods (internal default experience, mapping to external data and statistical default models) as well as other information and techniques as appropriate to estimate the average PD for each rating grade. Improvements in the rigor and consistency of credit risk measurement, the flexibility of models in responding to changes in the economic environment and innovations in financial products may produce estimates of credit risk that better reflect the credit risk of exposure. However, before a modeling approach could be used in the formal process of setting regulatory capital requirements for credit risk, regulators would have to be confident not only that models are being used to actively manage risk, but also that they are conceptually sound and empirically validated.

Returning to our **BASEL II** discussion, for operational risk, there are three different approaches - basic indicator approach or BIA, standardized approach or TSA, and the internal measurement approach (an advanced form of which is the advanced measurement approach). For market risk the preferred approach is VaR (value at risk).Under this new framework, regulators allow banks the discretion to calculate capital requirement for their banking books using "internal assessments" of key risk drivers, rather than the alternative regulatory standardized model. Thus the risk weights and capital charge are determined through the combination of quantitative inputs provided by bank and formulas specified by the Committee. Regarding the two other pillars of the new accord, the second pillar deals with the regulatory response to the first pillar, giving regulators much improved 'tools' over those available to them under Basel I. It also provides a framework for dealing with all the other risks a bank may face, such as systemic risk, pension risk, concentration

risk, strategic risk, reputational risk, liquidity risk and legal risk, which the accord combines under the title of residual risk. It gives banks a power to review their risk management system. The third pillar aims to promote greater stability in the financial system. Market discipline supplements regulation as sharing of information facilitates assessment of the bank by others including investors, analysts, customers, other banks and rating agencies. It leads to good corporate governance. The aim of the third pillar is to allow market discipline to operate by requiring lenders to publicly provide details of their risk management activities, risk rating processes and risk distributions. It sets out the public disclosures that banks must make that lend greater insight into the adequacy of their capitalization.

BASEL III refers to a new update to the Basel Accords that is under development. The term appeared in the literature as early as 2005 and is now in common usage anticipating this next revision to the Basel Accords. The Bank for International Settlements (BIS) itself began referring to this new international regulatory framework for banks as "Basel III" in September 2010. The draft Basel III regulations include: "tighter definitions of Common Equity; banks must hold 4.5% by January 2015, then a further 2.5%, totaling 7%.the introduction of a leverage ratio, a framework for counter-cyclical capital buffers, measures to limit counterparty credit risk, and short and medium-term quantitative liquidity ratios." In response to the recent financial crisis, the Basel Committee on Banking Supervision (BCBS) set forth to update their guidelines for capital and banking regulations: This consultative document presents the Basel Committee's proposals to strengthen global capital and liquidity regulations with the goal of promoting a more resilient banking sector. The objective of the Basel Committee's reform package is to improve the banking sector's ability to absorb shocks arising from financial and economic stress, whatever the source, thus reducing the risk of spillover from the financial sector to the real economy. The major proposed changes, due to be applied under the new capital adequacy framework are the following:

- Tier 1 capital (must be common shares and retained earnings)
- Tier 2 capital instruments will be harmonized.
- Tier 3 capital will be eliminated.
- The risk coverage of the capital framework will be strengthened.
- The capital requirements for counterparty credit exposures arising from banks' derivatives, repos and securities financing transactions will be strengthened.
- Additional incentives to move OTC derivative contracts to central counterparties (probably clearing houses)

- Incentives to strengthen the risk management of counterparty credit exposures.
- Introduction of a leverage ratio as a supplementary measure to the Basel II risk-based framework.
- Additional safeguards against model risk and measurement error by supplementing the risk based measure with a simpler measure that is based on gross exposures.
- Promote more forward looking provisions;
- Conserve capital to build buffers at individual banks and the banking sector that can be used in stress.
- Achieve the broader macro prudential goal of protecting the banking sector from periods of excess credit growth.
- Requirement to use long term data horizons to estimate probabilities of default,
- Downturn loss-given-default estimates, recommended in Basel II, to become mandatory
- Improved calibration of the risk functions, which convert loss estimates into regulatory capital requirements.
- Banks must conduct stress tests that include widening credit spreads in recessionary scenarios.
- Advocating a change in the accounting standards towards an expected loss (EL) approach
- Introduction of a global minimum liquidity standard for internationally active banks that includes a 30-day liquidity coverage ratio requirement underpinned by a longer-term structural liquidity ratio.

1.3 CREDIT RISK

Credit risk has long been an important and widely studied topic in bank lending decisions and profitability. For all banks, credit remains the single largest risk, difficult to offset, despite advances in credit measurement techniques and the diversification of portfolio. Continuing increases in the scale and complexity of financial institutions and in pace of their transactions demand that they employ sophisticated risk management techniques and monitor rapidly changing credit risk exposures. At the same time, fortunately, advances in information technology have lowered the cost of acquiring, managing and analyzing data, in an effort to build more robust and sound financial systems. In recent years, a number of the largest banks have developed sophisticated systems in an attempt to assess credit risk arising from important aspects of their business lines. What have been the benefits of the new model-based approach to risk measurement and management? The most important is that better risk measurement and management contribute to a more efficient capital allocation. When risk is better evaluated, it can be more accurately priced and it can be more easily spread among a larger number of market participants. The improvement in credit risk modeling has led to the development of new markets for credit risk transfer, such as credit derivatives and collateralized debt obligations (CDOs). These new markets have expanded the ways that market participants can share credit risk and have led to more efficient pricing of that risk.

Traditionally, there are two major research trends in bankruptcy prediction. One is investigating the situation of failure to find the symptoms. The other is comparing the prediction accuracy of the diverse classification methods. Bankruptcy prediction has long been an important and widely studied topic. The main impact of such research is in bank lending. Banks need to predict the probability of default of a potential counterparty before they extend a loan. This can lead to more sound lending decisions, and therefore result in significant capital savings. For the retail bankruptcy prediction problem, there is likewise an extensive amount of research. Credit risk has been the subject of much research activity, especially after realizing its practical necessity after a number of high profile bank failures in Asia.

Measuring credit risk accurately allows banks to engineer future lending transactions, so as to achieve targeted return/risk characteristics. The other benefit of the prediction of bankruptcies is for accounting firms. If an accounting firm audits a potentially troubled firm, and misses giving a warning signal (say a “going concern” opinion), then it faces costly law suits. The traditional

approach of banks for credit risk assessment is to produce an internal rating, which takes into account various quantitative as well as subjective factors, such as leverage, earnings, reputation, etc., through a scoring system. The problem with this approach is of course the subjective aspect of the prediction, which makes it difficult to make consistent estimates. Some banks, especially smaller ones, use the ratings issued by the standard credit rating agencies, such as Moody'sand Standard & Poor's. The problem with these ratings is that they tend to be reactive rather than predictive (for the agencies to change a rating of a debt, they usually wait until they have a considerably high confidence/evidence to support their decision. There is a need, therefore, to develop fairly accurate quantitative prediction models that can serve as very early warning signals for counterparty defaults. There are two main approaches to loan default/bankruptcy prediction. The first approach, the structural approach, is based on modeling the underlying dynamics of interest rates and firm characteristics and deriving the default probability based on these dynamics. The second approach is the empirical or the statistical approach. Instead of modeling the relationship of default with the characteristics of a firm, this relationship is learned from the data.

Sometimes a firm can become distressed and continue to operate in that condition for many years. On the other hand, some firms enter bankruptcy immediately after a highly distressing event, such as a major fraud. A number of factors influence these outcomes. They are audit, financial ratios, fraud indicators, start-up and stress which are measured by qualitative or quantitative variables. Bankruptcy occurs if the company cannot operate, pay liability, earn profits and obtain bad credits, etc. Forecasting bankruptcy can be thought of as a classification problem. With input variables as the financial and accounting data of a firm, we try to find out which category the firm belongs to bankruptcy or non-bankruptcy. In the last few decades quantitative methods known as credit scoring models have been developed for the credit granting decision. The objective of quantitative credit scoring models is to assign credit applicants to one of two groups: a "good credit" group that is likely to repay the financial obligation, or a "bad credit" group that should be denied credit because of a high likelihood of defaulting on the financial obligation. The first model employed for credit scoring, and a commonly used method today, is linear Discriminant analysis, a simple parametric statistical method. With the growth of the credit industry and the large loan portfolios under management today, the industry is actively developing more accurate credit scoring models. Even a fraction of a percent increase in credit scoring accuracy is a significant accomplishment. This effort is leading to the investigation of nonparametric statistical methods, classification trees, and neural network technology for credit scoring applications. Due to the proprietary nature of credit scoring, there is a paucity of research reporting the performance of commercial credit scoring applications. The

research that exists today focuses on two areas, the prediction of firm insolvency and the prediction of individual credit risk.

Bankruptcy prediction is the art of predicting bankruptcy and various measures of financial distress of public firms. The importance of the area is due in part to the relevance for creditors and investors in evaluating the likelihood that a firm may go bankrupt. The quantity of research is also a function of the availability of data: for public firms which went bankrupt or did not, numerous accounting ratios that might indicate danger can be calculated, and numerous other potential explanatory variables are also available. Consequently, the area is well-suited for testing of increasingly sophisticated, data-intensive forecasting approaches. A credit score is a numerical expression based on a statistical analysis of a person's credit files, to represent the creditworthiness of that person. A credit score is primarily based on credit report information typically sourced from credit bureaus. Lenders, such as banks and credit card companies, use credit scores to evaluate the potential risk posed by lending money to consumers and to mitigate losses due to bad debt. Lenders use credit scores to determine who qualifies for a loan, at what interest rate, and what credit limits. Lenders also use credit scores to determine which customers are likely to bring in the most revenue.

Over the last time, enormous strides have been made in the art and science of credit risk measurement. Banks have devoted increased attention to measuring credit risk and have made important gains, both by employing innovative and sophisticated risk modeling techniques and also by strengthening their more traditional practices. The specification of the process of default and rating migration is severely constrained by a lack of data on the historical performance of loans and other modeled variables. Most credit operations are not market to market and the predictive nature of a credit risk model does not derive from a statistical projection of future prices based on a comprehensive record of historical prices. The difficulties in specification are exacerbated by the longer term time horizons used in measuring credit risk, which suggest that many years of data, spanning multiple credit cycles, may be needed to estimate the process of default. Hence, in specifying models parameters, credit risk models require the use of simplifying assumptions and proxy data. The validation of credit risk models is fundamentally more difficult than the back testing of market risk models. Where market risk models typically employ a horizon of a few days, credit risk models generally rely on a time frame of 1 year or more; the longer holding period, coupled with the higher confidence intervals used in credit risk models, presents problems to model—builders in assessing the accuracy of their models.

2. CREDIT RISK MEASUREMENT

2.1 BANKRUPTCY PREDICTION

Prediction of corporate failure using past financial data is a well-documented topic. **Beaver (1966)** was one of the first researchers to study bankruptcy prediction and he had investigated the predictability of 14 financial ratios using 158 samples that consisted of failed and non-failed firms. Beaver's study was followed by **Altman (1968)** and his developed model based on the MDA to identify the companies into known categories. According to Altman, bankruptcy could be explained quite completely by using a combination of five (selected from an original list of 22) financial ratios. Altman utilized a paired sample design, which incorporated 33 pairs of manufacturing companies. The pairing criteria were predicated upon size and industrial classification. The classification of Altman's model based on the value obtained for the Z score had exhibited a predictive power of 96% for prediction one year prior to bankruptcy. These conventional statistical methods, however, have some restrictive assumptions such as the linearity, normality and independence among predictor or input variables. Considering that the violation of these assumptions for independent variables frequently occurs with financial data **(Deakin, 1976)**, the methods can have limitations to obtain the effectiveness and validity. Recently, a number of studies have demonstrated that artificial intelligence approaches that are less vulnerable to these assumptions, such as Neural Networks and Genetic Algorithms can be alternative methodologies for classification problems to which traditional statistical methods have long been applied. Thus we can define between two broad classes of models. The traditional and the soft computing models, the latter making use of the latest advances in the field of computer science and artificial intelligence.

A. Traditional Models

1. Expert systems. In an expert system, the credit decision is left to the local or branch lending officer. The person's expertise, subjective judgment and weighting of certain key factors are the most important determinants in the decision to grant credit. Bankers have relied on the so-called '5 Cs' of expert systems to assess credit quality: the expert analyzes these five key factors and performs a credit decision. These factors are: character (reputation of the firm), capital (leverage), capacity (volatility of the borrower's earnings), collateral and cycle (macroeconomic) conditions. These systems face two main problems: (i) human experts may be inconsistent and subjective in their

assessments, (ii) traditional expert systems specify no weighting scheme that would consistently order the '5 Cs' in terms of their relative importance in forecasting the default.

2. External rating systems. External credit ratings provided by firms specializing in credit analysis were first offered in the U.S. by Moody's in 1909. These companies offer bond investors access to low cost information about the creditworthiness of bond issuers. The usefulness of this information is not limited to bond investors. However, the rating system has been rather crude, with most loans rated as Pass-performing and only a minority of loans differentiated according to the four non-performing classifications.

3. Credit scoring systems. In regard with the credit scoring systems, the most commonly used traditional credit risk measurement model is the multiple Discriminant credit scoring analysis pioneered by **Altman (1968)**. The model identifies financial variables that have statistical explanatory powering differentiating "bad firms" from "good firms". Once the model's parameters are obtained, loan applicants are assigned a Z-score. In some circumstances, the score can be interpreted as a probability of default; in others, the score can be used as a classification system: it placed potential borrower into either a good or bad group, based on a score and a cut-off point. The credit scoring models can be subdivided in three main categories:

3.1 Models of linear Discriminant analysis, in particular, Altman Z-score model. The multivariate Analysis, through a discriminating function, permit to synthesize the value of more variables in a single Z value that compared with a cut-off point concurs to classify the loan applications into the groups of acceptance or rejection for the loan.

3.2 Models of linear regression: they identify, in a selected sample, some random variables (Xi,j) these variables reflect important information and they are used as independent variables in a linear regression in which the dependent variable is represented by the variable Z (that can assume 0 or 1 value alternatively). In this way, it identifies variables statistically meaningful in insolvency evaluating and it also estimates regression coefficients. This approach suffers from an important problem whenever the probability of default of one new borrower assumes external values to the interval (0 1).

3.3 Logit and probit models: the problem of linear model regarding the output not limited in interval (0,1) is solved by the model of logistic regression (logit) that uses an exponential transformation and

results of the regression analysis are included within this interval. The expression provides the conditional probability of finding the borrower i in the group of insolvent customers. The probit model only differs from the logit model as far as concerns the relative hypothesis to the distribution. It assumes that the distribution is the standardized normal and therefore F(Z) represents the accumulated function of the normal distribution. In the logit model F(Z) indicates the accumulated function of the logistic distribution, characterized from thicker tails. In the application, it does not determine important differences between the two models if not there are numerous extreme cases in the reference sample.

B. Soft Computing Models & Internal Approaches

The traditional credit scoring models rely on statistical restrictions and thus suffer from frequent violations among which we can refer to the deviation on the Normality Hypothesis and the Correlation between independent variables. The recent application of non-linear methods such as neural networks to credit risk analysis shows improvements on the traditional credit scoring models. The new approach credit risk models try to offer "internal model" approaches to measure the credit risk of a loan or a portfolio of loans. First, within the current generation of credit risk models, banks employ either of two conceptual definitions of credit loss, the default mode (DM) paradigm or the mark to market (MTM) paradigm. In the first paradigm a credit loss arises only if a borrower defaults within the planning horizon. In the absence of a default event, no credit loss would be incurred. In the case that a client defaults, the credit loss would reflect the difference between the bank's credit exposure and the present value of future net recoveries. In contrast to the DM paradigm, in the MTM models a credit loss can arise in response to the deterioration in an asset's credit quality. Given the rating transition Matrix associated with each client, Monte Carlo methods are generally used to simulate migration paths for each credit position in the portfolio. Second, there are different methodologies for the unconditional and conditional models. Unconditional approaches typically reflect customer or facility-specific information. Such models are currently not designed to capture business cycle effects, such as the tendency for internal ratings to improve (deteriorate) more during cyclical upturns (downturns). Instead, conditional models incorporate information on the state of the economy, such as levels and trends in indicators of economic and financial health, in domestic and international employment, in stock prices and interest rates, ecc. In these models, rating transition matrices are increased likelihood of an upgrade during an upswing in a credit cycle and vice versa. Finally, there are different techniques for measuring the interdependence of factors that contribute to credit losses. In measuring credit risk, the calculation of a measure of the dispersion of credit risk

requires consideration of the dependencies between the factors determining credit related losses, such as correlations among defaults or rating migrations, LGDs and exposures, both for the same borrower and among different borrowers. In the next lines we will describe in detail the quantitative measures, as well as the various documented techniques that are used today, by the vast majority of the financial institutions as well as the academic community.

2.2 FINANCIAL RATIOS

Financial ratios quantify many aspects of a business and are an integral part of the financial statement analysis. Financial ratios are categorized according to the financial aspect of the business which the ratio measures. Liquidity ratios measure the availability of cash to pay debt. Activity ratios measure how quickly a firm converts non-cash assets to cash assets. Debt ratios measure the firm's ability to repay long-term debt. Profitability ratios measure the firm's use of its assets and control of its expenses to generate an acceptable rate of return. Market ratios measure investor response to owning a company's stock and also the cost of issuing stock. Financial ratios may not be directly comparable between companies that use different accounting methods or follow various standard accounting practices. Most public companies are required by law to use generally accepted accounting principles for their home countries, but private companies, partnerships and sole proprietorships may not use accrual basis accounting. Large multi-national corporations may use International Financial Reporting Standards to produce their financial statements, or they may use the generally accepted accounting principles of their home country. There is no international standard for calculating the summary data presented in all financial statements, and the terminology is not always consistent between companies, industries, countries and time periods. Some useful abbreviations, commonly found in the accounting statements are the following: COGS = Cost of goods sold, or cost of sales, EBIT = Earnings before interest and taxes, EBITDA = Earnings before interest, taxes, depreciation, and amortization, EPS = Earnings per share. In the following lines we present some of the most important financial ratios used both in academic literature and in banking practice.

A. PROFITABILITY RATIOS

Profitability ratios measure the company's use of its assets and control of its expenses to generate an acceptable rate of return

1. Gross margin = Gross Profit / Net Sales

2. Operating margin = Operating Income / Net Sales

3. Net profit margin = Net Profit / Net Sales

4. Return on equity (ROE) = Net Income / Common Equity

5. Return on investment (ROI ratio or Du Pont Ratio) = Net Income / Common Equity

6. Return on assets (ROA) = Net Income / Total Assets

7. Return on capital (ROC) = EBIT (1-Tax Rate) / Invested Capital

8. Risk adjusted return on capital (RAROC) = Expected Return / Economic Capital

9. Basic Earnings Power Ratio = EBIT / Total Assets

B. LIQUIDITY RATIOS

Liquidity ratios measure the capability of the company to deal with its daily operational expenses obligations to other debtors.

1. Current ratio (Working Capital Ratio) = Current Assets / Current Liabilities

2. Acid-test ratio (Quick ratio) = Current Assets – Inventories / Current Liabilities

3. Cash ratio = Cash + Marketable Securities / Current Liabilities

4. Operation cash flow ratio = Operation Cash Flow / Total Debt

C. ACTIVITY RATIOS

Activity ratios measure how effectively the company can manage its accounts payable and receivable.

1. Average collection period = Accounts Receivable / Sales / 365

2. Operating Leverage = % Change in Net Operating Income / % Change in Sales

3. DSO Ratio = Accounts Receivable / Sales / 365

4. Average payment period = Accounts Payable / Credit Purchases / 365

5. Stock turnover ratio = Cost of Goods sold / Inventory

6. Receivables Turnover Ratio = Net Credit Sales / Net Receivables

D. DEBT RATIOS

Debt ratios measure the financial independence of the company as well as the adequacy of internal share capital.

1. Debt ratio = Total Liabilities / Total Assets

2. Debt to equity ratio = Long Term Debt + Leases / Common Equity

3. Long-term Debt to Assets = Long Term Debt / Total Assets

4. Time's interest-earned ratio / Interest Coverage Ratio = EBIT / Interest Expense

E. MARKET RATIOS

Market ratios measure the market pricing and valuation of the company as a whole as well as the estimation of the market regarding the future earnings of the company.

1. Earnings per share (EPS) = Net Earnings / Number of Shares

2. Payout ratio = Dividends / Net Earnings

3. Dividend Cover (the inverse of Payout Ratio) = Earnings per Share / Dividend per Share

4. P/E ratio = Market Price per Share / Earnings per Share

5. Dividend yield = Dividend / Market Price

6. Cash flow ratio or Price/cash flow ratio = Market Price per Share / PV of Cash Flow per Share

7. Price to book value ratio (P/B or PBV) = Market Price per Share / Account Price per Share

8. Price/sales ratio = Market Price / Sales

9. PEG ratio = Price per Earnings / Earnings per Share Growth

2.3 STATISTICAL METHODS

Starting in the late 1960s, the decision whether to grant credit to customers has gained more and more attention for credit industry as the credit industry has been experiencing double-digit growth rate during the past few decades. The objective of credit scoring models is to assign credit applicants to either a 'good credit' group that is likely to repay financial obligation or a 'bad credit' group whose application will be denied because of its high possibility of defaulting on the financial obligation. Besides, the evaluation performance can be improved by using credit scoring with streamlining the process and allowing the credit professional to focus only on unusual accounts. Moreover, the credit scoring can give the credit professional an exposure perspective, mitigate the risk flexibility, and reduce the cost of credit analysis. As a result, accounts with high probability of default can be monitored and necessary actions can be taken in order to prevent the account from being default. In response, various statistical methods have been proposed to support the credit decision. Generally, two essential linear statistical tools, Discriminant analysis and logistic regression, were most commonly applied to construct credit scoring models. Discriminant analysis is the first tool to be used in building credit scoring models. However, the utilization of linear Discriminant analysis (LDA) has often been criticized because of its assumption of the categorical nature of the credit data and the fact that the covariance matrices of the good and bad credit classes are unlikely to be equal. In addition to the LDA approach, logistic regression is an alternative to conduct credit scoring. Basically, the logistic regression model was emerged as the technique of choice in predicting dichotomous outcomes. For predicting dichotomous outcomes, logistic regression has been concluded as one of the most appropriate techniques.

2.3.1 DISCRIMINANT ANALYSIS

Discriminant analysis was first proposed by **Fisher (1936)** in the 1930s as a discrimination and classification tool. Nowadays, Discriminant analysis has been reported as the most commonly discussed and used statistical technique in modeling classification tasks. According to some attributes of the predictor variables, Discriminant analysis tends to look for the best linear combination of the predictor variables to classify the studying objects into two or more populations at the optimum accuracy. As to the statistical assumptions in implementing Discriminant analysis requires the data to be independent and normally distributed while the covariance matrix is also required to comply with the variation homogeneity assumption. If the covariance matrices of the given populations are not equal, then the separation surface of the Discriminant function is quadratic and hence in this case the quadratic Discriminant analysis (QDA) needs to be used. Despite the fact that LDA is only a special

case of QDA with stronger assumptions which should restrict its applications, in fact LDA has been reported to be a more robust method when the theoretical presumptions are violated. Given that the covariance matrices conforming to the prior assumptions of variation homogeneity, Fisher's linear Discriminant function is allowed to be used. The LDA can be expressed as $D=\beta_0+\beta_1X_1+\beta_2X_2+\ldots+\beta_nX_n$ Where D represents the Discriminant score, β_0 is the intercept term, and β_i (i=1,…,n) represents the β coefficient associated with the corresponding explanatory variable X_i (i=1,…,n). Discriminant analysis has been widely devoted to a considerably wide range of application areas, such as medicine, business, education, marketing research, finance, chemistry, biology, engineering and archaeology.

Linear Discriminant analysis (LDA) and the related Fisher's linear Discriminant are methods used in statistics, pattern recognition and machine learning to find a linear combination of features which characterize or separate two or more classes of objects or events. The resulting combination may be used as a linear classifier or, more commonly, for dimensionality reduction before later classification. LDA is closely related to ANOVA (analysis of variance) and regression analysis, which also attempt to express one dependent variable as a linear combination of other features or measurements. In the other two methods however, the dependent variable is a numerical quantity, while for LDA it is a categorical variable (i.e. the class label). LDA works when the measurements made on independent variables for each observation are continuous quantities. When dealing with categorical independent variables, the equivalent technique is Discriminant correspondence analysis. The Z-score formula for predicting bankruptcy was published in 1968 by Edward I. Altman and is a classic application of Discriminant analysis. The formula may be used to predict the probability that a firm will go into bankruptcy within two years. Z-scores are used to predict corporate defaults and an easy-to-calculate control measure for the financial distress status of companies in academic studies. The Z-score uses multiple corporate income and balance sheet values to measure the financial health of a company. The Z-score is a linear combination of four or five common business ratios, weighted by coefficients. The coefficients were estimated by identifying a set of firms which had declared bankruptcy and then collecting a matched sample of firms which had survived, with matching by industry and approximate size (assets). Discriminant analysis is computationally very sensitive to specific assumptions. In fact, we must make use of a wide range of diagnostics and statistical hypotheses tests to examine our data, before deciding to use this method.

1. Normal distribution. It is assumed that the data (for the variables) represent a sample from a multivariate normal distribution. You can examine whether or not variables are normally distributed with histograms of frequency distributions.

2. Homogeneity of variances/covariances. It is assumed that the variance/covariance matrices of variables are homogeneous across groups. Again, minor deviations are not that important; however, before accepting final conclusions for an important study it is probably a good idea to review the within-groups variances and correlation matrices.

3. Correlations between means and variances. The major threat to the validity of significance tests occurs when the means for variables across groups are correlated with the variances (or standard deviations). Intuitively, if there is large variability in a group with particularly high means on some variables, then those high means are not reliable.

4. Matrix ill-conditioning problem. Another assumption of Discriminant function analysis is that the variables that are used to discriminate between groups are not completely redundant. As part of the computations involved in Discriminant analysis, you will invert the variance/covariance matrix of the variables in the model. If any one of the variables is completely redundant with the other variables then the matrix is said to be ill-conditioned, and it cannot be inverted.

Finally, we should denote that it is valuable to use Discriminant analysis as a supporting tool for designing the topology of neural networks as we can learn more about the inner workings. Besides, as there is no theoretical method in determining the best input variables of a neural network model, the Discriminant analysis procedure can be implemented as a generally accepted method for determining a good subset of input variables when many potential variables are considered and thus giving statistical support in deciding the input vector of the designed neural network model.

2.3.2 LOGIT ANALYSIS

We usually observe violations of the conditions of the classical econometric regression models. (Normality of the distribution, linearity, homoskedasticity). At the same time quite often the predictions for the dependent variable (p, probability of bankruptcy) are not in the interval [0.1]. The solution to the problems came to connecting the probability p of the explanatory variables (a linear combination vx) through a function that gives the p values in [0.1]. The logistic regression is a method of multivariate statistical analysis) using a set of independent variables (independent variables) to investigate the movement of a categorical dependent variable. The logistic regression is useful in situations where we predict the presence or absence of an attribute or an event. A logistic function or logistic curve is a common sigmoid curve and it can model the "S-shaped" curve (abbreviated S-curve) of growth of some population P. The initial stage of growth is approximately exponential then, as saturation begins, the growth slows, and at maturity, growth stops.

Logistic regression is widely used in statistical modeling technique. The probability of a dichotomous outcome is related to a set of potential predictor variables in the form: $\log[p/(1-p)]=\beta_0+\beta_1x_1+\beta_2x_2+\ldots+\beta_ix_i$ where p is the probability of the outcome of interest, β_0 is the intercept term, and β_i (i=1,…,n) represents the β coefficient associated with the corresponding explanatory variable x_i (i=1,…,n). The dependent variable is the logarithm of the odds, {log [p/(1−p)]}, which is the logarithm of the ratio of two probabilities of the outcome of interest. These variables are usually selected for inclusion by using some form of backward or forward stepwise regression technique even thought these selection techniques may be prone to problems. And the maximization of the likelihood function is usually applied as the convergent criterion to estimate the coefficients of corresponding parameters when the logistic regression models are utilized. The logistic regression model does not necessarily require the assumptions of LDA.

One advantage of LDA is that ordinary least-square estimation procedure can be implemented to estimate the coefficients of the linear Discriminant function, whereas maximum likelihood methods are required for the estimation of logistic regression models. Another advantage of LDA over logistic regression is that prior probabilities and misclassification costs can easily be incorporated into the LDA approach. Logistic Regression, compared with LDA, requires much more data to achieve stable, meaningful results. It does however, have several advantages over Discriminant analysis, such as that it is more robust, the independent variables don't have to be normally distributed, or have equal variance in each group, it does not assume a linear relationship

between the independent variable and the dependent variable, it may handle nonlinear effects, you can add explicit interaction and power terms there is no homogeneity of variance assumption, and finally normally distributed error terms are not assumed.

A simple logistic function may be defined by the formula

$$P(t) = \frac{1}{1 + e^{-t}}$$

Where the variable P might be considered to denote a population and the variable t might be thought of as time. For values of t in the range of real numbers from $-\infty$ to $+\infty$, an S-curve is obtained. In practice, due to the nature of the exponential function e^{-t}, it is sufficient to compute t over a small range of real numbers such as [−6, +6].

2.3.3 PROBIT ANALYSIS

Probit Analysis (Conditional Probability Approach - CPA) makes use of a likelihood function against the proposed multi-dimensional discrimination analysis and has the advantage that it suffers less from problems of violation hypotheses. In this method, our objective it to assess the likelihood of a one particular event, e.g. a firm going bankrupt. We do not make a dichotomous analysis between bankrupt or non bankrupt. There is a relaxation of the restrictive assumptions of discriminatory analysis. Discriminant analysis is based on the normality of the distribution of data, but assumes randomness in sampling, in contrast to the classical logistic regression. Probit analysis, shares the same principles, with logit analysis, except from the distributional hypothesis. In probability theory and statistics, the probit function is the inverse cumulative distribution function (CDF), or quantile function associated with the standard normal distribution. It has applications in exploratory statistical graphics and specialized regression modeling of binary response variables. The standard normal distribution is commonly denoted as N(0,1) and it's CDF as Φ(z). Function Φ is a continuous, monotone increasing sigmoid function whose domain is the real line and range is (0,1). The probit function gives the 'inverse' computation, generating a value of an N(0,1) random variable, associated with specified cumulative probability. The idea of probit was published in by **Chester Ittner Bliss (1934)** in an article in Science on how to treat data such as the percentage of a pest killed by a pesticide. Bliss proposed transforming the percentage killed into a "probability unit" (or "probit") which was linearly related to the modern definition (he defined it arbitrarily as equal to 0 for 0.0001 and 10 for 0.9999). He included a table to aid other researchers to convert their kill percentages to his

probit, which they could then plot against the logarithm of the dose and thereby, it was hoped, obtain a more or less straight line.

2.3.4 GOMPIT ANALYSIS (WEIBULL REGRESSION)

The Weibull distribution, also called Gompit, is an asymmetric distribution, strongly negatively skewed, approaching zero only slowly, and 1 more rapidly than the probit and logit models. It is calculated by the following formula: $p_i = 1 - \exp(-\exp(x_i\beta + \beta_0))$

This distribution is used for classification in survival analysis and comes from "extreme value theory. Weibull regression is suitable for analyzing survival data in a regression-like format. This Model offers several advantages:

1. The analyst can estimate survival probabilities for individuals, together with confidence intervals. These help him interpret and describe results.

2. A single parameter describes whether individuals have decreasing, stable, or increasing risk (hazard) functions. This helps test theoretical predictions about rising or falling risks.

3. The method is an M-estimate (from robustness theory), which makes available several practical results.

2.4 ARTIFICIAL INTELLIGENCE METHODS

During the last twenty years, the field of artificial intelligence, has made significant breakthroughs, and continues to expand in depth and number of applications. Nowadays, we commonly find in literature, the term 'Soft Computing', which has to do with the application of computer science and programming in solving difficult problems of quantitative nature. Characteristic examples of this recent trend are among others, the use of Neural Networks and Genetic Algorithms. NNs fundamentally differ from parametric statistical models. Parametric statistical models require the developer to specify the nature of the functional relationship such as linear or logistic between the dependent and independent variables. Once an assumption is made about the functional form, optimization techniques are used to determine a set of parameters that minimizes the measure of error. In contrast, NNs with at least one hidden layer use data to develop an internal representation of the relationship between variables so that a priori assumptions about underlying parameter distributions are not required. As a consequence, better results might be expected with NNs when the relationship between the variables does not fit the assumed model. In the lines that follow we provide a thorough description of the aforementioned techniques.

2.4.1. NEURAL NETWORKS

Introduction

An artificial neural network is a system based on the operation of biological neural networks, in other words, is an emulation of biological neural system. An important aspect of the artificial neural networks is that there are different architectures, which consequently requires different types of algorithms, but despite to be an apparently complex system, a neural network is relatively simple. In general, neural networks serve two important functions: as pattern classifiers and as nonlinear adaptive filters. We will provide a brief overview of the theory, learning rules, and applications of the most important neural network models. An Artificial Neural Network is an adaptive, most often nonlinear system that learns to perform a function (an input/output map) from data. Adaptive means that the system parameters are changed during operation, normally called the training phase. After the training phase the Artificial Neural Network parameters are fixed and the system is deployed to

solve the problem at hand (the testing phase). The Artificial Neural Network is built with a systematic step-by-step procedure to optimize a performance criterion or to follow some implicit internal constraint, which is commonly referred to as the learning rule. The input/output training data are fundamental in neural network technology, because they convey the necessary information to "discover" the optimal operating point. The nonlinear nature of the neural network processing elements (PEs) provides the system with lots of flexibility to achieve practically any desired input/output map, i.e., some Artificial Neural Networks are universal mappers.

An input is presented to the neural network and a corresponding desired or target response set at the output (when this is the case the training is called supervised). An error is composed from the difference between the desired response and the system output. This error information is fed back to the system and adjusts the system parameters in a systematic fashion (the learning rule). The process is repeated until the performance is acceptable. It is clear from this description that the performance hinges heavily on the data. If one does not have data that cover a significant portion of the operating conditions or if they are noisy, then neural network technology is probably not the right solution. On the other hand, if there is plenty of data and the problem is poorly understood to derive an approximate model, then neural network technology is a good choice. In artificial neural networks, the designer chooses the network topology, the performance function, the learning rule, and the criterion to stop the training phase, but the system automatically adjusts the parameters. So, it is difficult to bring a priori information into the design, and when the system does not work properly it is also hard to incrementally refine the solution. But ANN-based solutions are extremely efficient in terms of development time and resources, and in many difficult problems artificial neural networks provide performance that is difficult to match with other technologies.

Biological Origins

Artificial neural networks emerged after the introduction of simplified neurons by **McCulloch & Pitts (1943)**. These neurons were presented as models of biological neurons and as conceptual components for circuits that could perform computational tasks. The basic model of the neuron is founded upon the functionality of a biological neuron.

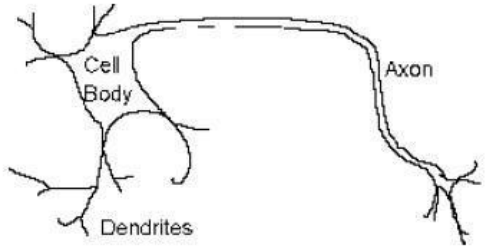

Fig. 1: The structure of a simple neural cell.

The neuron has four main regions to its structure. The cell body, or soma, has two offshoots from it, the dendrites, and the axon, which end in presynaptic terminals. The cell body is the heart of the cell, containing the nucleus and maintaining protein synthesis. A neuron may have many dendrites, which branch out in a treelike structure, and receive signals from other neurons. A neuron usually has only one axon which grows out from a part of the cell body called the axon hillock. The axon conducts electric signals generated at the axon hillock down its length. These electric signals are called action potentials. The other end of the axon may split into several branches, which end in a presynaptic terminal. Action potentials are the electric signals that neurons use to convey information to the brain. All these signals are identical. Therefore, the brain determines what type of information is being received based on the path that the signal took. The brain analyzes the patterns of signals being sent and from that information it can interpret the type of information being received. Myelin is the fatty tissue that surrounds and insulates the axon. Often short axons do not need this insulation. There are uninsulated parts of the axon. These areas are called Nodes of Ranvier. At these nodes, the signal traveling down the axon is regenerated. This ensures that the signal traveling down the axon travels fast and remains constant (i.e. very short propagation delay and no weakening of the signal). The synapse is the area of contact between two neurons. The neurons do not actually physically touch. They are separated by the synaptic cleft, and electric signals are sent through chemical interaction. The neuron sending the signal is called the presynaptic cell and the neuron receiving the signal is called the postsynaptic cell.

The signals are generated by the membrane potential, which is based on the differences in concentration of sodium and potassium ions inside and outside the cell membrane. Neurons can be classified by their number of processes (or appendages), or by their function. If they are classified by the number of processes, they fall into three categories. Unipolar neurons have a single process (dendrites and axon are located on the same stem), and are most common in invertebrates. In bipolar neurons, the dendrite and axon are the neuron's two separate processes. Bipolar neurons have a subclass called pseudo-bipolar neurons, which are used to send sensory information to the spinal

cord. Finally, multipolar neurons are most common in mammals. Examples of these neurons are spinal motor neurons, pyramidal cells and Purkinje cells (in the cerebellum). If classified by function, neurons again fall three separate categories. The first group is sensory, or afferent, neurons, which provide information for perception and motor coordination. The second group provides information (or instructions) to muscles and glands and is therefore called motor neurons. The last group, interneuronal, contains all other neurons and has two subclasses. One group called relay or projection interneurons have long axons and connect different parts of the brain. The other group called local interneurons are only used in local circuits.

The Mathematical Model

When creating a functional model of the biological neuron, there are three basic components of importance. First, the synapses of the neuron are modeled as weights. The strength of the connection between an input and a neuron is noted by the value of the weight. Negative weight values reflect inhibitory connections, while positive values designate excitatory connections. The next two components model the actual activity within the neuron cell. An adder sums up all the inputs modified by their respective weights. This activity is referred to as linear combination. Finally, an activation function controls the amplitude of the output of the neuron. An acceptable range of output is usually between 0 and 1, or -1 and 1.

Mathematically, this process is described in the figure

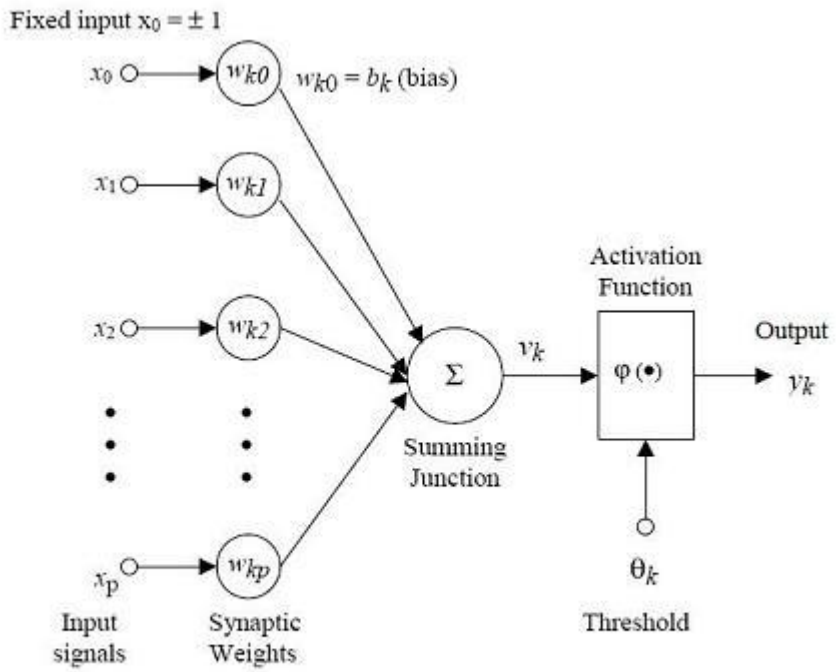

Fig. 2: The Connectionist nature of the architecture of an Artificial Neural Network

From this model the interval activity of the neuron can be shown to be:

$$v_k = \sum_{j=1}^{p} w_{kj} x_j$$

The output of the neuron, y_k, would therefore be the outcome of some activation function on the value of v_k.

Activation functions

As mentioned previously, the activation function acts as a squashing function, such that the output of a neuron in a neural network is between certain values (usually 0 and 1, or -1 and 1). In general, there are three types of activation functions, denoted by Φ (.)

1. Threshold Function which takes on a value of 0 if the summed input is less than a certain threshold value (v), and the value 1 if the summed input is greater than or eq

$$\varphi(v)=\begin{cases}1 & \text{if } v \geq 0\\ 0 & \text{if } v < 0\end{cases}$$

2. Piecewise-Linear function. This function again can take on the values of 0 or 1, but can also take on values between that depending on the amplification factor in a certain region of linear operation.

$$\varphi(v)=\begin{cases}1 & v \geq \frac{1}{2}\\ v & -\frac{1}{2} > v > \frac{1}{2}\\ 0 & v \leq -\frac{1}{2}\end{cases}$$

3. Sigmoid function. This function can range between 0 and 1, but it is also sometimes useful to use the -1 to 1 range. An example of the sigmoid function is the hyperbolic tangent function.

$$\varphi(v)=\tanh\left(\frac{v}{2}\right)=\frac{1-\exp(-v)}{1+\exp(-v)}$$

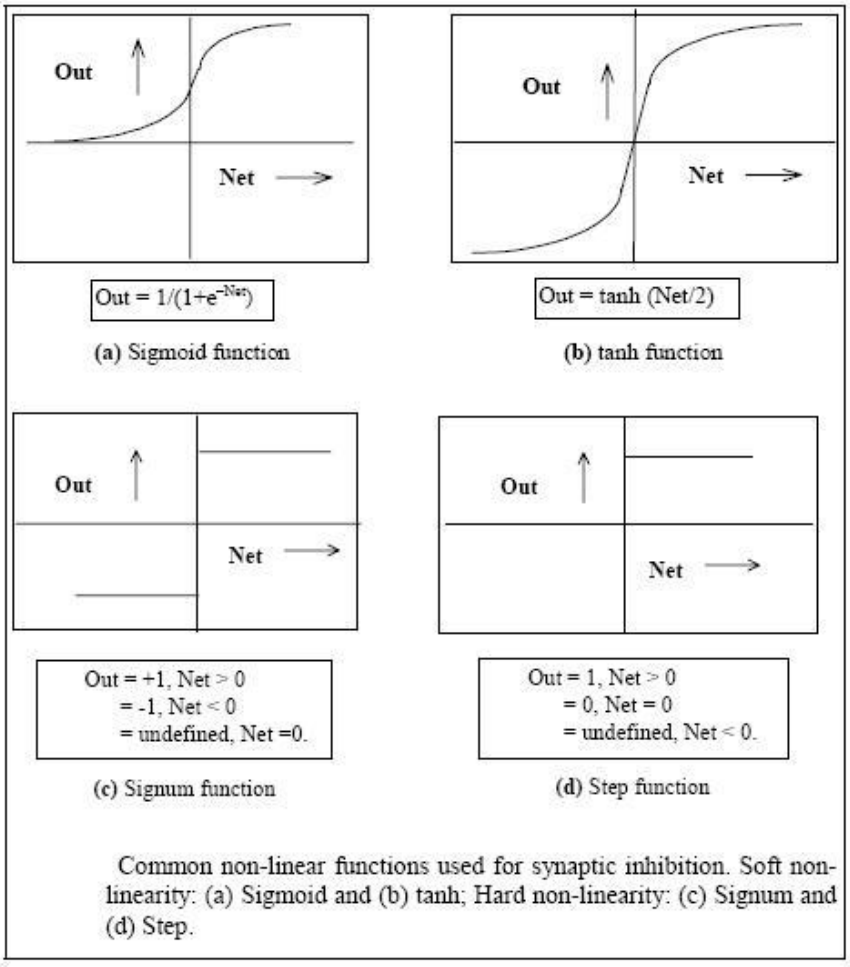

Fig. 3: Categorization of Activation Functions

Common non-linear functions used for synaptic inhibition. Soft non-linearity: (a) Sigmoid and (b) tanh; Hard non-linearity: (c) Signum and (d) Step.

Processing units - Neurons

Each unit performs a relatively simple job: receive input from neighbors or external sources and use this to compute an output signal which is propagated to other units. Apart from this processing, a second task is the adjustment of the weights. The system is inherently parallel in the sense that many units can carry out their computations at the same time. Within neural systems it is useful to distinguish three types of units: input units which receive data from outside the neural network, output units, which send data out of the neural network, and hidden units) whose input and output signals remain within the neural network. During operation, units can be updated either

synchronously or asynchronously. With synchronous updating, all units update their activation simultaneously, with asynchronous updating, each unit has a (usually fixed) probability of updating its activation at a time t, and usually only one unit will be able to do this at a time.

Neural Network topologies

1. Feed-forward neural networks, where the data from input to output units is strictly feed forward. The data processing can extend over multiple (layers of) units, but no feedback connections are present, that is, connections extending from outputs of units to inputs of units in the same layer or previous layers.

2. Recurrent neural networks that do contain feedback connections. Contrary to feed-forward networks, the dynamical properties of the network are important. In some cases, the activation values of the units undergo a relaxation process such that the neural network will evolve to a stable state in which these activations do not change anymore.

Training of Artificial Neural Networks

A neural network has to be configured such that the application of a set of inputs produces the desired set of outputs. Various methods to set the strengths of the connections exist. One way is to set the weights explicitly, using a priori knowledge. Another way is to 'train' the neural network by feeding it teaching patterns and letting it change its weights according to some learning rule. We can categorize the learning situations as following:

1. Supervised learning in which the network is trained by providing it with input and matching output patterns.

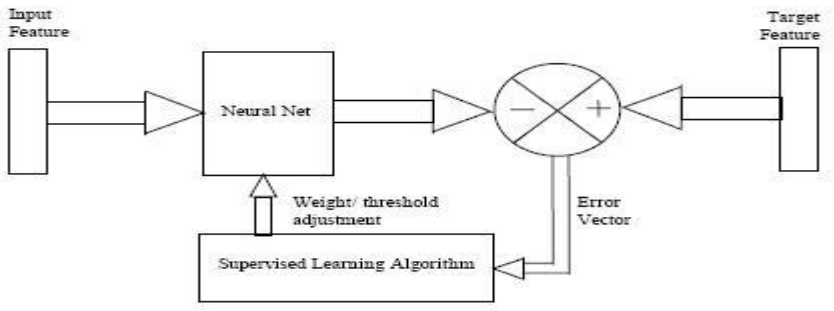

Fig. 4: A Supervised Feed Forward Artificial Neural Network

2. Unsupervised learning or Self-organization in which an (output) unit is trained to respond to clusters of pattern within the input. In this paradigm the system is supposed to discover statistically salient features of the input population. Unlike the supervised learning paradigm, there is no a priori set of categories into which the patterns are to be classified; rather the system must develop its own representation of the input stimuli.

3. Reinforcement Learning: This type of learning may be considered as an intermediate form of the above two types of learning. Here the learning machine does some action on the environment and gets a feedback response from the environment. The learning system grades its action good (rewarding) or bad (punishable) based on the environmental response and accordingly adjusts its parameters. Generally, parameter adjustment is continued until an equilibrium state occurs, following which there will be no more changes in its parameters. The self organizing neural learning may be categorized under this type of learning.

Optimization of Weights

Both learning paradigms supervised learning and unsupervised learning result in an adjustment of the weights of the connections between units, according to some modification rule. Virtually all learning rules for models of this type can be considered as a variant of the Hebbian learning rule suggested by **Hebb (1949).** The basic idea is that if two units j and k are active simultaneously, their interconnection must be strengthened. If j receives input from k, the simplest version of Hebbian learning prescribes to modify the weight w_{jk} with

$$\Delta w_{jk} = \gamma y_j y_k,$$

Where y is a positive constant of proportionality representing the learning rate. Another common rule does not use the actual activation of unit k but the difference between the actual and desired activation for adjusting the weights:

$$\Delta w_{jk} = \gamma y_j (d_k - y_k),$$

where d_k is the desired activation. This is often called the delta rule.

Multi-layer feed-forward networks

A feed-forward network has a layered structure. Each layer consists of units which receive their input from units from a layer directly below and send their output to units in a layer directly above the unit. There are no connections within a layer. The Ni inputs are fed into the first layer of hidden units. The activation of a hidden unit is a function Fi of the weighted inputs plus a bias, as given in the following equation:

$$y_k(t+1) = \mathcal{F}_k(s_k(t)) = \mathcal{F}_k\left(\sum_j w_{jk}(t)\, y_j(t) + \theta_k(t)\right),$$

The output of the hidden units is distributed over the next layer of hidden units, until the last layer of hidden units, of which the outputs are fed into a layer of output units.

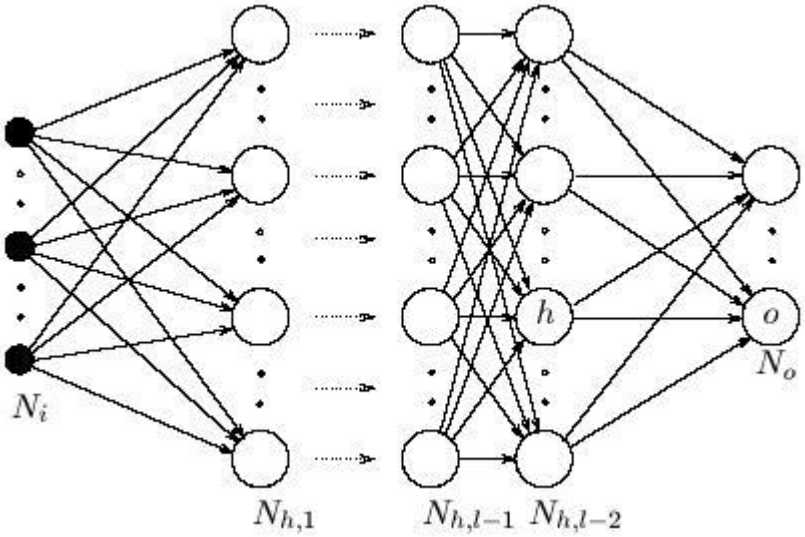

Fig. 5: The Architecture of a Multi Layer Perceptron

Back Propagation

A key feature of neural networks is an iterative learning process in which data cases (rows) are presented to the network one at a time, and the weights associated with the input values are adjusted each time. After all cases are presented, the process often starts over again. During this learning phase, the network learns by adjusting the weights so as to be able to predict the correct class label of input samples. Neural network learning is also referred to as "connectionist learning," due to connections between the units. The most popular neural network algorithm is back-propagation algorithm proposed in the 1980's. The Backpropagation algorithm was first proposed by **Paul Werbos (1974)**. It was however due to **Rumelhart and McClelland (1986)** that BackProp became

widely used. Once a network has been structured for a particular application, that network is ready to be trained. To start this process, the initial weights are chosen randomly. Then the training, or learning, begins. The network processes the records in the training data one at a time, using the weights and functions in the hidden layers, then compares the resulting outputs against the desired outputs. Errors are then propagated back through the system, causing the system to adjust the weights for application to the next record to be processed. This process occurs over and over as the weights are continually tweaked. During the training of a network the same set of data is processed many times as the connection weights are continually refined. Networks will not converge if there is not enough data to enable complete learning. Ideally, there should be enough data so that part of the data can be held back as a validation set.

When a learning pattern is clamped, the activation values are propagated to the output units, and the actual network output is compared with the desired output values, we usually end up with an error in each of the output units. Let's call this error e_o for a particular output unit o. We have to bring e_o to zero The simplest method to do this is the greedy method: we strive to change the connections in the neural network in such a way that, next time around, the error eo will be zero for this particular pattern. We know from the delta rule that, in order to reduce an error, we have to adapt its incoming weights according to.

$$\Delta w_{ho} = (d_o - y_o)y_h.$$

When we only apply this rule, the weights from input to hidden units are never changed, and we do not have the full representational power of the feed-forward network. In order to adapt the weights from input to hidden units, we again want to apply the delta rule. In this case, however, we do not have a value for δ for the hidden units. This is solved by the chain rule which does the following: distribute the error of an output unit o to all the hidden units that is it connected to, weighted by this connection. Differently put, a hidden unit h receives a delta from each output unit o equal to the delta of that output unit multiplied by the weight of the connection between those units. The application of the generalized delta rule thus involves two phases: During the first phase the input x is presented and propagated forward through the network to compute the output values for each output unit. This output is compared with its desired value do, resulting in an error signal for each output unit. The second phase involves a backward pass through the network during which the error signal is passed to each unit in the network and appropriate weight changes are calculated.

- The weight of a connection is adjusted by an amount proportional to the product of an error signal δ, on the unit k receiving the input and the output of the unit j sending this signal along the connection:

$$\Delta_p w_{jk} = \gamma \delta_k^p y_j^p.$$

- If the unit is an output unit, the error signal is given by

$$\delta_o^p = (d_o^p - y_o^p)\, \mathcal{F}'(s_o^p).$$

- Take as the activation function F the 'sigmoid' function as defined

$$y^p = \mathcal{F}(s^p) = \frac{1}{1 + e^{-s^p}}.$$

- In this case the derivative is equal to

$$\begin{aligned} \mathcal{F}'(s^p) &= \frac{\partial}{\partial s^p} \frac{1}{1 + e^{-s^p}} \\ &= \frac{1}{(1 + e^{-s^p})^2} (-e^{-s^p}) \\ &= \frac{1}{(1 + e^{-s^p})} \frac{e^{-s^p}}{(1 + e^{-s^p})} \\ &= y^p (1 - y^p). \end{aligned}$$

Such that the error signal for an output unit can be written as:

$$\delta_o^p = (d_o^p - y_o^p)\, y_o^p (1 - y_o^p).$$

- The error signal for a hidden unit is determined recursively in terms of error signals of the units to which it directly connects and the weights of those connections. For the sigmoid activation function:

$$\delta_h^p = \mathcal{F}'(s_h^p) \sum_{o=1}^{N_o} \delta_o^p w_{ho} = y_h^p (1 - y_h^p) \sum_{o=1}^{N_o} \delta_o^p w_{ho}.$$

The learning procedure requires that the change in weight is proportional to the true gradient descent requires when infinitesimal steps are taken. The constant of proportionality is the learning rate. For

practical purposes we choose a learning rate that is as large as possible without leading to oscillation. One way to avoid oscillation at large is to make the change in weight dependent of the past weight change by adding a momentum term:

$$\Delta w_{jk}(t+1) = \gamma \delta_k^p y_j^p + \alpha \Delta w_{jk}(t),$$

Where t indexes the presentation number and α is a constant which determines the effect of the previous weight change. Although, theoretically, the back-propagation algorithm performs gradient descent on the total error only if the weights are adjusted after the full set of learning patterns has been presented, more often, the learning rule is applied to each pattern separately, i.e., a pattern p is applied, Ep is calculated, and the weights are adapted (p = 1, 2,..... P).

Problems in the use of Neural Networks

1. Depending on which dataset is used, a high ratio of training-to-validation data does not yield meaningful learning.

2. The values fed to the input layer of a neural network are usually between '0' to '1'. However, with credit evaluation, the numerical values (input values) representing the attributes of a credit applicant vary marginally in value, and if a simple normalization process is applied to the whole dataset, say by dividing each value in the set by the largest recorded value, then much information would be lost across the different attributes.

3. Another problem with using neural networks in financial applications is the computational cost. The simplest MLP neural network has three-layers (input, hidden and output). As more layers are added, the computational cost and the processing time increase.

The Number of Hidden Neurons and Hidden Layers

Deciding the number of hidden neurons in layers is a very important part of deciding the overall neural network architecture. Though these layers do not directly interact with the external environment, they do however, have a tremendous influence on the final output. Both the number of hidden layers and number of neurons in each of these hidden layers must be considered. Using too few neurons in the hidden layers will result in something called under fitting. Under fitting occurs when there are too few neurons in the hidden layers to adequately detect the signals in a complicated data set. Using too many neurons in the hidden layers can result in over fitting. A large number of

neurons in the hidden layer can increase the time it takes to train the network. The amount of training time can increase enough so that it is impossible to adequately train the neural network. Obviously some compromise must be reached in the optimal number of neurons in the hidden layers. There are many rule-of-thumb methods for determining the correct number of neurons to use in the hidden layers. Some of them are summarized as follows.

- The number of hidden neurons should be in the range between the size of the input layer and the size of the output layer.
- The number of hidden neurons should be 2/3 of the input layer size, plus the size of the output layer.
- The number of hidden neurons should be less than twice the input layer size.

There are two methods that can be used to organize a trial and error search for the optimum network architecture, the "forward" and "backward" selection methods. The first method, begins by selecting a small number of hidden neurons. This method usually begins with only two hidden neurons. Then the neural network is trained and tested. The number of hidden neurons is then increased and the process is repeated so long as the overall results of the training and testing improved. The "forward selection method" is summarized in figure ??.

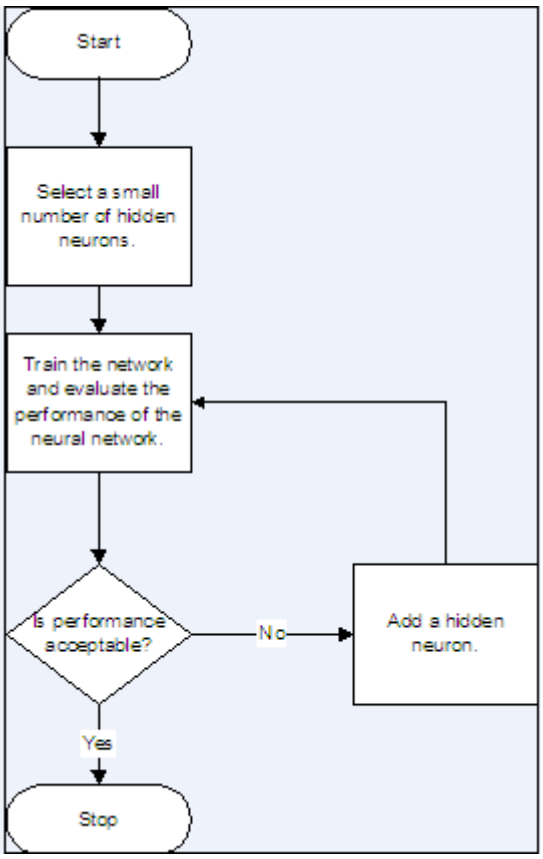

Fig. 6: Selecting the number of hidden neurons with forward selection

The second method, the "backward selection method", begins by using a large number of hidden neurons. Then the neural network is trained and tested. This process continues until the performance improvement of the neural network is no longer significant. Regardless of what the method is called, the resulting weights are virtually impossible for humans to understand. Patterns may be observable in some rare cases, but generally they appear to be random numbers. There is no quantifiable, best answer to the layout of the network for any particular application. There are only general rules picked up over time, such as the following:

- As the complexity in the relationship between the input data and the desired output increases, the number of the processing elements in the hidden layer should also increase.
- If the process being modeled is separable into multiple stages, then additional hidden layer(s) may be required.
- The amount of training data available sets an upper bound for the number of processing elements in the hidden layer(s). To calculate this upper bound, use the number of cases in the training data set and divide that number by the sum of the number of nodes in the input and output layers in the network. Then divide that result again by a scaling factor between five and ten. If we use too many artificial neurons the training set will be memorized. If that happens, generalization of the data will not occur, making the network useless on new data sets.

2.4.2 GENETIC ALGORITHMS

Introduction

Traditionally, mathematical optimization problems have been solved using analytical or indirect methods, which use analytical derivatives in order to find the optimum. Some of these methods include: Lagrange multipliers, Newton’s methods, and quadratic programming. In contrast, direct methods use a numerical rather than an analytical approach to solve an optimization problem. This involves calculating the value of the objective function F(y) for a number of different solution values y1, y2, y3... yn. By starting at some initial value(s), direct methods calculate successive values of the objective function for different values of y.Direct methods include among others: exhaustive sequential search, random search, simplex method, adaptive random search, simulated annealing and

genetic algorithms. **Holland (1975)** proposed a class of computational models, called Genetic Algorithms (GA), that mimic the biological evolution process for solving problems in a wide domain. The mechanisms under GA have been analyzed and explained later by many other scientists. Genetic Algorithms has three major applications, namely, intelligent search, optimization and machine learning. A Genetic Algorithm operates through a simple cycle of stages:

1. Creation of a population of strings (chromosomes)

2. Evaluation of each string (via a fitness function)

3. Selection of best strings (ranking)

4. Genetic manipulation to create new population of strings (crossover & mutation)

In most applications of GAs we are interested in solving an optimization problem. In this sense, an optimization problem is a search problem, in which a search space must be explored in order to maximize or minimize a cost or fitness function. In order to solve a search problem we therefore require a search strategy - a method for searching the space of possible solutions. Three of the simplest would be:

1. Enumerative search, an algorithm that enumerates all possible solutions to a given problem.

2. Calculus-based search - an algorithm that uses the derivative of the search space surface to either
2.1 Solve for the turning points.
2.2 Calculate the gradient of the surface in order to effect gradient descent or ascent (depending on whether we wish to minimize or maximize a given cost function).

3. Random search, where the algorithm samples the space of possible solutions and takes the best one found.

There are limitations with all three methods. First, enumeration is too slow. In most cases an enumeration of the search space would take hundreds of years in computation time (even for the fastest parallel machines that exist). Calculus-based techniques are also restrictive. Not only do they require a well-formed equation for the search problem, the equation must also be solvable either for turning points or for the derivative. In most problems the equation does not exist, the derivative does not exist, the equation is highly discontinuous, or the equation is unsolvable. In all such cases

calculus cannot be used. A random search, on the other hand, can always be used, but would seem to be the method of last resort. A random search fails to exploit any regularity that may exist within the search problem, and therefore may not be the most efficient use of computational resource. GA techniques have advantages over traditional non-linear solution techniques that cannot always achieve an optimal solution. A simplified comparison of the GA and the traditional solution techniques is illustrated in Figure?? Non-linear programming solvers generally use some form of gradient search technique to move along the steepest gradient until the highest point (maximization) is reached. In the case of linear programming, a global optimum will always be attained (). However, non-linear programming models may be subject to problems of convergence to local optima, or in some cases, may be unable to find a feasible solution. This largely depends on the starting point of the solver. A starting point outside the feasible region may result in no feasible solution being found, even though feasible solutions may exist. Other starting points may lead to an optimal solution, but it is not possible to determine if it is a local or global optimum. Hence, the modeler can never be sure that the optimal solution produced using the model is the "true" optimum.

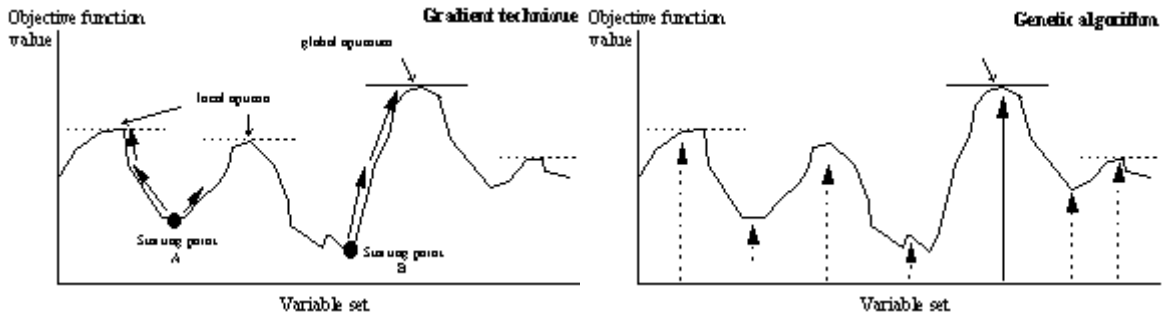

Fig. 7: Comparison of gradient search technique and genetic algorithm approach. Click above to download the images.

In their broadest sense genetic algorithms are a class of stochastic search algorithms that have been used to counter some of these problems. In the last few years there has been increasing interest in stochastic search techniques that combine elements of a random search with a deterministic search heuristic. The Genetic Algorithm (GA) is one such example. The principal components of a GA are as follows:

1. A fitness evaluation (a cost function) to guide the search while no auxiliary knowledge such as the derivative is required.

2. An encoding of the search space rather than the parameters themselves.

3. A population of points to search the space rather than a single point.

4. A probabilistic transition (movement) rules rather than deterministic rules.

The genetic algorithm (GA) is a search heuristic that mimics the process of natural evolution. This heuristic is routinely used to generate useful solutions to optimization and search problems. Genetic algorithms belong to the larger class of evolutionary algorithms (EA), which generate solutions to optimization problems using techniques inspired by natural evolution, such as inheritance, mutation, selection, and crossover. In a genetic algorithm, a population of strings (called chromosomes or the genotype of the genome), which encode candidate solutions (called individuals, creatures, or phenotypes) to an optimization problem, evolves toward better solutions. Traditionally, solutions are represented in binary system, as strings of 0s and 1s, but other encodings are also possible. The evolution usually starts from a population of randomly generated individuals and happens in generations. In each generation, the fitness of every individual in the population is evaluated, multiple individuals are stochastically selected from the current population (based on their fitness), and modified (recombined and possibly randomly mutated) to form a new population. The new population is then used in the next iteration of the algorithm. Commonly, the algorithm terminates when either a maximum number of generations has been produced, or a satisfactory fitness level has been reached for the population. If the algorithm has terminated due to a maximum number of generations, a satisfactory solution may or may not have been reached.

A typical genetic algorithm requires:

1. A genetic representation of the solution domain,
2. A fitness function to evaluate the solution domain.

Problem Structuring

A standard representation of the solution is as an array of bits. Arrays of other types and structures can be used in essentially the same way. The main property that makes these genetic representations convenient is that their parts are easily aligned due to their fixed size, which facilitates simple crossover operations. Variable length representations may also be used, but crossover implementation is more complex in this case. Tree-like representations are explored in genetic programming and graph-form representations are explored in evolutionary programming. The fitness function is defined over the genetic representation and measures the quality of the represented

solution. The fitness function is always problem dependent. A representation of a solution might be an array of bits, where each bit represents a different object, and the value of the bit (0 or 1) represents whether or not the object is used. In order to solve an optimization problem using a genetic algorithm, potential solutions or candidates are usually represented by vectors consisting of binary digits or bits. In general, the binary representation of an individual candidate is given by x = [x1; x2; x3; xb] where the value of each of the b elements is either a zero or one. This representation is based on the binary number system. Thus a particular candidate with a binary representation xi has a corresponding decimal equivalent value. Although the common approach is to use binary digits, some genetic algorithms represent candidates using real numbers. Thus, vectors consist of elements given by real numbers values instead of zeros and ones. This approach is employed in certain problems where there are a large number of different parameters.

Selection

Once we have the genetic representation and the fitness function defined, GA proceeds to initialize a population of solutions randomly, and then improve it through repetitive application of mutation, crossover, inversion and selection operators. Initially many individual solutions are randomly generated to form an initial population. The population size depends on the nature of the problem, but typically contains several hundreds of possible solutions. Traditionally, the population is generated randomly, covering the entire range of possible solutions (the search space). Occasionally, the solutions may be "seeded" in areas where optimal solutions are likely to be found. During each successive generation, a proportion of the existing population is selected to breed a new generation. Individual solutions are selected through a fitness-based process, where fitter solutions (as measured by a fitness function) are typically more likely to be selected. Certain selection methods rate the fitness of each solution and preferentially select the best solutions. Other methods rate only a random sample of the population, as this process may be very time-consuming. Most functions are stochastic and designed so that a small proportion of less fit solutions are selected. This helps keep the diversity of the population large, preventing premature convergence on poor solutions. Popular and well-studied selection methods include roulette wheel selection and tournament selection. The original method developed by **Holland (1975)** involves selecting candidates according to a probability distribution. The probability of a particular candidate being chosen is determined by its performance relative to the entire population. Thus the distribution is skewed towards the better performing candidates, giving them a greater chance of being selected. This method known as roulette wheel selection is not ideal in relatively small populations since there could be a

disproportionately large number of poorly performing candidates chosen for selection due to the random nature by which candidates are selected. One of these alternatives is the tournament selection method. This involves selecting two or more candidates at a time and then choosing the better performing candidate from the pair or group. For example, in a two party tournament selection two candidates will be chosen at random from the population. These two candidates are compared and the candidate with the greater performance is selected. This method captures the reasoning behind the theory of survival of the fittest, since two candidates participate in a live-or-die tournament, where only the better candidate lives while the other dies. Another approach is the genitor selection method, which is a ranking based procedure. This approach involves ranking all individuals according to performance and then replacing the poorly performing individuals by copies of the better performing individuals.

Reproduction

The next step is to generate a second generation population of solutions from those initially selected through genetic operators: crossover (also called recombination), and/or mutation. For each new solution to be produced, a pair of "parent" solutions is selected for breeding from the pool selected previously. By producing a "child" solution using the above methods of crossover and mutation, a new solution is created which typically shares many of the characteristics of its "parents". New parents are selected for each new child, and the process continues until a new population of solutions of appropriate size is generated. Although reproduction methods that are based on the use of two parents are more "biology inspired", some research suggests more than two "parents" are better to be used to reproduce a good quality chromosome. These processes ultimately result in the next generation population of chromosomes that is different from the initial generation. Generally the average fitness will have increased by this procedure for the population, since only the best organisms from the first generation are selected for breeding, along with a small proportion of less fit solutions, for reasons already mentioned above. Although Crossover and Mutation are known as the main genetic operators, it is possible to use other operators such as regrouping, colonization-extinction, or migration in genetic algorithms. This generational process is repeated until a termination condition has been reached. Common termination conditions are among others, a solution that satisfies some minimum criteria (in term of fitness), a fixed number of generations are reached, an allocated budget (computation time/financial cost) is reached. A simple generational genetic algorithm procedure includes the following steps:

1. Choose the initial population of individuals

2. Evaluate the fitness of each individual in that population

3. Select the best-fit individuals for reproduction

4. Breed new individuals through crossover and mutation operations to give birth to offspring

5. Evaluate the fitness of new individuals

6. Replace least-fit population with new individuals

7. Repeat on this generation until termination.

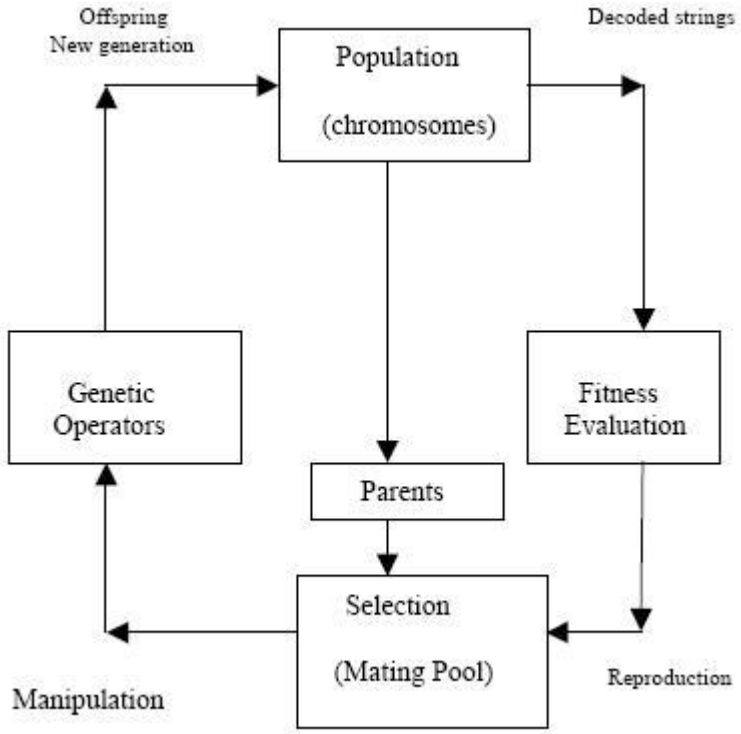

Fig. 8: The Genetic Algorithm Procedure

Each cycle in Genetic Algorithms produces a new generation of possible solutions for a given problem. In the first phase, an initial population, describing representatives of the potential solution, is created to initiate the search process. The elements of the population are encoded into bit-strings, called chromosomes. The performance of the strings, often called fitness, is then evaluated with the

help of some functions, representing the constraints of the problem. Depending on the fitness of the chromosomes, they are selected for a subsequent genetic manipulation process. It should be noted that the selection process is mainly responsible for assuring survival of the best-fit individuals. After selection of the population strings is over, the genetic manipulation process consisting of two steps is carried out. In the first step, the crossover operation that recombines the bits (genes) of each two selected strings (chromosomes) is executed. Various types of crossover operators are found in the literature. The single point and two point crossover operations are illustrated

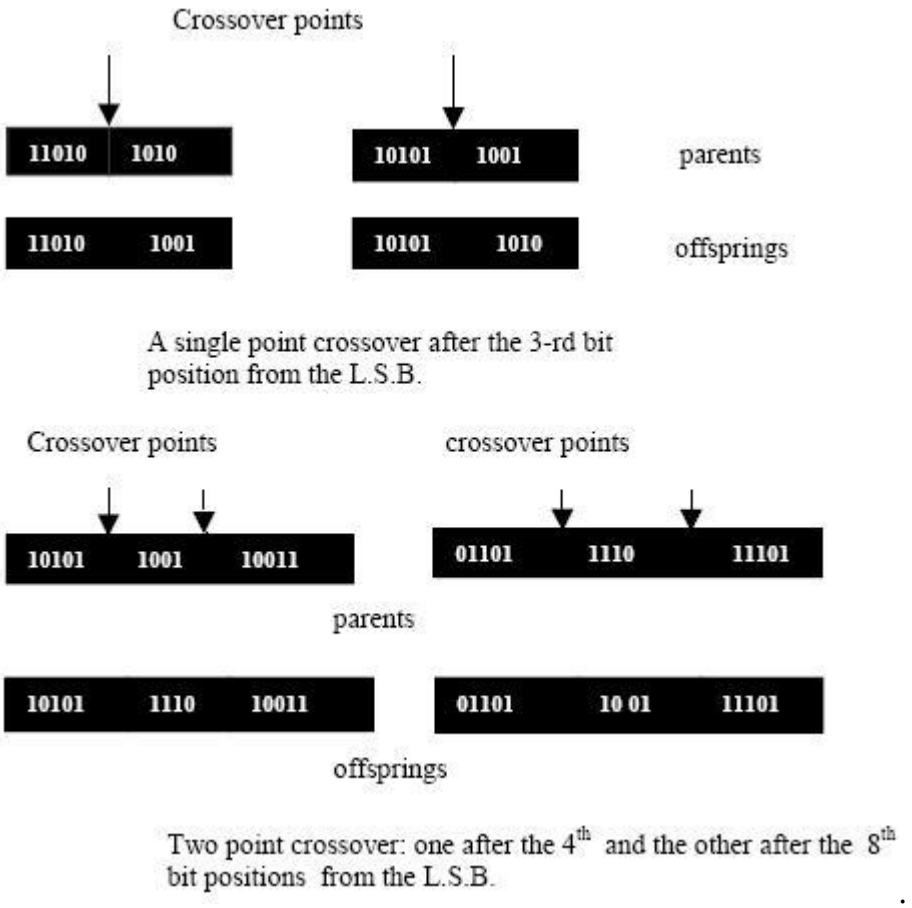

Fig. 9: One & Two Point Crossover

The crossover points of any two chromosomes are selected randomly. The second step in the genetic manipulation process is termed mutation, where the bits at one or more randomly selected positions of the chromosomes are altered. The mutation process helps to overcome trapping at local maxima. The offsprings produced by the genetic manipulation process are the next population to be evaluated. New genetic material can be introduced into the population through mutation. This increases the diversity in the population and unlike crossover, randomly redirects the search

procedure into new areas of the solution space which may or may not be beneficial. This action underpins the genetic algorithm's ability to find novel or inconspicuous solutions and to avoid getting anchored at local sub-optimal solutions. Mutation occurs by randomly selecting a particular element in a particular vector. If the element is a one it is mutated or switched to zero. One of the more common modifications for avoiding premature convergence is the introduction of a form of elitism. This technique is used in order to ensure that candidates with a high measure of performance are not replaced by relatively poorer candidates during the operations of crossover and mutation. In the worst case the best candidate may be lost during either of these operations. Elitism guarantees that the best solution represented by the highest performing candidate can never be lost. Thus the performance of the best candidate is strictly non-decreasing over successive iterations. However, if the number of elite candidates is set too high, then this could lead to deterioration in the exploration of the search space. In the following graph, we present an example of the mutation process.

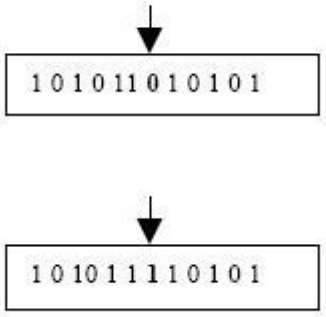

Fig. 10: Mutation of a chromosome at the 5th bit position.

Building Block Hypothesis

Genetic algorithms are simple to implement, but their behavior is difficult to understand. In particular it is difficult to understand why these algorithms frequently succeed at generating solutions of high fitness when applied to practical problems. The building block hypothesis (BBH) consists of:

1. A description of a heuristic that performs adaptation by identifying and recombining "building blocks", i.e. low order, low defining-length schemata with above average fitness.
2. A hypothesis that a genetic algorithm performs adaptation by implicitly and efficiently implementing this heuristic.

Short, low order, and highly fit schemata are sampled, recombined [crossed over], and resampled to form strings of potentially higher fitness. In a way, by working with these particular schemata [the building blocks], we have reduced the complexity of our problem. Instead of building high-performance strings by trying every conceivable combination, we construct better and better strings from the best partial solutions of past samplings. Because highly fit schemata of low defining length and low order play such an important role in the action of genetic algorithms, they are frequently called as building blocks. The building block hypothesis has been criticized on the grounds that it lacks theoretical justification, and experimental results have been published that draw the veracity of this hypothesis into question.

Important topics

Selection is clearly an important genetic operator, but opinion is divided over the importance of crossover versus mutation. Some argue that crossover is the most important, while mutation is only necessary to ensure that potential solutions are not lost. Others argue, that crossover in a largely uniform population only serves to propagate innovations originally found by mutation, and in a non-uniform population crossover is nearly always equivalent to a very large mutation (which is likely to be catastrophic). As with all current machine learning problems it is worth tuning the parameters such as mutation probability, crossover probability and population size to find reasonable settings for the problem class being worked on. A very small mutation rate may lead to genetic drift (which is non-ergodic in nature). A recombination rate that is too high may lead to premature convergence of the genetic algorithm. A mutation rate that is too high may lead to loss of good solutions unless there is elitist selection. There are theoretical but not yet practical upper and lower bounds for these parameters that can help guide selection. Often, GAs can rapidly locate good solutions, even for large search spaces. The same is of course also true for evolution strategies and evolutionary programming.

Criticism

Repeated fitness function evaluation for complex problems is often the most prohibitive and limiting segment of artificial evolutionary algorithms. Finding the optimal solution to complex high dimensional, multimodal problems often requires very expensive fitness function evaluations. In real world problems such as structural optimization problems, one single function evaluation may require several hours or even several days of complete simulation. Typical optimization methods can not deal with such types of problem. In this case, it may be necessary to forgo an exact evaluation and use an approximated fitness that is computationally efficient. It is apparent that amalgamation of

approximate models may be one of the most promising approaches to convincingly use GA to solve complex real life problems. In many problems, GAs may have a tendency to converge towards local optima or even arbitrary points rather than the global optimum of the problem. This problem may be alleviated by using a different fitness function, increasing the rate of mutation, or by using selection techniques that maintain a diverse population of solutions, although the No Free Lunch theorem proves that there is no general solution to this problem. A common technique to maintain diversity is to impose a "niche penalty", wherein, any group of individuals of sufficient similarity (niche radius) have a penalty added, which will reduce the representation of that group in subsequent generations, permitting other (less similar) individuals to be maintained in the population.

Operating on dynamic data sets is difficult, as genomes begin to converge early towards solutions which may no longer be valid for later data. Several methods have been proposed to remedy this by increasing genetic diversity somehow and preventing early convergence, either by increasing the probability of mutation when the solution quality drops (called triggered hyper mutation), or by occasionally introducing entirely new, randomly generated elements into the gene pool (called random immigrants). GAs cannot effectively solve problems in which the only fitness measure is a single right/wrong measure (like decision problems), as there is no way to converge on the solution (no hill to climb). In these cases, a random search may find a solution as quickly as a GA. However, if the situation allows the success/failure trial to be repeated giving (possibly) different results, then the ratio of successes to failures provides a suitable fitness measure. For specific optimization problems and problem instances, other optimization algorithms may find better solutions than genetic algorithms (given the same amount of computation time). The simplest algorithm represents each chromosome as a bit string. Typically, numeric parameters can be represented by integers, though it is possible to use floating point representations. The floating point representation is natural to evolution strategies and evolutionary programming. The notion of real-valued genetic algorithms has been offered but is really a misnomer because it does not really represent the building block theory that was proposed by Holland in the 1970s. Theoretically, the smaller the alphabet, the better the performance, but paradoxically, good results have been obtained from using real-valued chromosomes. A very successful (slight) variant of the general process of constructing a new population is to allow some of the better organisms from the current generation to carry over to the next, unaltered. This strategy is known as elitist selection.

It can be quite effective to combine GA with other optimization methods. GA tends to be quite good at finding generally good global solutions, but quite inefficient at finding the last few mutations to find the absolute optimum. Other techniques (such as simple hill climbing) are quite efficient at

finding absolute optimum in a limited region. Alternating GA and hill climbing can improve the efficiency of GA while overcoming the lack of robustness of hill climbing. As a general rule of thumb genetic algorithms might be useful in problem domains that have a complex fitness landscape as mixing, i.e., mutation in combination with crossover, is designed to move the population away from local optima that a traditional hill climbing algorithm might get stuck in. The commonly used crossover operators cannot change any uniform population. Mutation alone can provide ergodicity of the overall genetic algorithm process.

2.4.3 HYBRID METHODS

The most proper introductory remark to this section is by enumerating the strengths and weaknesses of the two major machine learning methodologies, namely, the Neural Networks and the Genetic Algorithms.

Advantages and Disadvantages of Neural Networks

(+) There is no need to assume an underlying data distribution such as usually is done in statistical modeling.

(+) Neural networks are applicable to multivariate non-linear problems.

(+) The transformations of the variables are automated in the computational process.

(-) Minimizing Overfitting requires a great deal of computational effort.

(-) They may attain a local but not global optimal solution.

(-) The sample size has to be large.

Advantages and Disadvantages of Genetic Algorithms

(+) It can quickly scan a vast solution set.

(+) Bad proposals do not affect the end solution negatively as they are simply discarded.

(+) The inductive nature of the GA means that it doesn't have to know any rules of the problem - it works by its own internal rules.

(-) Evolution is inductive. in nature life does not evolve towards a good solution - it evolves away from bad circumstances. This can cause GAs at finding a suboptimal solution.

The application of genetic algorithms to neural networks has followed two separate but related paths. First, genetic algorithms have been used to find the optimal network architectures for specific tasks. Modified standard genetic operators were then used to act upon populations of these genotypes to produce successively higher fitness levels. This line of work seems to sidestep the issue of a universal functional mapping since it leaves unresolved the question of whether the model's architecture performs poorly on a given task, due to the appropriateness of the given architecture or rather the inability of the backpropagation learning rule to achieve a global solution on the given architecture. The second direction involves optimization of the neural network using genetic algorithms for search. This is the direction of most of current research. With the exception of **Davis et al. (1997)**, NN research in the finance domain has not combined NN with expert systems. **Tam and Kiang (1992)** had made an attempt to optimize the NN structure using genetic algorithms. For instance, genetic algorithms can be used to find an optimal set of rules for evaluating business risk, which can then be used as input in building an NN that serves as a coherent business risk model. **Loia et al. (2000)** employed a similar methodology to build a pavement maintenance decision-support system. **Anandarajan et al. (2001)** reported results that suggest NNs combined with genetic algorithms are more effective in classifying financially distressed firms than are traditional back-propagation models.

There are several general limitations of NNs that must be considered in order to denote the need for the development of a hybrid methodology. Among the most vital issues is the ability of NNs to generalize—i.e., perform effectively in real-world situations. By manipulating the number of neurons and the number of hidden layers, an NN model can be made to learn the underlying patterns of practically any data set. However, the fact that a model learns the underlying pattern in a data set does not necessarily imply that the model will predict or classify effectively when confronted with data it has not previously seen. By memorizing the patterns in a training sample, a model might perform very well on the training patterns but may perform poorly on testing and validation patterns. **Cogger and Fanning (1997)** illustrate this potential problem, which is often referred to as overtraining. It is essential that NN models be thoroughly tested with data that were not used in building the model before being applied in the field. Validation based only on the testing sample may not be sufficient. While the testing sample may not be used directly in building the model, it is used in deciding when to stop training the network. In response to this concern, it is highly recommended that authors validate NN models with a special validation sample that is different from the testing

sample. NN researchers should also assess their NN models on validation samples with frequency distributions that look like the business risk conditions in the real world. In situations where data are scarce, the model may be validated with multiple versions of the testing sample simulated from a single set of testing patterns in a manner consistent with **Hansen et al. (1992)**.

One of the more troubling general limitations of NN models is that they do not currently allow researchers to assess the statistical significance of variables used in the model and it is difficult to explain the model conceptually. While models such as linear regression produce a set of coefficients that could be tested to draw inferences, NN models do not produce information that may be used for drawing inferences and assessing the statistical significance of input variables. An NN model is like a black box. The input and output are observable, but the internal processes used to link the input to the output are not. The classifications or predictions made by an NN model may, therefore, be difficult to explain and justify. Nonetheless, once an NN model is trained and tested, the model can be easily applied to make classifications or predictions from new data. Meticulous documentation of the processes used in building an NN model as well as careful sensitivity analysis is imperative given the inherent difficulty of explaining and conceptualizing the output of NN models.

Most supervised ANN applications use the BP algorithm. The BP algorithm has several limitations. One limitation is that learning time tends to be slow during neural network training. The learning time increases with the increase in the size of the data se. A second limitation occurs in the degree of difficulty in the training data itself. A few researchers have attempted to accelerate the learning that takes place with BP. One study included using variations in the learning rate (and corresponding step size) to decrease the learning time. Another study used various second-order techniques, which use a second derivative in the optimization process to utilize information related not only to the slope of the objective function but also its curvature. A few studies used least-squares optimization techniques. All of these techniques have offered improvements over the basic BP method. The degree of difficulty in training data has also been studied in literature. One study introduced an induction method called feature construction to help increase the accuracy of classification and to improve the learning time of an ANN. Feature construction is a different way of representing the training data prior to input to the neural network. Instead of using raw data as training data, higher-level characteristics or features are constructed from the raw data. These features are then input into an ANN. For example, if the purpose of an ANN application is to determine the financial risk of a corporation, then instead of using raw accounting data features of liquidity, profitability and cash flow could be used for more efficient learning. To construct features,

a feature construction algorithm called FC was used. The FC uses original data and builds new representations of the original data. Training data that are difficult to learn exhibit a high degree of dispersion. Feature construction can lead to a reduced degree of dispersion in the search space in which learning occurs.

Recently, a few researchers have used the principles of evolution to train ANN. Specifically, GA and hybrid architectures were used to learn the connection weights of an ANN. Among the advantages using genetic search hybrid architecture were:

1. Global-search approach that is less likely to get stuck in local optima,

2. Higher probability of convergence (as a result of global search) to a global optimum,

3. Heuristic global near-optimum (obtained by genetic search) solution is improved by local gradient-descent algorithm (BP) to obtain the true global optimal solution, and

4. Parallel genetic search approach offers several potential solutions for holdout sample.

5. The genetically evolved connection weights mitigate the well-known limitations of the gradient descent algorithm.

Genetic Algorithms are general-purpose evolutionary algorithms that can be used for optimization. When compared to traditional optimization methods, a GA provides heuristic near-optimal solutions. A GA uses a parallel (alternative chromosomes-solutions are evaluated simultaneously) search approach for locating the optimal solution. In a GA, each population member is a potential solution. At any given time, there are several potential solutions (equal to the population size). For every learning generation, several of the potential solutions are improved. The improvement in several potential solutions in one learning generation translates into a parallel search for heuristic near-optimum solution. The gradient-descent approaches use only one solution and sequentially improve it over time. The global and parallel nature of genetic search makes finding heuristic optimal solutions efficient when compared to the traditional local-search-based hill climbing and gradient-descent optimization approaches such as BP. For complex search spaces, a problem that is easy for GA may be extremely difficult for steepest ascent optimization approaches.

Genetic Algorithms are useful in their global-search capability, but do not perform as well after a heuristic global solution is found. In his study **Pendharkar (2007)** used GAs to perform a global search and obtain a heuristic near-optimal solution. After a heuristic solution was obtained,

Pendharkar used the BP algorithm to improve the heuristic near-optimal solution and possibly obtain the global optimum solution. A special selection procedure was proposed to select an appropriate solution vector for holdout sample so that the predictive accuracy of ANN can be improved.

BP and GA represent two different approaches for optimization. GA uses heuristic population-based search procedure that incorporates random variation and selection. GA and BP are different in several ways, for example, GA uses multi-point (population) search strategy and BP uses a single-point search strategy. The multi-point search strategy works in favor of GA as it increases the probability to escape from local optimum and obtain the desired global optimum. BP is a gradient-descent method. A few researchers have argued that gradient-descent procedures such as BP and others appear as single-point search strategies in that they require information from many different directions to be able to calculate the gradient. In a lack of the knowledge of an objective function, the gradient information can be obtained by several independent test trials along each of the weight axis. **Nissen and Propach (1998)** compared two population-based optimization techniques (GA and an evolution strategy) with two-point-based methods called pattern search and the modern threshold accepting (TA). Based on their experiments on a suite of six well-known functions of different dimensionality and characteristics, they report that population-based optimization approaches outperform point-based heuristic global-search approaches. In addition, population-based approaches were found to be more robust to the addition of objective function noise than point-based approaches.

Sexton, Alidaee, Dorsey and Johnson (1998a) pointed out that the gradient descent algorithm may perform poorly even on simple problems when predicting the holdout data. Their indication stems from the fact that backpropagation is a local search algorithm and may tend to fall into a local minimum. **Sexton, Dorsey and Johnson (1998b)** indicated that the use of the momentum, restarting training at many random points, restructuring the network architecture, and applying significant constraints to the permissible forms can fix it. They also suggested that one of the most promising directions is using global search algorithms to search the weight vector of the network instead of local search algorithms including the gradient descent algorithm. They employed GA to search the weight vector of ANN. The results showed that the GA-derived solution was superior to the corresponding backpropagation solution. **Gupta and Sexton (1999) and Ignizio and Soltys (1996)** also suggested that the GA-derived solutions are better than the gradient descent algorithm derived solutions. Some ANN research advocated that other global search algorithms can improve performance. In another paper, **Sexton, Dorsey and Johnson (1999)** again incorporated simulated annealing, one of global search algorithms, to optimize the network. They compared GA to

the simulated annealing. They concluded that GA outperformed simulated annealing. On the other hand, **Shin, Shin and Han (1998)** concluded that backpropagation with the gradient descent algorithm outperform ANN with GA in their application on bankruptcy prediction. They concluded that GA solution cannot always guarantee better performance than ANN trained with the gradient descent algorithm.

3. LITERATURE REVIEW

Beaver (1966) & **Altman (1968)** were the pioneers of the financial distress empirical approach. Beaver was one of the first researchers to study the prediction of bankruptcy using financial statement data. However, his analysis was very simple in that it was based on studying one financial ratio at a time and on the development of single cut-off threshold for each ratio. He also applied t-tests to evaluate the importance of individual accounting ratios within a similar pair-matched sample. The approach by Altman was about using linear models that classify between distress and non-distress firms using financial ratios as inputs. Altman used the classical multivariate Discriminant analysis technique (MDA) in a matched sample. Both the MDA model and the linear regression model (LR) have been widely used in practice and in many academic studies. They have been standard benchmarks for the financial distress prediction problem. The main criticism of the MDA was the restrictive statistical requirement posed by the model. Binary models such as Probit, Tobit and Logit, were able to overcome the main problems of MDA.

D. Whitley, Starkweather, Bogart (1990) provide an overview of several different experiments applying genetic algorithms to neural network problems. These problems include optimizing the weighted connections in feed-forward neural networks using both binary and real-valued representations, and using a genetic algorithm to discover novel architectures in the form of connectivity patterns for neural networks that learn using error propagation. In their paper the authors review work applying genetic algorithms to neural network connection weight optimization for several test problems and two real world applications, and they also review the foundations of genetic algorithms and research into theoretical problems related to the use of genetic algorithms for neural network weight optimization, and finally shows how genetic algorithms can be used to find interesting connectivities for small neural network problems. In a standard genetic algorithm the entire population undergoes reproduction in a single generation with offspring displacing parents. In

the algorithm used here, which they refer to as the GENITOR algorithm, two parents are first selected from the population. Copies of the parents are then recombined, producing two offsprings. One of the offsprings is randomly discarded and the other is then allowed to replace the lowest ranking string in the population – the offspring does not replace a parent string. This new string is then ranked according to its performance relative to the remainder of the population, so that it may now compete for reproductive opportunities. GENITOR appears to yield superior results when compared to a standard genetic algorithm when optimizing small binary encoded problems. They have also found that a distributed genetic algorithm can be used to speed up search while at the same time increasing the consistency of the genetic algorithm at finding precise solutions. The authors show that by occasionally swapping individuals between the subpopulations two complementary effects are achieved. First, the effects of genetic drift in the various subpopulations are countered. Second, the variability that results globally between the subpopulations is actually used at a local level as a source for new, yet high quality genetic material that allows diversity to be sustained in a way that constructively contributes to the search process. The other major problem which they address is the use of genetic algorithms to define the connectivity of a neural network. The approach they have used is to define a "maximally connected" feed-forward network and then use the genetic algorithm to discover a combination of connections that enhances learning ability. This means that the approach is a kind of pruning algorithm, but it is distinct from other kinds of pruning methods in that connections are not simply removed. The space of "possible connectivities" is explored with connections being both removed and reintroduced. Their results show that it is possible to find connectivities which actually enhance a network's ability to learn.

Shaw and Gentry (1990) applied inductive learning methods to risk classification applications and found that inductive learning's classification performance was better than probit or logit analysis. They have concluded that this result can be attributed to the fact that inductive learning is free from parametric and structural assumptions that underlie statistical methods.

Odom and Sharda (1990) had made use of a model with five input variables the same as the five financial ratios used in Altman's study, and one hidden layer with five nodes and one node for the output layer. They took a research sample of 65 bankrupt firms between 1975 and 1982, and 64 non-bankrupt firms, overall 129 firms. Among these 74 firms (38 bankrupt and 36 non-bankrupt firms) were used to form the training set, while the remaining 55 firms (27 bankrupt and 28 non-bankrupt firms) were used to make holdout sample. An MDA was conducted on the same training set as a benchmark. As a result, NNs correctly classified 81.81% of the hold out sample while MDA only achieved 74.28%.

Chung and Tam (1992) compared the performance of two inductive learning algorithms (ID3 and AQ) and NNs using two measures; the predictive accuracy and the representation capability. Results generated by the ID3 and AQ are more explainable yet they have less predictive accuracy than NNs. The predictive accuracy of ID3 and AQ is 79.5% while that of NN is 85.3%.

Tam and Kiang (1992) compared an NN models' performance with a linear Discriminant model, a logit model, the ID3 algorithm, and the k-nearest neighbor (k-NN) approach using the commercial bank failure data. The bank data were collected for the period between 1985 and 1987 and consisted of 59 failed and 59 non-failed. Among the evaluated models, NNs showed more accurate and robust results.

Fletcher and Goss (1993) compared an NN's performance with a logit regression model. Their data were drawn from an earlier study and were limited to 36 bankrupt and non-bankrupt firms. Their model used three financial variables, and because of the very small sample size, they used a variation of the 18-fold cross-validation analysis. The NN models had higher prediction rates than the logit regression model for almost all risk index cutoff values.

Telia La~Inen, Kaisa Sere (1996) are focusing on three alternative techniques-linear Discriminant analysis, logit analysis and genetic algorithms that can be used to empirically select predictors for neural networks in failure prediction. The selected techniques all have deterrent assumptions about the relationships between the independent variables, Linear discriminant analysis is based on linear combination of independent variables, logit analysis uses the logistical cumulative function and genetic algorithms is a global search procedure based on the mechanics of natural selection and natural genetics. In an empirical test all three selection methods chose different bankruptcy prediction variables. The best prediction results were achieved when using genetic algorithms.

Han, Jo, and Shin (1997) also compared an NN models' performance with an MDA, a logit model, and suggested the post-model integration method by finding an optimal combinational weight of the individual models' outputs. The sample data used in this experiment was composed of 1274 medium and small sized manufacturing companies that went bankrupt during the period between 1993 and 1995 and the same number of the non-bankrupt companies. Among the total of 2548 companies, 2293 companies are used for training and 255 are used for validation set. Among the models, the integrated model had the highest level of accuracies (79.1%) in the given data sets, followed by an NN model (78.7%), and a logit (76.5%).

Messier and Hansen (1988) extracted bankruptcy rules using a rule induction algorithm that classifies objects into specific groups based on observed characteristics ratios. They drew their data from two prior studies and began with 18 ratios. Their algorithm developed a bankruptcy prediction rule that employed five of these ratios. This method was able to correctly classify 87.5% of the holdout data set.

Randall S. Sexton Robert E. Dorsey and John D. Johnson (1998) demonstrate that such constraints and restructuring are unnecessary if a sufficiently complex initial architecture and an appropriate global search algorithm is used. We further show that the genetic algorithm cannot only serve as a global search algorithm but by appropriately defining the objective function it can simultaneously achieve a parsimonious architecture. The value of using the genetic algorithm over backpropagation for neural network optimization is illustrated through a Monte Carlo study which compares each algorithm on in-sample, interpolation, and extrapolation data for seven test functions. In all seven problems the best test size for BP was one. Also, for these problems the Logicon algorithm generated sizably inferior estimates. The only factor that appeared to be problem specific was the learning rate. Although there was not a lot of variation in the GA performance there was a significant amount in the BP solutions. This wide variability supports the need for the researcher to use many different configurations and a variety of starting values when using BP for optimizing a NN. Even though the NN's trained with the GA had far fewer epochs, each of the best solutions for all seven problems was superior to the corresponding best BP solution. These results demonstrate the BP tendency to converge to a local solution. The authors demonstrated that local solutions often perform poorly on even simple problems when forecasting out of sample. They have attempted to address this problem with ad hoc procedures such as stopping optimization at suboptimal solutions (not over training) or adjusting the neural network architecture to make optimization easier. They then demonstrated that if a global optimization algorithm is used, these arbitrary procedures are not necessary. Further they showed that by using a global search algorithm such as the genetic algorithm, the objective function can be set to balance the trade-off between the over parameterization of the model that may over fit the data and a parsimonious ANN that can provide a more robust solution.

R. OG stermark (1999) make use of a Neuro-Genetic Algorithm conducting Heteroskedastic Time-Series Processes Empirical Tests on Global Asset Returns. Fairly extensive comparisons between alternative ARCH-models and a prespecified heteroskedastic neural network are carried out on the European factor of a representative global asset return database.The neuro-genetic network is able to annihilate all autoregressive and heteroskedastic dependence of a difficult time series process. The holdout behavior of the neuro-genetic models is similar to the high-order ARCH presented in the

study. This result provides further evidence on the flexibility and power of the genetic hybrid framework as a solver of complicated econometric and mathematical estimation/programming problems. Their study also suggests several interesting paths for future research. Firstly, the structure for the network d the width and the number of layers d is a difficult issue that usually is unknown in estimation problems involving complicated time-series processes. The genetic algorithm could be adapted to the search for a suitable structure from the observed data. Secondly, the heavy computations usually involved in Neuro-genetic computation might be effected by means of parallel processing. This would, for example, allow entertaining multiple neural networks simultaneously in the mutation stage for a set of (randomly) selected offsprings.

Nii O. Attoh-Okine (1999) uses in this paper, real pavement condition and traffic data and specific architecture are used to investigate the effect of learning rate and momentum term on back-propagation algorithm neural network trained to predict flexible pavement performance. On the basis of the analysis it is concluded that an extremely low learning rate around 0.001–0.005 combination and momentum term between 0.5–0.9 do not give satisfactory results for the specific data set and the architecture used. It is also established that the learning rate and momentum term, and validation data can be used to identify when over-learning is taking place in a training set. The contribution of the learning rate and the momentum term in backpropagation neural network algorithm for pavement performance prediction was analyzed using actual pavement conditions and traffic data and specific architecture. The learning rate and the momentum term are very important in identifying over-learning and when to stop training. Further, it appears that a very small learning rate, roughly 0.001 and relatively high momentum term between 0.5–0.9 do not provide an appropriate combination for a three layered network for pavement performance prediction. It appears that a learning rate of around 0.2 to 0.5 and momentum term of around 0.4-0.5 seem to provide the appropriate combination for the pavement performance prediction. Discussions on the sizes of training sets as a function of the network structure and effect on learning rate and momentum terms need further research so that a more generalized result can be proposed.

Zurada et al. (1999) explain some of the conflicting results of prior studies. They argued that NNs are not superior to logit when modeling bankruptcy with a traditional dichotomous response variable. However, NN models are superior to logit models when an ordinal scale (0=healthy, 1=dividend cut, 2=loan default or favorable debt accommodation, 3=bankruptcy) is used to represent financial distress. However, an examination of these authors' results indicates that much of the reported superiority of their NN model resided in its accuracy in classifying healthy companies. In all

cases, the NN model performed worse than the regression model in classifying bankrupt firms (response category 3 in the authors' multicategory ordinal scale).

Guoqiang Zhang, Michael Y. Hu, B. Eddy Patuwo, and Daniel C. Indro (1999) make use of ANNs to study the relationship between the likelihood of bankruptcy and the relevant financial ratios. They employ a fivefold cross-validation approach to investigate the robustness of the neural networks in bankruptcy prediction. A sample of manufacturing firms that have filed for bankruptcy from 1980 through 1991 is selected from the pool of publicly traded firms in the United States on New York, American and NASDAQ exchanges. These cutoff dates for the 12 year sample period ensure that the provisions of the 1978 Bankruptcy Reform Act have been fully implemented and that the disposition of all bankrupt firms in the sample can be established by the 1994 year end. Two cross-validation schemes will be implemented. First, as in most neural networks classification problems, arc weights from the training sample will be applied to patterns in the test sample. In this study, a fivefold cross-validation is used. They split the total sample into five equal and mutually exclusive portions. Training will be conducted on any four of the five portions. Testing will then be performed on the remaining part. As a result, five overlapping training samples are constructed and testing is also performed five times. The average test classification rate over all five partitions is a good indicator for the out-of-sample performance of a classifier. Second, to have a better picture of the predictive capability of the classifier for the unknown population, we also test each case using the whole data set. The idea behind this scheme is that the total sample should be more representative of the population than a small test set which is only one fifth of the whole data set. In addition, when the whole data set is employed as the test sample, sampling variation in the testing environment is completely eliminated since the same sample is tested five different times. The variability across five test results reflects only the effect of training samples. The results from neural networks are compared to those of logistic regression. They choose this technique because it has been shown that the logistic regression is often preferred over Discriminant analysis in practice. Furthermore, the statistical properties of logistic regression are well understood. Since logistic regression is a special case of the neural network without hidden nodes, it is expected in theory that ANNs will produce more accurate estimates than logistic regression particularly in the training sample.

West, **(2000)** in his research comparing the results of five models of neural networks: multilayer perceptron (MLP), mixture-of-experts (MOE), radial basis function (RBF), learning vector quantization (LVQ) and fuzzy adaptive resonance (FAR), to assess the forecasting accuracy of credit risk. The sample included data from two countries: Germany with 700 creditworthy loan applications and 300 non-solvent and 24 variables and Australia with 307 solvent, 383 non-trustworthy

applications and 14 variables. The neural network models have made and tested ten repetitions with 10-fold cross validation. These results were compared with results of traditional credit risk assessment methods such as linear Discriminant analysis (LDA), the logistic regression (LR), the k-nearest neighbors, the kernel density estimation and decision trees / classification (CART). The key variables used in this study were the credit history, the kind of professional activities, the asset structure, the overall debt burden, the loan amount and purpose of credit. The results of this research suggest that neural network credit scoring models can achieve fractional improvements in credit scoring accuracy ranging from 0.5 up to 3%. The use of neural network credit scoring models, however, will require some modeling skills to develop network topologies and devise superior training methods. While the multilayer perceptron is the most commonly used neural network model, the mixture-of experts and radial basis function neural networks should be considered for credit scoring applications. The mixture-of-experts neural network is slightly more accurate than the other credit scoring models for the two data sets investigated in this research. A possible source of advantage for the MOE is the ability to partition the input space so that network training converges closer to the global minimum in the error surface. This research also suggests that logistic regression is a good alternative to the neural models. Logistic regression is slightly more accurate than the neural network models for the average case, which includes some inferior neural network training iterations. It is also clear that logistic regression should be the choice of the classical parametric models. Logistic regression is 0.02–0.03 more accurate than linear Discriminant analysis for the two data sets investigated.

Randall S. Sexton and Jatinder N. D. Gupta (2000), using five chaotic time series functions, compare a genetic algorithm with backpropagation for training NNs. The chaotic series are interesting because of their similarity to economic and financial series found in financial markets. Using the time series functions taken from chaotic time series literature coupled with the genetic adaptive operations developed by Dorsey and Mayer, this paper describes and directly compares the GA with the standard Norm-Cum-Delta BP algorithm (under the batch mode) on effectiveness, ease-of-use, and efficiency for training NNs. The inherent complexity of estimating the chaotic time series functional values makes them an appropriate choice for this comparison. Further, while several variations of BP and other search techniques, like radial basis function networks, could have been used for this comparison, this paper intends to show that the GA can significantly outperform the standard BP, leaving other comparisons for future research. For the in-sample and out-of-sample estimates, the GA found solutions that are better than BP even though, unlike BP, the GA was limited to the number of epochs trained for converging upon a solution. It should be mentioned that the worst

GA-trained network for all problems and architectures was found to perform better than the best BP-trained network. In order to give BP an additional opportunity to find better solutions, two additional BP runs were conducted using the same methodology as before, but included 8 and 10 hidden node architectures. As before, each of these runs included 10 replications for all BP parameter configurations, totaling 1000 replications for each problem and architecture. Out of these 10,000 additional replications, no solution was found that outperformed the GA's original results. . In every problem instance, the GA algorithm was found to provide statistically superior solutions. Since the GA had predefined parameter settings, its ease-of-use was much greater than BP. This algorithm also found these superior solutions in less CPU time, which is a major concern in NN research. Although, BP is by far the most popular method for training NNs, it is apparent from this research that a global search technique, like the GA, may be more suitable for solving this type of nonlinear problem and should be further evaluated in future research.

Kyoung-jae Kim and Ingoo Han (2000), propose a genetic algorithms (GAs) approach to feature discretization and the determination of connection weights for artificial neural networks (ANNs) to predict the stock price index. In this study, GA is employed not only to improve the learning algorithm, but also to reduce the complexity in feature space. GA optimizes simultaneously the connection weights between layers and the thresholds for feature discretization. The genetically evolved weights mitigate the well-known limitations of the gradient descent algorithm. In addition, globally searched feature discretization reduces the dimensionality of the feature space and eliminates irrelevant factors. Experimental results show that GA approach to the feature discretization model outperforms the other two conventional models. This study proposes GA approach to feature discretization (GAFD) for ANN. In this study, GA supports the simultaneous optimization of connection weights and feature discretization. GAFD takes into consideration the dependent feature by fitness function in GA. GA iterates the evolution of the populations to maximize the fitness function. GAFD simultaneously discretizes all features into the intervals at the exact thresholds. In addition, GAFD determines the maximal number of thresholds automatically. GAFD is classified as an exogenous, global, hard, and non-parameterized discretization method. GAFD may find optimal or near-optimal thresholds of discretization for maximum predictive performance because GA searches the optimal or near-optimal parameters to maximize the fitness function.

Thomas G. Calderon, John J. Cheh1 (2002) focus on the use of neural networks (NNs) as an enabler of the new business risk auditing framework and give an insight into future research opportunities. The paper reviews several published studies, which are grouped into six categories—

preliminary information risk assessment (1 study), control risk assessment (2 studies), errors and fraud (6 studies), going-concern audit opinion (3 studies), financial distress (3 studies), and bankruptcy (12 studies). The paper includes a brief introduction to NNs, followed by a description and analysis of the methods employed by and findings of researchers who used NNs as a tool for research in the auditing and risk assessment domain. The literature review leads to discussion of several broad foci areas that need further exploration in order to gain a better understanding of the efficacy of NNs as an enabler of business risk-based auditing. A discussion of the general limitations of NNs as an auditing and risk assessment tool and an outline of implications for future research opportunities conclude the paper.

Doumpos, Kosmidou, Baourakis, Zopounidis (2002) explore the prospects and implement a new Discriminant method based on Multicriteria decision making (MCDA), in the assessment credit risk. The data was taken from the loan portfolio of the Commercial Bank of Greece, including 1.411 companies for the period 1994-1997. These data establish two sets: the training sample consisting of 200 companies-100 of high and 100 of low credit risk and the testing sample consisting of the remaining 1.211 with 1093 companies of high and 118 of low risk. According to available data of these companies, 11 financial ratios were used to assess credit risk. Summarizing the results, for the same sample of companies during the examined period showed that the Discriminant analysis and Probit Analysis had a higher classification error both in the training and the testing sample, compared with MHDIS Multicriteria Approach.

Kyung-Shik Shin, Yong-Joo Lee (2002) propose a genetic algorithms (GAs) approach in this study and illustrate how GAs can be applied to bankruptcy prediction modeling. An advantage of their approach is by using GAs is that it is capable of extracting rules that are easy to understand for users like expert systems. The preliminary results show that rule extraction approach using GAs for bankruptcy prediction modeling is promising. Their research pertains to a bankruptcy prediction modeling which can provide a basis for credit rating system. Their data set contains 528 externally audited mid-sized manufacturing firms, 264 of which filed for bankruptcy and the other 264 for non-bankruptcy during the period 1995–1997. We apply two stages of input-variable-selection process. In the first stage, we select 55 variables by factor analysis, independent-samples t-test (between input variable and output variable) and Mann-Whitney U test (for qualitative variables). In the second stage, we select nine financial variables using the stepwise methods to reduce the dimensionality. The aim of input-variable-selection approach is to select the input variables satisfying the univariate test first, and then select significant variables by stepwise method for refinement. The final conclusions of the authors are the following:

1. Although they derived multiple rules using traditional GAs, it is necessary to extend the GAs through use of a niching method. Unlike the traditional GAs, which makes the population eventually converge around a single point in the solution space, the GA that uses a niching method converges about multiple solutions or niches.

2. The current rule structure is quite limited. As a next research step, this structure will be considerably extended by incorporating additional features. It is likely that more informative features will possibly lead to improved results, although we should consider the efficiency problem.

3. Further improvements may be obtained by incorporating qualitative factors as well as quantitative ones. In our next research, we plan to include qualitative variables in extracting the prediction rules.

Nayer Wanas, Gasser Auda, Mohamed S. Kamel, and Fakhreddine Karray (2002) have shown empirically that the best performance of a neural network is when the number of hidden nodes is log (T), where T is the number of training samples. This value represents the optimal considering both performance of the neural network and the computational cost associated with it**.** In this study they show, empirically, that the best performance of a neural network occurs when the number of hidden nodes is equal to log (T), where Tis the number of training samples. This value represents the optimal performance of the neural network as well as the optimal associated computational cost. We also show that the measure of entropy in the hidden layer not only gives a good foresight to the performance of the neural network, but can be used as a criterion to optimize the neural network as well. This can be achieved by minimizing the network entropy (Le. maximizing the entropy in the hidden layer) as a means of modifying the weights of the neural network. The best number of hidden units, according to the authors depends in a complex way on, the numbers of input and output units**,** the number of training cases, the amount of noise in the targets, the complexity of the function or classification to be learned, the architecture, the type of hidden unit activation function and on the training algorithm. In most situations, there is no way to determine the best number of hidden units without training several networks and estimating the generalization error of each. If you have too few hidden units, you will get high training error and high generalization error due to under fitting and high statistical bias. If you have too many hidden units, you may get low training error but still have high generalization error due to Overfitting and high variance.

Neophytou and Molinero (2004) make use of a Multidimensional Scaling Method for predicting bankruptcy of firms. More specifically they made use of accounting data for 50 industrial

companies in the United Kingdom for the years 1988 to 1999. Through these observations they had drawn the following results:

1. Linear Discriminant analysis suffers from the problem of strong simplifying assumptions of normal distribution of ratios and constant Variance / Covariance between the non-bankrupt companies.

2. The Logistic regression but has the disadvantage that only indirectly classifies businesses as it does is to calculate probabilities of bankruptcy and not be placed directly in a business group of bankrupt or not bankrupt.

3. The multidimensional scaling method has the advantage of not making assumptions about the distribution of ratios, visualizes the main characteristics of the data allowing thus the introduction and qualitative data analysis.

4. The values of the various models generally used to predict bankruptcy and other financial applications typically do not remain stable but depend on the state economy.

5. Generally it has become clear that there is no best combination of input variables for incorporation into predictive models. The aim is usually to use the representative ratios from all the major categories

Shin and Lee (2004) explored the probability of modeling the risk of corporate bankruptcy using neural networks. Their sample consisted of 2,088 industrial small-and mid-caps for the period 1997 to 2000, 50% of which went bankrupt in the period in research and the others continued to operate. The sample included 1,670 companies for the learning sample with the remaining 418 to compose the testing sample. The first step was selecting 50 indicators among 90 different financial parameters, with the help of one-sided statistical tests based on t-tests. The second step was using three techniques for selecting the most important indicators (Discrete Multivariate Analysis, Expert Knowledge & Genetic Algorithms). The main conclusions of the study were the following:

1. The Discriminant analysis selected as the most important ratios, the Return on Assets, Common Equity to Total Capital and the Asset Turnover Ratios.

2. Expert Knowledge selected as the most important ratios, the rate of sales growth and the working capital to total assets.

3. In genetic algorithms as important ratios were selected, the growth in sales, the ratio of financial expenses to sales, the ratio of net working capital to sales and the ratio of cash flows to financial costs.

4. It was suggested that Post Model Estimation, combining the results of three neural networks can provide the maximum possible accuracy. The need to merge the results stems from the significant decrease in classification accuracy for companies which have an estimated score near the defined threshold point.

Gaganis, Pasiouras & Tzanetoulakos (2005) in their study on a sample of 984 small firms (between 1997 to 2004) in the United Kingdom evaluated the performance and quality grading of five different methodologies to distinguish between bankrupt and non bankrupt companies. More specifically, their comparison included the following results: The observed superiority of Multicriteria methodologies (MHDIS & UTADIS) in the test sample is probably due to the fact that the philosophy underlying the function assumes arranged groups making these techniques being more suitable for applications, such as in the case of bankruptcy prediction business. The methodology of Support Vector Machines achieved high accuracy for bankrupt companies. They also concluded that the performance of Stacked Models (Combinatorial) is largely affected by the algorithm that combines the various methods. Using the ROC curve the Logistic Regression and the Discriminant analysis, manifested the most promising results.

Doumpos & Pasiouras (2005) explored the possibility of developing classification models which can reproduce very accurately the credit ratings of professional rating agencies. Using the Multicriteria methodology UTADIS (analytic-synthetic approach) they investigated whether it is possible to construct models of non-binary classification. Their test sample contained two types of data. Different companies in the same period, as that of the learning sample. Companies other than those of the learning sample and placed in time, at a later period than that covered by the learning. The sample consisted of 500 British companies in the non-financial sector (100 companies selected randomly for 5 risk groups) covering the period 1999-2001. The main conclusions of the study were the following: The variables whose influence on a country's credit rating was statistically significant

was a direct indicator of liquidity, net and gross profit margin, interest coverage ratio of profits, turnover rate of creditors and leverage. The UTADIS excels among all alternative methodologies (Discriminant analysis, Logistic regression, Nearest Neighborhood Classifiers, probabilistic type Neural Networks and Artificial Neural Networks) in terms of classification accuracy for all-risk groups. The authors also denote that it is necessary to use large sample sizes when using techniques from the fields of machine learning such as neural networks. There is a prospect of achieving higher predictive accuracy when using hybrid models that combine several methods.

Lee & Chen (2005) investigated the performance of a hybrid rating system that combines the technology of artificial neural networks and the Multivariate Adaptive Regression Spline or MARS. In the first step they constructed the scoring model (which identifies the most important explanatory financial variables) which will then be deployed as input nodes to enter the neural network. The MARS technique allows modeling complex nonlinear relationships between variables while not being based on very restrictive assumptions. The sample includes financial data from commercial banks in Taipei on loan applications. More specifically, data come from 510 clients of which the 10% are cases of default. Applications include variables such as loan amount, type of physical collateral, the number of guarantors, the intended use of credit, the reputation of the borrower, etc. The empirical results demonstrate the superiority of the proposed integrated methodology compared to Discriminant analysis, logistic regression and compared with the individual use of MARS or neural networks. Finally the two authors denote the use of MARS as a tool for the design and development of the topology of neural networks, while helping to identify the significant independent variables that can be entered as inputs to the network.

Muriel Perez (2006) have retained and analyzed 30 studies in which the authors use neural networks to solve companies' classification problems (healthy and failing firms). They draw two remarks. The former tackles the choice of the selected samples and criteria of this selection. The latter deals with the choice of the variables of input. The choice of the samples is made through four main criteria: the branch of industry, the size of the company, the location and finally the ratio of sound and failing companies in the samples. Regarding the number of selected ratios, two techniques co-exist and they are actually two steps of the same process. On the one hand, the authors use the ratios stemming from the literature and their personal research (but they remain unexplained), thus creating a wide range of ratios. They subsequently proceed by selection to reduce the number of ratios and to keep the most relevant ones within the scope of their application by using traditional statistical techniques (analysis in main components, study of correlation between variables). On the

other hand, the authors also proceed to a review of the literature now only selecting the range of ratios stemming from the analysis of previous data having been set up for other applications.

Jae H. Min and **Chulwoo Jeong (2009)** propose a new binary classification method for predicting corporate failure based on genetic algorithm, and validate its prediction power through empirical analysis. Establishing virtual companies representing bankrupt companies and non-bankrupt ones, respectively, the proposed method measures the similarity between the virtual companies and the subject for prediction, and classifies the subject into either bankrupt or non-bankrupt one. The values of the classification variables of the virtual companies and the weights of the variables are determined by the proper model to maximize the hit ratio of training data set using genetic algorithm. In order to test the validity of the proposed method, they compare its prediction accuracy with those of other existing methods such as multi-Discriminant analysis, logistic regression, decision tree, and artificial neural network, and it is shown that the binary classification method proposed in this paper can serve as a promising alternative to the existing methods for bankruptcy prediction. The proposed method is somewhat similar to cluster analysis as both methods classify the subjects into several clusters. But the proposed method differentiates itself from cluster analysis in the following aspects. The method classifies the subjects into clusters to maximize the prediction accuracy, the matching rate of the subject's bankruptcy status and the representative firm's one, with prior knowledge of whether each subject is bankrupt or not, while cluster analysis does its classifying job without the prior knowledge of the status of each cluster. The classification method proposed in this paper may contribute to academia as well as practitioners in the following aspects.

1. The binary classification method in this paper can make a contribution to credit risk management of financial institutions such as banks. Ever increasing competition among banks makes the bankruptcy prediction to become their inevitable agenda for systematic credit risk management, and their ability of accurate prediction for corporate failures plays an extremely important role for them to survive in the market and sustain growth through generating profits from their lending practices. Also, at a macro level, the method is expected to contribute to economic development of a nation by helping banks to allocate their funds efficiently and effectively to the prospective firms requesting credit loans according to their respective financial soundness.

2. This study purposed to develop a new classification method as a promising alternative to existing methods for predicting corporate failures. In this respect, it served its end. In addition, the method suggested in this study has the flexibility in its application range such that it can be applied in other areas such as product purchase prediction and project risk management among others.

Chih-Fong Tsai (2009) have applied data mining and machine learning techniques to solve the bankruptcy prediction and credit scoring problems. As feature selection is an important step to select more representative data from a given dataset in data mining to improve the final prediction performance, it is unknown that which feature selection method is better. The authors aim at comparing five well-known feature selection methods used in bankruptcy prediction, which is t-test, correlation matrix, stepwise regression, principle component analysis (PCA) and factor analysis (FA) to examine their prediction performance. Multi-layer perceptron (MLP) neural networks are used as the prediction model. Five related datasets are used in order to provide a reliable conclusion. Regarding the experimental results, the t-test feature selection method outperforms the other ones by the two performance measurements. There are three stages to complete the proposed experiment The first stage is to build a multi-layer perceptron (MLP) neural network as the baseline model since it is the most widely used in bankruptcy prediction. In this stage, they do not apply any feature selection methods. The second stage uses the five feature selection methods individually for generating more appropriate features. Then, there are five different new generated feature sets which are used to train the MLP model. The third stage is to evaluate the models' performance. They consider two evaluation methods to verify these models. They are average accuracy and Type I and II errors. In order to make a reliable comparison, they have used five datasets including Australian Credit, German Credit, Japanese Credit, Bankruptcy dataset and the UC competition datasets. The five datasets belong to bankruptcies and credit evaluations match the definition of bankruptcy. They have used MLP neural networks with the back-propagation learning algorithm as the baseline prediction model. Regarding the experimental results, feature selection methods applied on selecting more representative variables certainly increase the performance of prediction. On average, t-test is superior to others and Stepwise is on the second position. For the percentage of reducing the original variables, stepwise outperforms the others, which provides the highest feature reduction rate. However, the result of using stepwise for the bankruptcy prediction and credit scoring problems is instable based on the chosen five datasets. In summary, t-test performs stably and provides higher prediction accuracy and lower Type I and II errors.

Abdou (2009) in his research compares the results of conventional techniques such as Discriminant analysis (DA), the logistic regression (LR) and the weight of evidence measure (WOE) with artificial neural networks such as the probabilistic neural network (PNN), the multilevel feed-forward networks (MLFN) and a combination of neural networks (BNS4-MLFN-5N), in terms of credit risk. The data come from a commercial bank in Egypt, and consists of 630 loan applications of which 49 were rejected. The final data set contained 433 'good' loans and 148 'bad' loans. Still, the

data were divided into 389 cases forming the training sample and 192 cases in the testing sample. The main conclusions of the study were:

1. The Discriminant analysis had the highest average rate of correct classification among conventional methods of credit risk assessment.

2. All methods provided more accurate results for the good credit group than the bad, except for the BNS4-MLFN-5N and WOE, which provided an accuracy of almost 100% for bad credit group.

3. The highest percentage of correct classification of good credit quality was achieved by logistic regression.

4. Overall in terms of average accuracy of classification and misclassification costs, Neural Networks had a clear advantage.

Chia-Ming Wang a, Yin-Fu Huang (2009) compared the relative performance of SVM with several other classifiers, such as decision stump, naive Bayes, nearest neighbor (k = 3, 5), back-propagation neural networks, C4.5decision tree, perceptron, and linear Discriminant analysis (LDA). All the performance results of classifiers were obtained through five-fold cross validation to minimize the impacts of data dependency and prevent the over-fitting problem. It was observed that the evolutionary based feature selection with new criteria outperforms other techniques in finding a large and important part of the Pareto frontier the feature selection problem.Since data pre-processing and feature selection are important steps in the knowledge discovery process, the further work will apply these techniques with other classifiers to large scale problems. Moreover, new multi-objective evolutionary algorithms should be considered.

Chih-Fong Tsai, Ming-Lun Chen (2010) study credit rating by hybrid machine learning techniques. They discriminate between clustering and classification methods. Regarding this comparative result, the 'Classification + Classification'hybrid model performs the best, which provide 83.44%prediction accuracy. In particular, logistic regression used as the first component combined with neural networks as the second component, i.e. LR + NN, is superior to the other models. Moreover, it also can maximize the profit. On the other hand, LR + NN can accurately predict the most normal and overdue accounts, which means that it provides the lowest Type I and II errors. Therefore, this hybrid model can be regarded as the optimal credit rating system. It is

interesting that the 'Classification + Clustering' and'Clustering + Clustering' hybrid models do not outperform single classification models. This implies that the clustering techniques cannot provide reasonable credit rating results. Therefore, to develop a hybrid learning credit model, there are four different ways to combine the two machine learning techniques. They are:(1) combining two classification techniques, (2) combining two clustering techniques, (3) one clustering technique combined with one classification technique, and (4) one classificationtechnique combined with one clustering technique.

Marinakis, Marinaki & Zopounidis (2010) used a nature inspired methodology, known as Honey Bees Mating Optimization (HBMO) to determine credit risk. Their sample included 1330 observations in non-financial companies of Great Britain for the period 1999-2001. The observations of the sample were classified into five risk categories according to the probability of bankruptcy. The categories were arranged in ascending probability of bankruptcy such as: Secure, Stable, Normal, Unstable and High Risk. The benchmarks were a set of 26 ratios derived from financial statements. The HBMO algorithm was used for the selection of the most important indicators in conjunction with three types of classification models, the 1-nearest, the K-nearest and WK-nearest neighbors. The results of this method were compared with those of alternative algorithms such as Particle Swarm Optimization, the Ant Colony Optimization, a Metaheuristic Genetic algorithm and a tabu search algorithm. The authors conclude that the HBMO algorithm provides the highest accuracy of classification of businesses especially when this is used in conjunction with the 1-nearest neighbor classifier. The average number of economic variables selected for inclusion in the formed classification model was 11. Without the use of an algorithm for optimizing the selection of appropriate explanatory variables in individual models, the classification accuracy was 12% below from even the lowest accuracy classification technique when used as an algorithm for selecting input features.

Fang-Mei Tseng, Yi-Chung Hub (2010) compare four bankruptcy prediction models: Logit, quadratic interval logit, neural and fuzzy neural networks. Actually, they show that a backpropagation MLP with a single hidden layer and any fixed continuous sigmoidal function is sufficient to approximate any continuous function. However, the appropriate numbers of hidden nodes and hidden layers cannot be known in advance.

In the next four tables we discuss in a more intuitive representation some of the most important empirical findings, based on the literature of the last 40 years. More specifically, in Table 1 we depict the results in regard with the comparison of the various bankruptcy prediction methodologies. In Table 2 we refer to the frequency of use of the alternative bankruptcy forecasting techniques, and finally in Tables 3 & 4 we present the financial variables used in credit scoring literature. Summarizing this work we can make the following observations.

1. Neural Networks seem to outperform the traditional bankruptcy forecasting methodologies, such as the Linear Discriminant Analysis and the Logit Analysis.

2. Most frequently used credit scoring methods are the Discriminant, Logit and the Neural Networks models.

3. Most frequently used financial ratios, used as inputs to the models are:

a. Net Income / Total Assets

b. Operating Income / Total Assets

c. Current Assets / Current Liabilities

d. Working Capital / Total Assets

e. Assets Turnover Ratio

f. Cash Flow / Total Debt

g. Equity / Debt

Author(s)	Empirical Results
Wilson & Sharda (1994)	NN dominate Discriminant Analysis and other Statistical Techniques in Bankruptcy forecasting
Fletcher and Goss (1993)	NN dominates Logistic Regression in small samples
Altman (1994)	Discriminant Analysis slightly better than NN in holdout samples. NN superior in healthy companies.
Poddig (1995)	NN with different preprocessing techniques dominates discriminant analysis in bankruptcy prediction
Kerling (1996)	NN has almost equal performance as the Statistical Techniques
Brocket (2007)	NN dominates to Discriminant Analysis as an Early Warning System for Insolvency Prediction
Boritz & Kennedy (2007)	NN has lower type I and higher type II errors than Statistical Approaches
Leshno & Spector (1996)	Prediction Capability of NN depends on Sample size and learning algorithm. Large number of Iterations can cause overfitting.
Yang et al. (1999)	DA dominates NN & PNN - Oil and Gas Companies
Zurada et al. (1999)	NN dominate LR in when an ordinally scaled variable is used to represent financial distress
Barniv et al. (1997)	NN dominates Logit & DA on the Training Sample but not in the Testing Sample
Bell (1997)	NN not significantly superior to Logit
Etheridge & Sriram (1997)	DA dominates NN in 1 year ahead forecasts
Tan (1996)	NN dominates Probit Model
Coats & Fant (1993)	NN dominates Statistical Techniques - Audit Companies
Salchenberger et al. (1992)	NN equal to Logit on a matched sample but better on a Sample with more healthy companies
Tam & Kiang (1992)	NN dominates Decision Trees and Non Parametric Techniques
Raguhapathi et al. (1996)	NN with two hidden layers achieved 99% Prediction Accuracy
Odom & Sharda (1990)	NN performed at least as well as DA

Table 1. Comparative Performance of Neural Networks and Statistical Techniques

PAPERS / METHODS	West (2000)	Doumpos, Kosmidou, Zopounidis (2002)	Neophytou & Molinero (2004)	Shin & Lee (2004)	Gaganis, Pasiouras & Tzanetoulakos (2005)	Doumpos & Pasiouras (2005)	Lee & Chen (2005)	Abdou (2009)	Marinakis, Marinaki & Zopounidis (2010)
Discriminant Analysis	X	X	X	X	X		X	X	
Logit Analysis	X		X		X	X	X	X	
Probit Analysis		X							
Multidimensional Scaling			X*						
Neural Networks	X*			X*		X	X	X*	
Genetic Algorithms				X					X*
Decision Trees	X								
Support Vector Machines					X				
UTADIS					X*	X*			
MHDIS		X*			X*				
K – Nearest Neighbors	X					X			X
MARS							X*		

Table 2. Bankruptcy prediction Techniques used in selected literature.

* The asterisk denotes which method provided superior results in the specific study.

PAPERS/ RATIOS	Sharda & Wilson (1993)	Ahn, Cho & Kim (2000)	McKee (2000)	Doumpos Kosmidou Baourakis Zopounidis (2002)	McKee & Lensberg (2002)	Baek & Cho (2003)	Bian & Mazlack (2003)	Cielen, Peeters & Vanhoof (2004)	Jones & Hensher (2004)	Andres, Landajo & Lorca (2005)	Lee, Booth & Alam (2005)	Gaganis, Pasiouras & Tzanetoulakos (2005)	Marinakis, Marinaki & Zopounidis (2010)
STL / EQ				X									
STL / TL							X						
NPM													X
IEC								X	X				
GPM				X			X						X
Assets / Employee												X	
EQ / TL	X	X				X		X	X		X		X
NI / EQ	X					X	X				X		
NI / TA	X	X	X	X	X	X	X	X	X		X	X	
NI / EMP												X	
CC										X			
RTR			X										
GTR		X	X										
ATR	X			X		X	X		X	X	X		
CA – INS / STL								X		X		X	
CA / STL		X	X	X	X		X	X				X	X
CA / S			X										
CA / TA			X					X					
TE								X					
ASSETS					X								
WE										X			
AG												X	
Sales Growth										X			
Total Assets										X			
TL / TA			X					X				X	
CF / TL		X					X		X				
CF / NI					X								
OCF / TA									X				

Abbreviations: STL (Short Term Liabilities), EQ (Equity), TL (Total Liabilities), NPM (Net Profit Margin), IEC (Interest Expense Coverage), GPM (Gross Profit Margin), NI (Net Income), TA (Total Assets), EMP (Employees), CC (Cost of Capital), RTR (Receivables Turnover Ratio), GTR (Goods Turnover Ratio), ATR (Assets Turnover Ratio), CA (Current Assets), INS (Investments), TE (Tax Expenses) CF (Cash Flow), OCF (Operating Cash Flow), AG (Assets Growth), WE (Wage Expenses), S (Sales)

Table 3. Financial ratios and their frequency of use in the empirical literature. (Part A: 1990-2010)

PAPERS/ RATIOS	Altman (1968)	Altman (1977)	Beaver (1966)	Blum (1974)	Deakin (1974)	Edminster (1972)	Fitzpatrick (1932)	Merwin (1942)	Ramser (1931)	Tafler (1982)	Winakor (1935)
C/CL						X	X				
CF / CL						X					
CF / TD			X	X	X						
C / NS					X						
C / TA					X						
CA / CL		X	X		X			X			
CA / NS					X						
CA / TA					X						
CL / EQ						X					
EQ / FA							X				
EQ / NS						X			X		
INV / NS						X					
MVE / BVD	X	X									
TD / EQ								X			
NI / TA			X		X						
QA / INV				X							
NS / TA	X								X		
OI / TA	X	X								X	
EBIT / INT		X									
QA / CL					X						
QA / NS					X						
QA / TA					X					X	
ROE				X							
RE / TA	X	X									
ROS							X			X	
TD / TA			X		X						
WC / NS					X	X					
WC / EQ										X	
WC / TA	X		X		X			X			X

Abbreviations: C (Cash), CL (Current Liabilities), CF (Cash Flow), TD (Total Debt), TA (Total Assets), NS (Net Sales), EQ (Equity), INV (Inventory), ROE (Return on Equity), ROS (Return on Stock), WC (Working Capital), OI (Operating Income), LTD (Long Term Debt), NI (Net Income), QA (Quick Assets), CA (Current Assets), EBIT (Earnings Before Interest), MVE (Market Value of Equity), FA (Fixed Assets), RE (Retained Earnings)

Table 4. Financial ratios and their frequency of use in the empirical literature. (Part B: 1930 - 1990)

4. RESEARCH DESIGN IN LITERATURE

In this chapter we will form a concrete set of rules and laws, based on a thorough examination of a large number of published papers in the field of credit risk modeling. This systematic approach is going to be our 'validated' guide in order to proceed further with our empirical work. In the following lines we divide the architecture of our empirical study in 12 discrete dimensions. In each of them we have collected and document the most important findings of other authors. These findings will work as a toolkit in order to build upon and construct our research study. This is

1. Data Normalization, Missing & Wrong Values (attributes with large numeric ranges dominate those with small numeric ranges)

Missing values: it is suggested to substitute the empty space with the arithmetical mean of the field, being the mean calculated as the mean of the existing value belonging to that field for all the businesses in the overall period of collecting.

Data Normalization: must be performed in order to feed the net with data ranging in the same interval for each input node. A suitable choice is to use the interval [0, 1] for each input node. The most common way for normalizing data is the Min–Max linear transformation to [0, 1], except in case of outliers existence. In such occasion, using the Min–Max formula a lot of useful information can be lost. In this case the logarithmic formula must be used. This formula is flexible because it can be defined by the user, provided that the argument of the formula being <1 and is used as follows: $x = \log m(x + 1)$, where m is near to the actual maximum of the field analyzed. We add 1 to the argument to avoid the value being >1.

Jae H. Min, **Chulwoo Jeong (2008):** used the means and standard deviations of the financial ratios of 2814 firms to standardize them as Z-values, and the observations (the firms) whose Z-values were beyond the range of [−3, 3] were considered outliers and excluded from the data set

Randall S. Sexton, **Robert E. Dorsey**, **and John D. Johnson (1998):** GA does not need require normalization, in contrast to BP.

Han, J. W., & Kamber, M. (2001): The normalization (in order to equalize the effects of variables of different scales) procedure (min-max) is performed before the feature selection process (that removes noisy, correlated, or abundant features, dimensionality).

2. Training and Testing Sample ratio.

No definite guidelines have been established for dividing the sample into training and holdout sample. Some researchers advocate a 60-40 split, while others prefer a 75-25 Split. The most adopted solution is 60% for training and 40% for testing.

Kyung-Shik Shin Yong-Joo Lee (2002): Training 90% and Testing 10%

Jae H. Min, **Chulwoo Jeong (2008):** Training 60% and Testing 40%

West (2000): 10 fold cross validation (90% Training – 10% Testing)

Fang-Mei Tseng a, Yi-Chung Hub (2010): 80% and 20% of the given data are randomly partitioned into a training set and a testing set, respectively.

3. Bankrupt and Non Bankrupt ratio

(Weiss & Provost, 2001): show that the naturally occurring class distribution often is not best for learning, and often substantially better performance can be obtained by using a different class distribution. They also documented that when classes are imbalanced, it would cause seriously negative effects on the classification performance and they recommend the use of a matched sample.

(Drummond & Holte 2003): Oversampling is ineffective, when we have a class imbalance.

Jain and Nag (1997): If a categorization problem is characterized by unequal frequencies of the two states of interest, then the NN model should (at least) be validated with a sample that contains a realistic frequency distribution of the output variable. This issue merits further investigation.

Zhang et al. (1999): used a matched sample

Jae H. Min, **Chulwoo Jeong (2008):** used a matched sample

Wilson and Sharda (1994): 50% Bankrupt –50% Non Bankrupt

Hsieh (2005): 70% Performing – 30% Non Performing

Baesens (2003): 70% Performing – 30% Non Performing

Kim and Sohn (2004): 70% Performing – 30% Non Performing

Rumelhart et al. (1986): 69% Performing and 31% Non Performing

4. Number of Hidden layers & Hidden Nodes (trial & error e.g. backward & forward)

Two hidden layers are needed only in high frequency data. For many practical problems there's no reason to use any more than one hidden layer. Problems that require two hidden layers are rarely encountered. Using too few (many) neurons in the hidden layers will result in something called under fitting (over fitting). Under fitting occurs when there are too few neurons in the hidden layers to adequately detect the signals in a complicated data set. Many neurons need more training time. Most neural networks can operate with a hidden layer of 2% to 30% the size of the input layer. One output node is needed for a two group classification problem. The number of input nodes is the number of predictor variables

Han, J. W., & Kamber, M. (2001): 1 HL

Cybenko (1989): has proved that only one layer of hidden units is adequate to approximate any function (universal approximation theorem)

Fang-Mei Tseng a, Yi-Chung Hub (2010): argue that too many weights, few training patterns, or many iterations should have adverse effects in the generalization capability of the NN (as a result of overtraining)

Fang-Mei Tseng a, Yi-Chung Hub (2010): Number of hidden neurons equal to ½ of the number of input attributes.

Lippman (1987): no more than three layers are required in perceptron-like feed-forward networks because a three-layer network can generate arbitrarily complex decision regions.

Pao (1989): small number of nodes and layers.

Kyoung-jae Kim and Ingoo Han (2000): Pruning in order to trim network size: remove nodes that do not seriously affect the performance of the NN, 1 HL for the majority of problems and HN equal to Neurons in Input Layer and Neurons in Output Layer / 2 or Max, twice the number of input nodes, or the number of principal components. Number of Input Nodes equal to the number of input variables.

Rumelhart et al. (1986): 1 HL and 19 HN

Zhang et al. (1999): use 1 HL and 1-15 HN

Randall S. Sexton and Jatinder N. D. Gupta (2000): 1 HL with 2, 4, 6 HN for BP and GA

Randall S. Sexton, **Robert E. Dorsey**, **and John D. Johnson (1998):** use 6 HN

Chih-Fong Tsai (2009): use 1 HL and 8-16-32-64 HN

Kusiak (2001): large number of weights helps to reduce training and generalization error and to avoid local minima

Nii O. Attoh-Okine (1999): argues that the more hidden units a NN has, the less likely is it to trap into local minima.

Hsieh (2005): recommends 1 HL. Too large number of layers and/or neurons in these layers, too long training. He defines the number of HN equal to the half of the number of inputs. Start training, add one neuron at a time, and stop when there is no further improvement in forecasting accuracy.

Hornik et al. (1989): showed that one single hidden layer is sufficient to permit approximation of any function. As regards the size of this layer, it is not determined by the theory and its construction requires empirical processes.

Funahashi (1989): a NN with a single HL can approximate any unknown function.

Jae H. Min, **Chulwoo Jeong (2008):** 2 HL and the number of hidden nodes in each layer is equal to the number of inputs.

Nelson and Illingworth (1991): four hidden units for each input unit.

5. Fitness Functions – Accuracy Measures – Measures of Error

- Determination Index ($R^{2)}$
- Mean Absolute Error (MAE)
- Mean Absolute Percentage Error (MAPE)
- Mean Square Error (MSE)
- Sum of Squared Errors (SSE)

Randall S. Sexton and Jatinder N. D. Gupta (2000): use RMSE (for comparisons of the models)

Rumelhart et al. (1986): used RMSE as objective function

Zhang et al. (1999): minimize MSE or SSE

Jae H. Min & Chulwoo Jeong (2009): use SSE as objective function

West (2000): uses the MSE as the objective function

Randall S. Sexton, **Robert E. Dorsey**, **and John D. Johnson (1998):** use MSE

$$\mathrm{MSE} = \frac{1}{N}\sum_{j-1}^{N}(a_j - y_j)^2,$$

$$\mathrm{RMSE} = \sqrt{\sum_{p-1}^{P}\sum_{j-1}^{N}(O_{pj} - t_{pj})^2 / NP},$$

6. Activation Function

There is an overlapping between sound companies and failing ones, in terms of their corresponding financial ratios. The sigmoid functions are not the only ones which have the ability to manage this nonlinearity.

Han, J. W., & Kamber, M. (2001): Logistic function

Zhang et al. (1999): use the Logistic (Sigmoid) Function

Nii O. Attoh-Okine (1999): use the sigmoid Function.

7. Training Algorithm

Han, J. W., & Kamber, M. (2001): Pattern by Pattern learning is used to immediately update the connection weights after each input output pair has been presented. In contrast to batch update (weight update after a full pass through the training set)

Randall S. Sexton, **Robert E. Dorsey**, **and John D. Johnson (1998):** GA slower than BP but with higher classification capability and leading to a more parsimonious structure. Test size: the number of input patterns, after the pass of which the weights will be updated.

Zhang et al. (1999): use the BP and 50 different initial random weights.

Nii O. Attoh-Okine (1999): argue that updating the weights in a BackProp network can be done either after the presentation of each pattern (pattern learning), or after all of the patterns in the training set have been presented (epoch learning). If the learning rate is small, there are little differences between the two procedures. The initial weights to the network should be unequal, in order to overcome the symmetry problem, that happens due to the existence of the momentum term, and causes the weights to move to same direction, without the possibility to change direction, in the weight space, during the search process.

Randall S. Sexton, **Robert E. Dorsey**, **and John D. Johnson (1998):** 10 initial random weighting schemes for the BP and 10 initial random populations for the GA (always with a different random seed)

8. Thresholds

Zhang et al. (1999): use 0.2 & 0.5

Kyung-Shik Shin Yong-Joo Lee (2002): use GA to calculate Thresholds for the variables.

West (2000): uses 0.5

9. Input Variables

Homogeneity in the sample of companies e.g. Large (Small) – Industrial (Retail) due to the different dynamics and idiosyncratic patterns, underlying their corresponding descriptive financial variables.

Altman (1968): 1. Working capital/total assets 2. Retained earnings/total assets 3. Earnings before interest and taxes/total assets 4. Market value equity/book value of total debt 5. Sales/total assets.

Zhang et al. (1999): additionally used the current assets / current liabilities ratio. He used the value of the ratios for the year or three years immediately before the filing for bankruptcy. Bankruptcies occurring after a period of 1, 2 or 3 years

Kyung-Shik Shin Yong-Joo Lee (2002) Value Added / Total Assets, Net Income / Common Equity, Quick Ratio, Liquidity Ratio, Current Liabilities / Total Assets, Retained Earnings / Total Assets, Stockholder's Equity / Total Assets, Financial Expenses / Total Assets, Operating Income / Operating Expenses.

Jae H. Min, **Chulwoo Jeong (2008):** use 27 financial ratios

10. Data Preprocessing (Feature selection - transformation – discretization – extraction)

Data Preprocessing Methods: Feature Transformation – Subset Selection

1. Feature Extraction (Discard those variables you think, that are not pertinent to the problem, use only one of two or more strongly correlated variables, use function of variables)

2. Feature Construction

3. Feature Discretization (dimensionality reduction – elimination of irrelevant features – parsimonious models - Character variables assigned the values 0 or 0.5)

3.1 Endogenous (SOM – Clustering – Decision Trees)

3.2 Exogenous (Cramer V, Entropy, K-nearest Neighbors, GA)

Nii O. Attoh-Okine (1999): used the t-test

Chih-Fong Tsai (2009): used the t-test (others Correlation Matrix – PCA – FA – GA - SOM)

Chih-Fong Tsai (2009): reducing the number of irrelevant or redundant (due to Multicollinearity) features drastically reduces the running time of a learning algorithm and yields a more general concept. Except from this, there are additional potential benefits of feature selection, which are facilitating data visualization and data understanding, reducing the measurement and storage requirements, reducing training and utilization times, complexity and learning time and possible noise, defying the curse of dimensionality to improve prediction performances. The t-test method is used to determine whether there is a significant difference between two group's means. It helps to answer the underlying question: do the two groups come from the same population, and only appear differently because of chance errors, or is there some significant difference between these two groups.

Kusiak (2001): feature transformation (SOM) differs from feature subset selection in that the latter does not generate new features and selects a subset of original features.

Kyung-Shik Shin Yong-Joo Lee (2002): t-test between input and output variables.

11. Various Validation Measures

Han, J. W., & Kamber, M. (2001): Type I (Good Credit classified as bad)**,** Type II (Bad Credit classified as good) – more important – ratio of 5 to 1 in the misclassification function)**,** either use 10fold cross validation or 10 different initial random weighting schemes.

Chih-Fong Tsai (2009): uses Type I (when we classify a non BKR as a BKR) and Type II errors (when we classify a BKR as non BKR). Statistical Significance of the difference in the accuracy of the proposed methodologies.

West (2000): Mc Nemar's chi-square tests are used for finding significant differences among the models. Cost Function. In credit scoring applications, it is generally believed that the costs of granting credit to a bad risk candidate, C_{12} is significantly greater than the cost of denying credit to a good risk candidate, C_{21}. Evaluation of the cost function also requires estimates of the prior probabilities of good credit π_1 and bad credit π_2 in the applicant pool of the credit scoring model. These prior probabilities are estimated from reported default rates. The ratio n_2/N_2 measures the false positive rate, the proportion of bad credit risks that are granted credit, while the ratio n_1/N_1 measures the false negative rate, or good credit risks denied credit by the model.

$$\text{Cost} = C_{12}\pi_2\frac{n_2}{N_2} + C_{21}\pi_1\frac{n_1}{N_1}.$$

Jae H. Min & Chulwoo Jeong (2009): McNemar Chi Square test for the detection of significant differences between the models. Wilcoxon matched pairs signed ranks test for the comparison or the results between the training and testing samples.

Jain and Nag (1997): recommend the following weighted efficiency (WE) measure that takes into consideration both error rates: WE = (OC)(PSC)(CSR) where OC is the percentage of ventures correctly classified, PSC is the percentage of successful ventures correctly classified, and CSR is the ratio of the number of ventures correctly classified as successful to total number of ventures classified as successful.

Calderon (1999): Cost Function C = $(1-q)p_1\ c_1+q\ p_2\ c_2$ where C is the total cost of misclassification, q is an a priori probability that the information risk associated with a company's financial statements is high, p_1 and p_2 are the probability of a type 1 and a type 2 error, respectively (estimated by the type 1 and type 2 error rates for the classification model), c_1 and c_2 are the cost of type 1 and type 2 errors. The expected cost of a type 2 error should be higher than the expected cost of a type 1 error.

12. Genetic Algorithm and Neural Network Parameters

- Sample Size

Kyung-Shik Shin Yong-Joo Lee (2002): use 528 companies.

Jae H. Min, **Chulwoo Jeong (2008):** use 2.542 companies

- Iterations

10.000 – 60.000 epochs are sufficient – no learning added value after that point.

Chih-Fong Tsai (2009): use 50-100-200-400 (BPN)

Jae H. Min, **Chulwoo Jeong (2008):** 20.000 (BPN)

Randall S. Sexton, **Robert E. Dorsey**' **and John D. Johnson (1998):** 20.000 epochs for the BPN & 5.000 Generations for the GA

Rumelhart et al. (1986): 3.000 Iterations

Kyung-Shik Shin Yong-Joo Lee (2002): use 3.000 trials.

- Momentum (enables that the adoption of weights in the network during the training avoiding the local minimum-it adds a proportion of the previous weight changes to the current weight changes-makes learning faster)

Fang-Mei Tseng a, Yi-Chung Hub (2010): for the back-propagation neural network, they set the maximum run, learning rate and momentum to 1000, 0.9, and 0.2, respectively.

Han, J. W., & Kamber, M. (2001): Large LR the gradient descent will oscillate**.** Large Momentum can cause instability.

Nii O. Attoh-Okine (1999): recommend momentum of around 0.4-0.5 in 1HL NN, and reduce the speed of learning (through its decrease) in the later stages of the training procedure in order to avoid overshooting.

Randall S. Sexton, **Robert E. Dorsey**' **and John D. Johnson (1998):** set to 0.9

Randall S. Sexton and Jatinder N. D. Gupta (2000): set Momentum between 0.0-0.9 (to escape local solutions – weight changes reflect past direction movements)

- Learning rate (ratio of the weights' change after the actual correction is evaluated)

Rumelhart et al. (1986): LR 0.002, 0.004 (the best), 0.006

Randall S. Sexton, **Robert E. Dorsey**' **and John D. Johnson (1998):** set to 0.5 or 1

Randall S. Sexton and Jatinder N. D. Gupta (2000): - Dynamic Learning Rate – Asymptotic Convergence is guaranteed only when a small LR is used.

Nii O. Attoh-Okine (1999): recommend 0.2-0.5 in a 1HL NN. If it is too large, there is the risk for oscillation of weights.

$\Delta w_{ji}(n+1) = \varepsilon d_{pj} a_{pi} + \alpha \Delta w_{ji}(n)$

- Cross over rate

Fang-Mei Tseng a, Yi-Chung Hub (2010): population size is 50, reproduction rate is 0.9, crossover rate is 0.9, mutation rate is 0.01, and maximum number of generations is 100 and 1000, respectively.

Kyung-Shik Shin Yong-Joo Lee (2002): define 0.5-0.7

Jae H. Min, **Chulwoo Jeong (2008):** 0.5

- Mutation rate

Kyung-Shik Shin Yong-Joo Lee (2002): 0.06-0.12

Jae H. Min, **Chulwoo Jeong (2008):** 0.1

D. Whitley, T. Starkweather and C. Bogart (1990): recommend the gradual increase in the mutation rate in order to enforce the potential of search and sustain diversity among the possible solutions.

- Population of Chromosomes

Jae H. Min & Chulwoo Jeong (2009): use 20 set of weights, Population size is 50

Jae H. Min, **Chulwoo Jeong (2008):** Each chromosome is composed of 9 genes for the weights and the number of representative firms times 9 genes for the representative firms' classification variables. The genes for the weights are set to be real numbers in the range of [0, 1], and the genes for the representative firms' variables are set to take real numbers over the range [−3, 3]. For initial values, random numbers of [0, 1] are used for the genes for the weights, and the random variates from standard normal distribution are used for the genes for the representative firms' variables.

Jae H. Min, **Chulwoo Jeong (2008):** 100 chromosomes.

- Bias Terms (is like the Threshold in the single Perceptron)

Nii O. Attoh-Okine (1999): Bias is a constant input given to neurons. e.g. in a normal feed forward network, you might have 2 input units, 2 hidden units and 1 output unit. a constant bias value (let's say 1) will go into the hidden and output units in addition to the input from the input units. Neural Networks are like black boxes in that they cannot identify the relative importance of each of the input variables. They also need a long training process.

5. EMPIRICAL RESEARCH

5.1 INPUT DATA SELECTION AND PREPROCESSING

The data concern a large number of Greek industrial companies, provided by the database of a regional business services group. The classification has been based on actual information regarding the credit quality of the various companies such as loan delinquencies and unpaid checks. The observations concern 11.334 Greek companies from various business sectors. The companies have been classified by the bank specialist in two groups. The first group is used as the default one and includes 321 companies. The second group is the non-default one and consists 11.013 companies. The dependent variable y takes the value of 0 if there is default and a value of 1, if there is no default. The credit risk classification problem will be to discriminate the companies in these two groups. Classifying a company in class 1 means that the company's potential loans is low risk and acceptable. Classifying an observation in class 2 means that the company's potential loans is of high risk and unacceptable**.** Table 5 presents the eight inputs to the models. The inputs 1 to 4 and 6 to 7 concerns financial ratios and the inputs 5 and 8 concern absolute values.

INPUT INDEX	NOTATION	DESCRIPTION OF INPUTS
1	EBIT/TA	Earnings before income tax / total assets
2	D/NFA	Depreciation / net fixed assets
3	E/TA	Equity / total assets
4	IE/NS	Interest expenses / net sales
5	CP	Collection period
6	NS/TA	Net sales / total assets
7	NS/CL	Net sales / current liabilities
8	NS	Net sales

Table 5. Input Variables

	N	Range	Skewness		Kurtosis	
	Statistic	Statistic	Statistic	Std. Error	Statistic	Std. Error
profit before income tax/total assets	11334	,738013997	,269	,023	3,466	,046
Depreciation/Net fixed assets	11334	,975799772	-,151	,023	-,791	,046
equity/total assets	11334	1,114837678	,107	,023	-,569	,046
interest expenses/net sales	11334	,263705812	-3,254	,023	14,862	,046
collection period	11334	339	,540	,023	-,402	,046
net sales/total assets	11334	4,14555179	1,722	,023	4,497	,046
net sales/current liabilities	11334	1,540896721E1	3,041	,023	12,153	,046
net sales	11334	76859304	4,804	,023	25,161	,046
Valid N (listwise)	11334					

Table 6. Descriptive Statistics

		profit before income tax/total assets	Depreciation/Net fixed assets	equity/total assets	interest expenses/net sales
profit before income tax/total assets	Pearson Correlation	1	,137**	,075**	,267**
	Sig. (2-tailed)		,000	,000	,000
	N	11334	11334	11334	11334
Depreciation/Net fixed assets	Pearson Correlation	,137**	1	,026**	,134**
	Sig. (2-tailed)	,000		,006	,000
	N	11334	11334	11334	11334
equity/total assets	Pearson Correlation	,075**	,026**	1	,122**
	Sig. (2-tailed)	,000	,006		,000
	N	11334	11334	11334	11334
interest expenses/net sales	Pearson Correlation	,267**	,134**	,122**	1
	Sig. (2-tailed)	,000	,000	,000	
	N	11334	11334	11334	11334

Table 7. Correlations I

**. Correlation is significant at the 0.01 level (2-tailed).

		collection period	net sales/total assets	net sales/current liabilities	net sales
collection period	Pearson Correlation	1	-,438**	-,334**	-,024**
	Sig. (2-tailed)		,000	,000	,009
	N	11334	11334	11334	11334
net sales/total assets	Pearson Correlation	-,438**	1	,405**	,046**
	Sig. (2-tailed)	,000		,000	,000
	N	11334	11334	11334	11334
net sales/current liabilities	Pearson Correlation	-,334**	,405**	1	,026**
	Sig. (2-tailed)	,000	,000		,005
	N	11334	11334	11334	11334
net sales	Pearson Correlation	-,024**	,046**	,026**	1
	Sig. (2-tailed)	,009	,000	,005	
	N	11334	11334	11334	11334

Table 8. Correlations II

**. Correlation is significant at the 0.01 level (2-tailed).

INPUT DATA DISTRIBUTION			
	2002	**2003**	**Total**
Good Companies	5.529	5.484	11.013
Bad Companies	179	142	321
Total	5.708	5.626	**11.334**

Table 9. Number of Observations

	AVERAGE VALUES							
	EBIT / TA	**D / NFA**	**E / TA**	**IE / NS**	**CP (days)**	**NS / TA**	**NS / CL**	**NS**
Full Sample	4,37%	51,26%	42,11%	2,62%	143,70	1,02	2,48	4.784.173,56 €
Good Comp.	4,54%	51,51%	42,45%	2,54%	142,35	1,03	2,51	4.857.550,55 €
Bad Comp.	-1,64%	42,74%	30,43%	5,34%	190,09	0,71	1,33	2.266.725,42 €

Table 10. Average Values of Input Variables

CORRELATION MATRIX OF INPUT VARIABLES								
	EBIT / TA	**D / NFA**	**E / TA**	**IE / NS**	**CP (days)**	**NS / TA**	**NS / CL**	**NS**
EBIT / TA	1,00	0,14	0,07	0,27	-0,13	0,34	0,20	0,06
D / NFA	0,14	1,00	0,03	0,13	0,06	0,17	0,13	0,03
E / TA	0,07	0,03	1,00	0,12	-0,06	-0,28	0,47	-0,01
IE / NS	0,27	0,13	0,12	1,00	-0,33	0,36	0,29	0,02
CP (days)	-0,13	0,06	-0,06	-0,33	1,00	-0,44	-0,33	-0,02
NS / TA	0,34	0,17	-0,28	0,36	-0,44	1,00	0,40	0,05
NS / CL	0,20	0,13	0,47	0,29	-0,33	0,40	1,00	0,03
NS	0,06	0,03	-0,01	0,02	-0,02	0,05	0,03	1,00

Table 8. Correlation Matrix

According to the previous table, we do not observe strong linear correlations among the input variables of our sample. The most powerful interdependences are between:

- {Net Sales / Current Liabilities **&** Equity / Total Assets} with Correlation = **0.47**
- {Net Sales / Total Assets **&** Collection Period} with Correlation = **0.44**
- {Net Sales / Current Liabilities **&** Net Sales / Total Assets} with Correlation = **0.40**

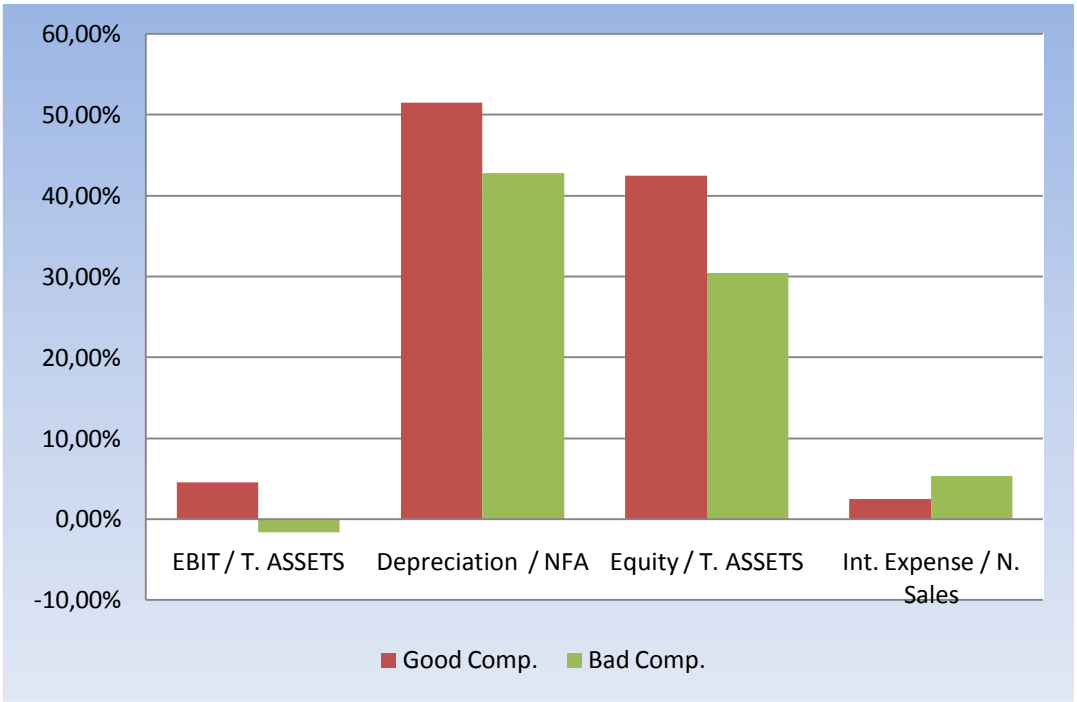

Fig. 11: Financial Ratios & Credit Risk

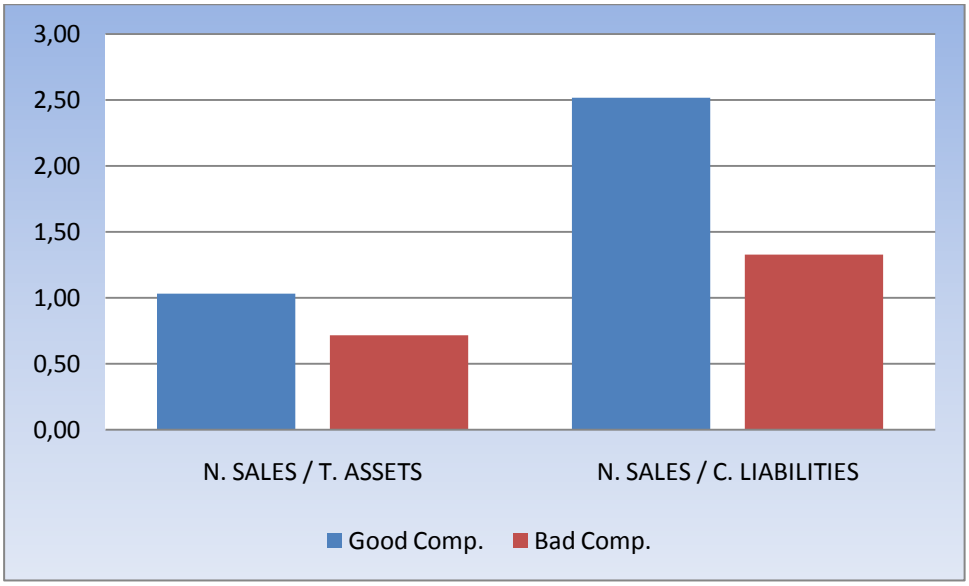

Fig. 12: Net Sales & Credit Risk

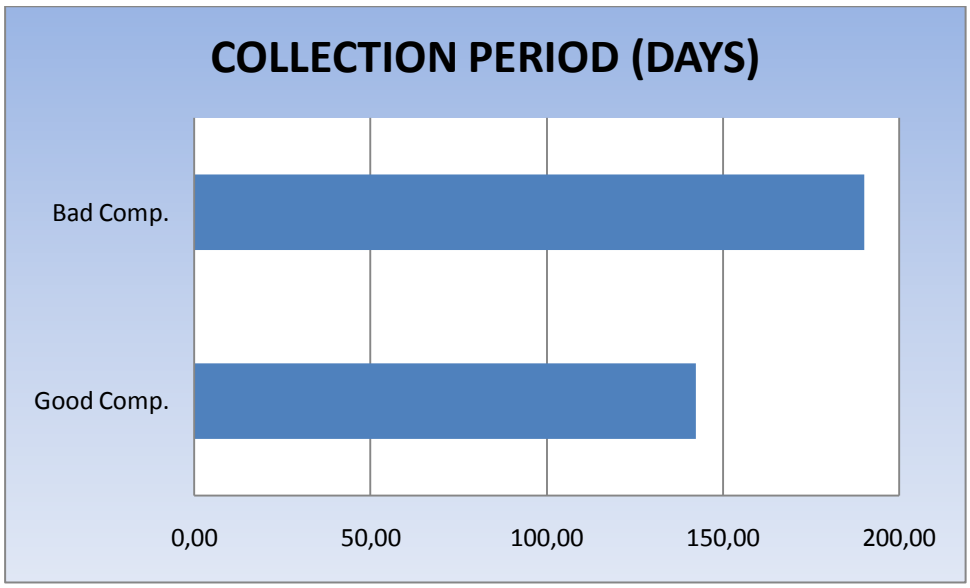

Fig. 13: Collection Period & Credit Risk

VARIANCE OF INPUT VARIABLES								
	EBIT / TA	D / NFA	E / TA	IE / NS	CP (days)	NS / TA	NS / CL	NS
Good	0,0111	0,0590	0,0571	0,0012	8226,4615	0,4556	5,2378	1,30113E+14
Bad	0,0092	0,0697	0,0604	0,0047	14075,7108	0,3733	2,5564	1,49982E+13

Table 9. Variance of Input Variables

T TEST TWO SAMPLE ASSUMING UNEQUAL VARIANCES								
	EBIT / TA	D / NFA	E / TA	IE / NS	CP (days)	NS / TA	NS / CL	NS
Test Statistic	11,29*	5,94*	8,59*	7,31*	7,10*	9,07*	12,86*	10,66*
DF	342	335	337	324	330	342	358	499
P(T<=t) one-tail	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
t Critical one-tail	1,6493	1,6494	1,6494	1,6496	1,6495	1,6493	1,6491	1,6479
P(T<=t) two-tail	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
t Critical two-tail	1,9669	1,9671	1,9670	1,9673	1,9672	1,9669	1,9666	1,9647

Table 10. Test for Equality in Mean between good and bad companies (5% Sign. Level)

Based on the above observations we can clearly derive the following relations:

1. Higher EBIT / TA => Lower Credit Risk
2. Higher Depreciation / Net Fixed Assets => Lower Credit Risk
3. Higher Equity / Total Assets => Lower Credit Risk
4. Higher Interest Expense / Net Sales => Higher Credit Risk
5. Higher Collection Period => Higher Credit Risk
6. Higher Net Sales / Total Assets => Lower Credit Risk
7. Higher Net Sales / Current Liabilities => Lower Credit Risk
8. Higher Net Sales => Lower Credit Risk

The first step before the initialization of the training and validation phase of the neural network is the transformation of the input data. This procedure is done in order to facilitate the nonlinear estimation process and to avoid underflow or overflow problems. After the training and validation of the model, the output of the neural network is transformed again in its original form Helge Petersohn scaling function is used to scale the data for zero to one, denoted [0,1]. It transforms a variable x_k to z_k and is calculated by the following formula:

$$z_{k,t} = \frac{1}{1+\exp\left[\left(\frac{\ln\left[\bar{z}_k^{-1}-1\right]-\ln\left[\underline{z}_k^{-1}-1\right]}{\max(x_k)-\min(x_k)}\right)\left[x_{k,t}-\min(x_k)\right]+\ln\left[\underline{z}_k^{-1}-1\right]^{-1}\right]}$$

5.2 NEURAL NETWORK & GENETIC ALGORITH ARCHITECTURE

This study uses a feed forward network that consisted by an input layer, an intermediate hidden layer which has three neurons, and an output layer. One hidden layer is very commonly used in economic and financial applications. The model has eight inputs nodes representing the firm applicant's characteristics and one output node representing the identified class. The neuron transfer function which is applied is a sigmoid function, which exhibits desirable properties such as being nonlinear and continuously differentiable. The objective function is computed for each candidate solution. In this study we use the sum of squared errors for the objective function to be consistent with BP. A probability is assigned to each solution based on the value of the objective function. For example, using the sum of the squared errors, the solutions which result in the smallest sum of squared errors are assigned the highest probabilities. The second generation begins by randomly selecting a new population from the former.

A prespecified number of solutions are chosen with replacement so that good solutions are likely to be well represented in the new population and poor solutions are unlikely to be drawn. This is known as reproduction. This new population of solutions (all of which existed in the prior generation) is next randomly grouped into pairs of solutions and a subset of the weights from each solution are switched with its paired solution (crossover). This creates two possible solutions, each with some parameters (weights) from each of the parent solutions. Finally, each solution has a small probability that any of its weights may be replaced with a value uniformly selected from the parameter space (mutation). In this study 50% of the data concern the year 2002 and the rest 50% of the data concerns the year 2003.

5.3 RESEARCH DESIGN

The number of hidden layers is set equal to 1, according to Han, J. W., & Kamber, M. (2001), Cybenko (1989), Lippman (1987), Rumelhart et al. (1986), Zhang et al. (1999), Randall S. Sexton and Jatinder N. D. Gupta (2000) and Chih-Fong Tsai (2009). The number of input nodes is set equal to 8 the same as the number of input variables. The number of hidden nodes is set equal to 4 according to Tseng F.M., Yi-Chung Hub (2010), Nelson and Illingworth (1991), Randall S. Sexton and Jatinder N. D. Gupta (2000) and Hsieh (2005). The neuron transfer function is the sigmoid according to Han, J. W. & Kamber, M. (2001), Zhang et al. (1999), Nii O. Attoh-Okine (1999). The Learning / Testing Ratio is set equal to (60% / 40%) according to Jae H. Min, Chulwoo Jeong (2008). The learning rate is set equal to 0.1 according to Nii O. Attoh-Okine (1999). The testing tolerance (=0.000001).

- True Positive Accuracy: classification performance of healthy companies.
- True Negative Accuracy: classification performance of bankrupt companies.
- Type I error is the frequency of misclassification of an observed bankrupt company.
- Type II error is the frequency of misclassification of a non bankrupt company.

- BPN (Back Propagated Neural Network)
- GA (genetically evolved neural network)
- DA (Discriminant Analysis).
- IN (In Sample – Learning)
- OUT (Out of Sample – Testing)

5.3.1 THE EFFECTS OF THE NUMBER OF ITERATIONS IN THE NEURAL NETWORK OPTIMIZATION

1. Matched Sample: **Wilson and Sharda (1994), Zhang et al. (1999), Jae H. Min, Chulwoo Jeong (2008)**

- Number of Bankrupt Companies: 321 & Number of Non Bankrupt Companies: 321

- Number of Iterations: 100 [a] & 500 [b] & 800 [c], **Chih-Fong Tsai (2009)**

Table 11.1 Predictive Accuracy in Training / Testing Sample (%)

Methodology		W. Average Accuracy			True Positive Accuracy			True Negative Accuracy		
		a	b	c	a	b	c	a	b	c
BPN	IN	86.06	83.96	91.74	100	67.91	87.85	72.12	100	95.64
	OUT	72.37	77.63	85.02	100	55.25	78.99	44.75	100	91.05
GA *	IN	93.38	93.22	92.99	92.37	94.86	93.93	94.39	91.59	92.06
	OUT	85.99	85.21	86.58	84.05	89.11	89.11	87.94	81.32	84.05
DA	IN	87.92			87.01			88.83		
	OUT	84.63			84.05			85.21		
LOGIT	IN	91.82	93.30	93.22	89.41	93.77	93.77	94.24	92.83	92.68
	OUT	84.44	85.02	84.44	79.38	87.55	86.77	89.49	82.49	82.10
PROBIT	IN	93.30	93.22	93.07	93.30	93.61	93.46	93.30	92.83	92.68
	OUT	77.63			55.25			100		
GOMPIT	IN	86.06			100			72.12		
	OUT	77.63			55.25			100		

*Population Size: 50 & Number of Generations: 100, **Tseng F.M., Yi-Chung Hub (2010)**

Table 11.2 Prediction Errors In Training / Testing Sample (%)								
Methodology		Type I			Type II		
		a	b	c	a	b	c
BPN	IN	27.88	0	4.36	0	32.09	12.15
	OUT	55.25	0	8.95	0	44.75	21.01
GA*	IN	5.61	8.41	7.94	7.63	5.14	6.07
	OUT	12.06	18.68	15.95	15.95	10.89	10.89
DA	IN	11.17			12.99		
	OUT	14.79			15.95		
LOGIT	IN	5.76	7.17	7.32	10.59	6.23	6.23
	OUT	10.51	17.51	17.90	20.62	12.45	13.23
PROBIT	IN	6.70	7.17	7.32	6.70	6.39	6.54
	OUT	0			44.75		
GOMPIT	IN	27.88			0		
	OUT	0			44.75		

*Population Size: 50 & Number of Generations: 100 **Tseng F.M., Yi-Chung Hub (2010)**

In **Tables 11.1 & 11.2** we can observe the following:

1. The average classification accuracy of the BP algorithm increases when we augment the number of iterations from 100 to 800. More specifically, when we use a larger number of replicating steps in the algorithm we are able to achieve a higher classification performance of bankrupt companies. This element is important, as most of the financial institutions, seek to minimize this aspect of counterparty risk, or else words, the type I error.

2. For all the methodologies, we observe deterioration in the average classification performance in the out of sample case. This may be partly attributed the 60% / 40% ratio used for training and testing.

3. The Back Propagated Neural Network demonstrates higher classification accuracy for bankrupt companies than for non bankrupt when we use 500 and 800 observations, while the inverse is true in case we use 100 observations. GA, LOGIT and Discriminant Analysis exhibit relative stability in their classification performance, while GOMPIT posses high performance in the classification of non bankrupt companies in the training sample, but low in the testing sample.

2. Non Matching Sample: **Hsieh (2005), Baesens (2003), Kim and Sohn (2004), Rumelhart et al. (1986)**

- Number of Bankrupt Companies: 321 & Number of Non Bankrupt Companies: 1007

- Number of Iterations: 100 [a] & 500 [b] & 800 [c], **Chih-Fong Tsai (2009)**

Table 11.3 Predictive Accuracy in Training / Testing Sample (%)

Methodology		W. Average Accuracy			True Positive Accuracy			True Negative Accuracy		
		a	b	c	a	b	c	a	b	c
BPN	IN	93.26	93.26	93.26	100	100	100	86.52	86.52	86.52
	OUT	86.63	86.63	86.63	100	100	100	73.26	73.26	73.26
GA *	IN	93.26	93.64	93.41	95.93	96.61	96.99	90.59	90.66	89.83
	OUT	88.79	88.42	88.70	94.92	95.29	97.55	82.67	81.54	79.85
DA	IN	85.88			77.54			94.23		
	OUT	85.59			77.21			93.97		
LOGIT	IN	93.49	93.60	93.56	99.55	97.36	97.29	87.42	89.83	89.83
	OUT	87.29	88.79	88.79	99.62	97.36	97.36	74.95	80.23	80.23
PROBIT	IN	93.26	93.52	93.64	100	97.44	97.52	86.52	89.61	89.76
	OUT	63.37			26.74			100		
GOMPIT	IN	93.26			100			86.52		
	OUT	63.37			26.74			100		

*Population Size: 50 & Number of Generations: 100 **Tseng F.M., Yi-Chung Hub (2010)**

Table 11.4 Prediction Errors In Training / Testing Sample (%)							
Methodology		Type I			Type II		
		a	b	c	a	b	c
BPN	IN	13.48	13.48	13.48	0	0	0
	OUT	26.74	26.74	26.74	0	0	0
GA*	IN	9.41	9.34	10.17	4.07	3.39	3.01
	OUT	17.33	18.46	20.15	5.08	4.71	2.45
DA	IN	5.77			22.46		
	OUT	6.03			22.79		
LOGIT	IN	12.58	10.17	10.17	0.45	2.64	2.71
	OUT	25.05	19.77	19.77	0.38	2.64	2.64
PROBIT	IN	13.48	10.39	10.24	0	2.56	2.48
	OUT	0			73.26		
GOMPIT	IN	13.48			0		
	OUT	0			73.26		

*Population Size: 50 & Number of Generations: 100 **Tseng F.M., Yi-Chung Hub (2010)**

In **Tables 11.3 & 11.4** we can observe the following:

1. For all the methods, the average classification performance, deteriorates when we move to the testing sample.

2. Discriminant Analysis compared to the BP algorithm demonstrates higher performance in the classification of bankrupt companies, while the Back Propagated Network, dominates Discriminant Analysis in the classification of non bankrupt companies.

3. Comparing the performance of GA and BP, we observe almost equal average accuracy, while the first attains superior performance in the case of bankrupt companies and the second in the case of non bankrupt companies.

4. LOGIT approximately exhibits an equal performance with the genetic algorithm approach, while the latter seems more efficient in the classification of bankrupt companies for both in sample and out of sample cases.

5.3.2 PERFORMANCE OF THE HYBRID NEUROGENETIC ALGORITHM

In this section, we study the performance of a proposed mixture of optimization methods that makes use in successive order of both the gradient descent optimization technique and the genetic algorithm evolutionary approach. The above hybrid is compared to the more traditional approaches of a single network optimization technique.

- Number of Iterations: 500 & **1.000** (the results for the latter are in brackets)
- Number of Generations: 100
- Population Size: 50
- Matching Sample (Bankrupt & Non Bankrupt)
- Training / Testing: 60% / 40%

Table 11.5 Predictive Accuracy In Training / Testing Sample (%)

Methodology		W. Average Accuracy	TP Accuracy	TN Accuracy
Back Propagated N.Network	IN	92.52	92.37	92.68
	OUT	85.41	83.66	87.16
Genetically Optimized N. Network	IN	93.30	92.83	93.77
	OUT	86.19	86.38	85.99
Genetically Optimized + Back propagated N. Network	IN	83.96 **(86.06)**	67.91 **(99.69)**	100 **(72.43)**
	OUT	77.63 **(72.76)**	55.25 **(99.61)**	100 **(45.91)**

Table 11.6 Prediction Errors In Training / Testing Sample (%)			
Methodology		Type I	Type II
Back Propagated N.Network	IN	7.32	7.63
	OUT	12.84	16.34
Genetically Optimized N. Network	IN	6.23	7.17
	OUT	14.01	13.62
Genetically Optimized + Back propagated N. Network	IN	0 **(27.57)**	32.09 **(0.31)**
	OUT	0 **(54.09)**	44.75 **(0.39)**

In **Tables 11.5 & 11.6** we can observe the following:

1. The Genetically optimized Neural Network approach dominates in every aspect the Back Propagation approach for the in sample case. The BP method is slightly superior to the GA only in the classification accuracy of bankrupt companies when we use the testing sample.

2. The Hybrid Approach that combines the two methods of optimization, is by far the most formidable technique in the classification of bankrupt companies, irrespectively of the sample we use (learning or testing). It has however, inferior performance in the classification of non bankrupt companies. We shall however denote that when switching to a non matching sample (with an increased ratio of non bankrupt to bankrupt), this novel approach achieves a very well balanced score of 92.62% TP Accuracy and 92.77% TN Accuracy for the in sample case and 92.09% TP Accuracy and 85.69% TN Accuracy for the out of sample case. In such case this method is equal or even better in performance than the GA approach. Summarizing, the hybrid approach seems promising, but still needs further research in the fine tuning procedure of its parameters, due to the complexity of the method.

3. The increase in the number of iterations from 500 to 1000 had cross directional influences. More specifically, the average in sample classification accuracy improved, while the out of sample performance deteriorated.

5.3.3 ARTIFICIAL INTELLIGENCE METHODS VS DISCRIMINANT ANALYSIS

TABLE 12 – 800 Iter.		Accuracy (%)		
Method		**Average**	**TP**	**TN**
BPN	IN	91.74	87.85	95.64*
	OUT	85.02	78.99	91.05*
GA	IN	92.99 *	93.93*	92.06
	OUT	86.58 *	89.11*	84.05
DA	IN	87.92	87.01	88.83
	OUT	84.63	84.05	85.21

***** The highest predictive accuracy

TABLE 13 – 800 Iter.		Errors (%)	
Method		**Type I**	**Type II**
BPN	IN	4.36*	12.15
	OUT	8.95*	21.01
GA	IN	7.94	6.07*
	OUT	15.95	10.89*
DA	IN	11.17	12.99
	OUT	14.79	15.95

* The smallest percentage error

In **Tables 12 & 13** we can observe the following:

1. The use of a Genetic Algorithm (GA) approach in the Bankruptcy Prediction problem offers superior average classification performance compared with the use of the traditional Back Propagation (BP) Neural Network and the Discriminant Analysis for both the training and the testing sample.

2. The BP algorithm seems to be the most capable technique in the classification of bankrupt companies. In other words Artificial Intelligence methods dominate the commonly used statistical method of Discriminant Analysis.

3. For all the above techniques, the classification performance deteriorates in out of sample predictions. The GA approach seems to being better equipped to classify the non bankrupt companies in both the Training & Testing Sample while the other two approaches, are better in classifying bankrupt companies.

5.3.4 ARTIFICIAL INTELLIGENCE METHODS VS LOGIT & PROBIT METHODS.

TABLE 14 – 500 Iter.		Accuracy (%)		
Method		**Average**	**TP**	**TN**
BPN	IN	83.96	67.91	100*
	OUT	77.63	55.25	100*
GA	IN	93.22	94.86*	91.59
	OUT	85.21*	89.11*	81.32
LOGIT	IN	93.30*	93.77	92.83
	OUT	85.02	87.55	82.49
PROBIT	IN	93.22	93.61	92.83
	OUT	77.63	55.25	100*

* The highest predictive accuracy

TABLE 15 – 500 Iter.		Errors (%)	
Method		**Type I**	**Type II**
BPN	IN	0*	32.09
	OUT	0*	44.75
GA	IN	8.41	5.14*
	OUT	18.68	10.89*
LOGIT	IN	7.17	6.23
	OUT	17.51	12.45
PROBIT	IN	7.17	6.39
	OUT	0*	44.75

* The smallest percentage error

In **Tables 14 & 15** we can observe the following:

1. The GA approach exhibits the higher average predictive performance in the Testing Sample, while the LOGIT method has slightly superior accuracy than the GA in the Learning Sample.

2. GA dominates all the other techniques in the Non Bankrupt companies' classification domain. When, however it comes, to the case of Bankrupt companies then, it seems that the BP Algorithm and interestingly the PROBIT method, demonstrate the best performance.

3. PROBIT method is the only technique that achieves a very high performance in the classification of Bankrupt companies and for the Testing sample. This capability is very interesting and can lead to further investigations.

5.3.5 MATCHED AND NON MATCHED SAMPLES.

TABLE 16 – 500 Iter.		Matched Sample Accuracy (%)			Non Matched Sample Accuracy (%)		
Method		**Average**	**TP**	**TN**	**Average**	**TP**	**TN**
BPN	IN	83.96	67.91	100	93.26	100	86.52
	OUT	77.63	55.25	100	86.63	100	73.26
GA	IN	93.22	94.86	91.59	93.64	96.61	90.66
	OUT	85.21	89.11	81.32	88.42	95.29	81.54
DA	IN	87.92	87.01	88.83	85.88	77.54	94.23
	OUT	84.63	84.05	85.21	85.59	77.21	93.97
LOGIT	IN	93.30	93.77	92.83	93.60	97.36	89.83
	OUT	85.02	87.55	82.49	88.79	97.36	80.23
PROBIT	IN	93.22	93.61	92.83	93.52	97.44	89.61
	OUT	77.63	55.25	100	63.37	26.74	100

In **Table 16** we can observe the following:

1. For all the methods except from the PROBIT, the average classification performance improves when we introduce in our sample, a larger number of companies. In the latter, the out of sample average classification capability deteriorates with the inclusion of a larger number of non bankrupt companies. While counterintuitive the classification performance of PROBIT for non bankrupt companies vastly decays when we introduce more healthy companies in our sample. Surprisingly enough, irrespectively of the sample used (matched / non matched), the classification accuracy of PROBIT, remains stable in the out of sample case, and extraordinarily high.

2. The BP Algorithm demonstrates an important improvement in the classification performance of non bankrupt companies, when we change the ratio of bankrupt and non bankrupt companies (more non bankrupt). This was naturally expected as Neural Networks, are by construction mapping mechanisms that are trained to learn the intrinsic characteristics of the sample that we input to them. In other words by augmenting the number of healthy companies, we simply help the Network, to be able to classify more efficiently the non bankrupt companies.

5.3.6 CHROMOSOME POPULATION SIZE & GENETIC ALGORITHM PERFORMANCE

TABLE 17 – Matched Sample		Accuracy (%)		
GA	**100 Generations**	**Average**	**TP**	**TN**
50	IN	91.74	94.24	89.25
	OUT	84.05	89.49	78.60
100	IN	92.99	92.06	93.93
	OUT	85.02	84.05	85.99
150 (200 gen)	IN	92.13 (93.15)	94.24 (91.90)	90.03 (94.39)
	OUT	83.66 (85.80)	90.27 (84.44)	77.04 (87.16)
200	IN	92.52	95.79	89.25
	OUT	85.41	93.39	77.43

In **Table 17** we can observe the following:

1. There is a weakly positive relation between the population size and the average classification performance of the genetic algorithm. However no significant improvement is detected when we augment the population of chromosomes. The most notable change is in the classification accuracy of bankrupt companies when the number of chromosomes increases from 50 to 100. In other words when we want to improve the classification prediction performance for financially distressed companies it's better to use a slightly increased number of population members. In this way, we can minimize the Type I error, that is usually the number one priority when deciding whether to give a loan to a potential customer or not.

2. We also observe that when a further increase in the number of chromosomes is applied (from 100 to 150) then deterioration in the performance of the net is observed. This issue is resolved when we contemporaneously increase the number of generations. This means that in Genetic Algorithm optimization there must be an optimal ratio of generation's number and population size. It's not efficient to change only one of these two parameters each time we want to improve the performance of our model. The researchers should always keep in mind to sustain an optimal mixture of this values in order to attain the best possible results.

5.3.7 NUMBER OF GENERATIONS & GENETIC ALGORITHM PERFORMANCE

TABLE 18 – Matched Sample		Accuracy (%)		
GA	**PopSize 50**	**Average**	**TP**	**TN**
50	IN	93.38	94.70	92.06
	OUT	83.85	87.55	80.16
100	IN	93.07	93.93	92.21
	OUT	85.41	87.94	82.88
150	IN	93.15	93.15	93.15
	OUT	84.24	85.60	82.88
300	IN	92.91	93.93	91.90
	OUT	85.02	86.38	83.66

In **Table 18** we can observe, that no significant change in the classification accuracy is observed when we transit through alternate number of generations. The highest in sample average accuracy is demonstrated when we use 50 generations, while for the out of sample case, the average accuracy is maximized when we use 100 generations.

5.3.8 NUMBER OF ITERATIONS IN BACK PROPAGATION ALGORITHM PERFORMANCE

TABLE 19 – Matched Sample		Accuracy (%)		
BPN Iterations		**Average**	**TP**	**TN**
500	IN	83.96	67.91	100
	OUT	77.63	55.25	100
800	IN	91.74	87.85	95.64
	OUT	85.02	78.99	91.05
2.000	IN	90.81	83.49	98.13
	OUT	82.49	67.70	97.28
10.000	IN	86.53	73.52	99.53
	OUT	79.38	58.75	100

In **Table 19** we can observe the following:

1. There is a non monotonic relation between the number of iterations and the average accuracy of the back propagated neural network. The highest average classification accuracy is attained we use 800 iterations in the optimization process. From the above behavior, it can be easily derived that a large number of repetitions in the optimization phase does not always lead to superior results. We must also notice that the out of sample performance of the BP algorithm, even for the 10.000 iterations case, is still better than that observed when we use 500 iterations. This means that the network needs a relatively large number of observations in order to achieve adequate classification performance in out of sample predictions.

2. The dynamics of true negative accuracy exhibit an interesting pattern. The highest classification performance of the BP algorithm is observed when we use 500 and 10.000 iterations irrespectively of the sample we use (training or/and testing).

5.3.9 NUMBER OF HIDDEN NEURONS AND NEURAL NETWORK PERFORMANCE

TABLE 20 – Matched Sample		Accuracy (%)		
BP 500 Iter	**500 Iter**	**Average**	**TP**	**TN**
3	IN	93.15	93.61	92.68
	OUT	85.21	87.55	82.88
6*	IN	89.95	81.78	98.13
	OUT	83.46	70.43	96.50
GA Pop: 50 Gen: 100				
3	IN	92.99	94.70	91.28
	OUT	85.80	89.88	81.71
6**	IN	92.76	94.24	91.28
	OUT	85.80	89.49	82.10

In **Table 20** we can observe the following:

1. When we use the BP algorithm we observe that an increase in the number of hidden neurons from 3 to 6 leads to a deterioration of classification accuracy of non bankrupt companies, but simultaneously to a significant (especially for the out of sample case) improvement in the classification performance of bankrupt companies. In other words it depends on the importance that the researcher attributes in the minimization of type I and type II errors in the decision of the number of nodes that he will use for the hidden layer.

2. In contrast to the BP algorithm, the Genetic Algorithm optimization approach of the Neural Networks, exhibits significant robustness in the change of the number of hidden units. This minimizes the risk of choosing a non optimal number of hidden units when we make use of the evolutionary algorithm approach.

3. When we increase the number of iterations from 500 to 1.000, the performance of the BP algorithm deteriorates for both the training and testing samples. A further increase in the number of hidden nodes from 3 to 9 keeping the number of iterations fixed, did not provided superior results to the 3-nodes setting. Not substantial improvement is observed in the GA performance when we increase the number of hidden units from 6 to 9.

5.3.10 THE EFFECTS OF THE LEARNING / TESTING RATIO

TABLE 21 – 500 Iter. Matched Sample		**60% Training - 40% Testing Accuracy (%)**			**70% Training - 30% Testing Accuracy (%)**		
Method		**Average**	**TP**	**TN**	**Average**	**TP**	**TN**
BPN	IN	83.96	67.91	100	91.12	94.39	87.85
	OUT	77.63	55.25	100	81.09	94.82	67.36
GA	IN	93.22	94.86	91.59	91.12	94.55	87.69
	OUT	85.21	89.11	81.32	80.83	95.34	66.32
DA	IN	87.92	87.01	88.83	88.20	86.19	90.20
	OUT	84.63	84.05	85.21	85.49	89.12	81.87
LOGIT	IN	93.30	93.77	92.83	92.52	93.61	91.43
	OUT	85.02	87.55	82.49	82.64	92.75	72.54
PROBIT	IN	93.22	93.61	92.83	92.45	93.61	91.28
	OUT	77.63	55.25	100	86.79	73.58	100

In **Table 21** we can observe the following:

1. The BP algorithm demonstrates significant improvement in the classification accuracy of non bankrupt companies, when we use a larger portion of our sample for training. It exhibits however a dramatic deterioration in the classification performance of bankrupt companies especially in the out of sample case.

2. For all the methods, except for the PROBIT, the classification accuracy of bankrupt companies in the out of sample case deteriorates. This phenomenon can be attributed to the distributional characteristics of our sample. The Training sample contains 179 bankrupt companies and 270 non bankrupt companies, while the testing sample contains 142 bankrupt companies and only 51 non bankrupt. The above synthesis had as a consequence the models to be well trained to identify non bankrupt companies. The large number of bankrupt companies in the testing sample led naturally in a downward direction the classification accuracy of bankrupt companies.

3. As expected the GA algorithm exhibited superior average classification performance compared to the BP optimized neural network in the 60% - 40% ratio, while the two methods are on a par in the 70% - 30% case. In the following table we present the performance results in the case we use an equal sized training and testing sample. GA once more surpasses in average accuracy the BP algorithm.

TABLE 22 – 500 Iter. Matched Sample		**50% Training - 50% Testing Accuracy (%)**		
Method		**Average**	**TP**	**TN**
BPN *	IN	86.99	73.99	100
	OUT	78.35	56.70	100
GA	IN	93.77	95.48	92.06
	OUT	86.14	90.34	81.93
DA	IN	87.69	85.67	89.72
	OUT	87.07	84.42	89.72
LOGIT	IN	94.16	94.55	93.77
	OUT	85.98	86.92	85.05
PROBIT	IN	94.24	94.86	93.61
	OUT	77.88	55.76	100

* Increasing the number of iterations from 500 to 1.000 leads to inferior results.

CONCLUSIONS & FUTURE RESEARCH PATHS

Credit scoring has gained more and more attention as the competition between financial institutions has come to a totally conflicting stage. More and more financial institutions are seeking better strategies through the help of credit scoring models and hence credit scoring techniques have been widely used. Modeling techniques like traditional statistical analyses and artificial intelligence techniques have been developed in order to successfully attack the credit scoring tasks. Discriminant analysis is the most commonly used statistical credit scoring techniques, but often being criticized due to its strong model assumptions. On the one hand, the artificial neural networks is becoming a very popular alternative in the credit scoring tasks due to its associated memory characteristic, generalization capability and outstanding credit scoring capability. However, it is also being criticized for its long training process. The failure of a company is a somewhat complex field which is still not the subject of a complete theory. On the other hand, the large availability of the companies accounting elements on the banking data bases, meets the need for the neural networks, as this methodology needs large samples, in order to work properly and produce valid results. Bankruptcy forecast is not a problem that can be separated linearly. An artificial neural network does not require a linear assumption of separation of the classes. The bankruptcy forecast remains a multidimensional issue. It is a problem of classification and not a problem of chronological series forecast. To a large extent, it is quite easy to account for the popularity of the multi-layer perceptron. On the one hand, this type of network has already been the subject of many conceptual studies allowing the determination of its capacities more accurately (particularly within the scope of classification), although the theoretical approach remains inadequate to allow its construction without difficulty. On the other hand, the availability of the accounts data on sound and failing companies perfectly fits in with the required network (input and output data).

The purpose of this study was to explore the performance of credit scoring using hybrid modeling procedure in integrating the neural network approach with the genetic algorithms optimization technique. For verifying the feasibility on this proposed integrated approach, the credit scoring task was performed on large dataset of Greek companies. Analytic results demonstrated that neural network models had the highest average correct classification rate in comparison with Discriminant analysis, PROBIT and GOMPIT models, while the LOGIT model, also exhibited very good performance. These findings justify the presumptions that neural networks having better capability of capturing linear & nonlinear relationship among variables. The research findings did support the hypothesis that the two-stage hybrid credit scoring approach proposed in this study will

have better credit scoring accuracies and better convergence characteristics for the designed neural networks model, but it is still needed further research.

Future researches may aim at collecting more important variables that will increase the credit scoring accuracies. Using other newly developed classification methodologies, like classification and regression tree (CART) and multivariate adaptive regression splines (MARS), in evaluating their credit scoring capabilities are also recommended. Integrating other artificial intelligence techniques, like fuzzy set theory, in further refining the network structure may lead in improving the credit scoring accuracies. There is also a number of open issues that should be addressed by the research community. Even though a prediction of the default event is by itself very useful, an estimate of the default probability is very desirable. Having a probability of default rather than a (binary) prediction of default is valuable for a bank. The other open issue is to consider macroeconomic indicators as inputs to the NN. The prevailing economic conditions (as well as the current interest rates) can have a significant effect on the probability of bankruptcy. There are very few studies that consider these factors in conjunction with NN models. This should therefore be a recommended study.

REFERENCES

- Altman, E. I. (1968), Financial ratios, Discriminant analysis and the prediction of corporate bankruptcy, Journal of Finance 23, 589–609.
- Altman et al., (1977), ZETA ANALYSIS, a new model to identify bankruptcy risk of corporations, Journal of Banking and Finance 1, 29–54.
- Altman, E. I., (1983), Corporate financial distress — A complete guide to predicting, avoiding and dealing with bankruptcy, Wiley, New York.
- Altman, E. I., G. Marco, and F. Varetto (1994), Corporate Distress Diagnosis: Comparisons Using Linear Discriminate Analysis and Neural Networks, Journal of Banking and Finance, 18, 505–529.
- Anandarajan et al., (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 & Management 10, 69–81.
- Atsalakis G., Doumpos M., Zopounidis C. (2007), Credit Risk Assessment by neural networks and genetic algorithms, Technical University of Crete, Department of Production Engineering and Management, Financial Engineering Laboratory, Working Paper
- Baek J. and S. Cho (2003), Bankruptcy Prediction for credit risk using an auto associative neural network in Korean firms, in: IEEE International Conference on Computational Intelligence for Financial Engineering, Hong-Kong.
- Barbro B., Teija L. and Kaisa S. (1996), Neural networks and genetic algorithms for bankruptcy predictions Expert Systems with Applications, Volume 11, Issue 4, 407-413
- Bart Baesens, Rudy Setiono, Christophe, Mues Jan Vanthienen (2003), Using Neural Network Rule Extraction and Decision Tables for Credit-Risk Evaluation, Journal Management Science archive Volume 49 Issue 3.
- Beaver W. (1966), Financial ratios as prediction of failure. Empirical research in accounting: Selected studies. Journal of Accounting Research 4, 71–111.
- Bell T.B.(1997), Neural nets or the logit model? A comparison of each model's ability to predict commercial bank failures, International Journal of Intelligent Systems in Accounting, Finance and Management 6, 249–264.
- Bian H., L. Mazlack (2003), Fuzzy-rough nearest neighbor classification approach, in: 22nd International Conference of the North American Fuzzy Information Processing Society Proceedings, 500–505.

- Bliss CI. (1934), The method of Probits, Science 79 (2037): 38–39
- Blum, M. (1974), Failing company Discriminant analysis. Journal of Accounting Research, l-25.
- Boritz J. and D. Kennedy (1995), Effectiveness of neural networks types for prediction of business failure. Expert Systems with Applications 9, 503–512.
- Boritz and Kennedy (2007), Predicting Business Failures in Canada, Accounting Perspectives, Volume 6, Issue 2, 141–165,
- Brockett et al. (2007), A comparison of neural network, statistical methods, and variable choice for life insurers' financial distress prediction. Journal of Risk and Insurance, 73, 397-419
- Calderon T.G. and J.J. Cheh (2002), A Roadmap for future neural networks research in auditing and risk assessment, International Journal of Accounting Information Systems 3, 203–236.
- Chih-Fong Tsai (2009), Feature selection in bankruptcy prediction, Knowledge-Based Systems, Volume 22, Issue 2, 120-127
- Chih-Fong Tsai, Ming-Lun Chen (2010), Credit rating by hybrid machine learning techniques, Applied Soft Computing archive Volume 10 Issue 2.
- Chung H. and K. Tam (1992), A comparative analysis of inductive learning algorithm. Intelligent Systems in Accounting, Finance and Management 2, 3–18.
- Coats, P., and L. Fant (1992), A Neural Network Approach to Forecasting Financial Distress, Journal of Business Forecasting, 10, 9-12.
- Coats, P., and L. Fant, (1993), Recognizing Financial Distress Using a Neural Network Tool, Financial Management, 142-155
- Cogger KO, Fanning K. (1997) An introduction to adaptive logic networks with an application to audit risk assessment. Unpublished working paper, Central Missouri State University.
- Cybenko, G. (1989), Approximation by superpositions of a sigmoidal function. Mathematics of Control, Signals, and Systems, 2, 303–314.
- Davis et al., (1997), Supporting a complex audit judgment task: an expert network approach. European Journal of Operational Research 103, 350–372.
- Deakin, E. (1974), A Discriminant analysis of predictors of business failure. Journal of Accounting Research, 167-179.
- Deakin E.B. (1976), Distributions of Financial Accounting Ratios, Some Empirical Evidence: The Accounting Review, 90-96
- Doumpos M., Kosmidou K., Baourakis G., Zopounidis C., (2002), Credit risk assessment using a Multicriteria hierarchical discrimination approach: A comparative analysis, European Journal of Operational Research, Vol. 138, 392-412

- Doumpos M., Pasiouras F., (2005), Developing and Testing Models for Replicating Credit Ratings: A Multicriteria Approach, Computational Economics, Vol. 25, 327-341
- Edminster R. (1972), An empirical test of financial ratio analysis for small business failure prediction. Journal of Financial and Quantitative Analysis, I, 1477-1493.
- Etheridge H. and Sriram R., (1997), A comparison of the relative cost of financial distress models. International Journal of Intelligent Systems in Acounting, Finance & Management. 6, 235–248.
- Fisher, R. A. (1936), The Use of Multiple Measurements in Taxonomic Problems, Annals of Eugenics 7, 179–188.
- Fitzpatrick, P. (1932), A comparison of the ratios of successful industrial enterprises with those of failed companies. The Accountants Publishing Company.
- Fletcher, D., E. Goss (1993), Forecasting with Neural Networks—an Application Using Bankruptcy Data. , Information Management, 24, 59–167.
- Frydman H., E.I. Altman and D. Kao (1985), Introducing recursive partitioning for financial classification: The case of financial distress, Journal of Finance 40, 269–291.
- Funahashi K.I. (1989), On the approximate realization of continuous mappings by neural networks. Neural Networks 2, 183–192.
- Gaganis, C., Pasiouras F., Tzanetoulakos, A., (2005), A Comparison and Integration of Classification Techniques for the Prediction of Small UK Firms Failure, Journal of Financial Decision Making, Vol.1, No.1, pp. 61-75
- Guoqiang Zhang, Michael Y. Hu, B. Eddy Patuwo and Daniel C. Indro (1999), Artificial neural networks in bankruptcy prediction: General framework and cross-validation analysis, European Journal of Operational Research Volume 116, Issue 1, 16-32
- Han et al., (1997), The hybrid systems for credit rating. Journal of the Korean Operations Research and Management Science Society 22, 163–173.
- Han, J. W., & Kamber, M. (2001), Data mining: Concepts and techniques. San Francisco: Morgan Kaufmann.
- Hansen et al., (1992) Artificial intelligence and generalized qualitative-response models: an empirical test on two audit decision-making domains. Decision Sciences 23, 708–723.
- Hebb, D.O. (1949). The Organization of Behavior. John Wiley & Sons, NY.
- Hennawy El, R., Morris, R. (1983), The significance of base year in developing failure prediction models. Journal of Business Finance and Accounting, 209-223.
- Holland J.H., (1975). Adaptation in natural and artificial systems, The University of Michigan Press

- Holte R.C., C4.5, Class Imbalance, and Cost Sensitivity: Why Under-Sampling beats Over-Sampling, Institute for Information Technology, National Research Council Canada, Ottawa, Ontario, Canada,
- Hornik K, Stinchcombe M, White H (1989), Multilayer feed forward networks are universal approximators, Neural Networks, 2, 359–366
- Hsieh, Nan-Chen (2005), Hybrid mining approach in the design of credit scoring models, Expert Systems with Applications, Volume 28, Issue 4, 655-665
- Hussein A. Abdou (2009), Genetic programming for credit scoring: The case of Egyptian public sector banks, Expert Systems with Applications: An International Journal archive Volume 36 Issue 9
- Ignizio J.P. and J.R. Soltyas (1996), Simultaneous design and training of ontogenic neural network classifier, Computers Operations Research 23, 535–546.
- Jae H. Min, Chulwoo Jeong (2009), A binary classification method for bankruptcy prediction. Expert Systems with Applications, 36(3): 5256-5263.
- Jain B.A., B.N. Nag, (1997), Performance evaluation of neural network decision models. Journal of Management information Systems, 14, 201–216
- Jatinder N. D. Gupta and Randall S. Sexton (1999), Comparing Back Propagation with a genetic algorithm for neural network training, Omega Volume 27, Issue 6, 679-684
- Jatinder N. D. Gupta, Randall S. Sexton, (2000) Selecting Scheduling Heuristics Using Neural Networks. INFORMS, Journal on Computing 12(2): 150-162, 13
- Jo H., I. Han and H. Lee (1997), Bankruptcy Prediction Using Case-Based Reasoning, Neural Network, and Discriminant Analysis, Expert Systems with Applications, Vol. 13 No. 2, 97-108
- Jones S., D.A. Hensher (2004), Predicting firm financial distress: A mixed logit model, Accounting Review 79, 1011–1038.
- Kerling M. (1996), Corporate distress diagnosis, an international comparison, in: A.P.N. Refenes, Y. Abu-Mostafa, J. Moody, A. Weigend (Eds.), Neural Networks in Financial Engineering, World Scientific, Singapore, 407-422
- Kim and Sohn, (2004), Managing loan customers using misclassification patterns of credit scoring model, Expert Systems with Applications. 26, 567-573.
- Kusiak A. (2001)), Feature Transformation Methods in Data Mining, IEEE transactions on electronics packaging manufacturing, vol. 24, no. 3
- Kyung-shik Shin, Yong-Joo Lee, (2002), A genetic algorithm application in bankruptcy prediction modeling. Expert Systems with Applications, Volume 23, Number 3, 321-328

- Kyung-shik Shin and Kyoung Jun Lee (2004), Bankruptcy prediction modeling using multiple neural network models, Springer.
- Kyoung-jae Kim, and Ingoo Han (2000), Genetic algorithms approach to feature discretization in artificial neural networks for the prediction of stock price index, Expert Systems with Applications 19, 125–132
- Kyoung-jae Kim and Ingoo Han (2007), A case-based reasoning system with the two-dimensional reduction technique for customer classification, Expert Systems with Applications: An International Journal, Volume 32 Issue 4
- Leshno, M., & Spector, Y. (1996). Neural network prediction analysis: the bankruptcy case. Neurocomputing, 10, 125–247.
- Lee E., John Wenyu Wand (2005), Statistical Methods for Survival Analysis, Wiley Interscience
- Lee T.S., Chenb I.F., (2005), A Two Stage Hybrid Credit Scoring Model using Artificial Neural Networks and Multivariate Adaptive Regression Splines, Expert Systems with Applications, Vol. 28, 743-752
- Lippmann R.P., (1987), An introduction to computing with neural nets. IEEE ASSP Magazine 4–22.
- Loia et al., (2000), Merging fuzzy logic, neural networks, and genetic computation in the design of a decision-support system. International Journal of Intelligent Systems 15, 575–594.
- Marinaki M., Marinakis Y., Zopounidis C., (2010), Honey Bees Mating Optimization Algorithm for Financial Classification Problems, Applied Soft Computing, Vol.10, 806-812
- Nayer W. et al., (1999), Feature-based decision aggregation in modular neural network classifiers, Pattern Recognition Letters 20, 1353-1359
- Neophytou, E., Molinero, C.M. (2004), Predicting Corporate Failure in the UK: A Multidimensional Scaling Approach, Journal of Business Finance and Accounting, Vol. 31, No.5-6, pp 677-710
- McKee T.E. (2000), Developing a bankruptcy prediction model via rough sets theory, International Journal of Intelligent Systems in Accounting, Finance and Management 9, 59–173.
- Michael J. Shaw James A. Gentry (1990), Inductive Learning for Risk Classification, IEEE Expert: Intelligent Systems and Their Applications archive Volume 5 Issue 1
- Min J.H. and Y.-C. Lee (2005), Bankruptcy prediction using support vector machine (SVM) with optimal choice of kernel function parameters, Expert Systems with Applications 28, 603–614.
- Nelson and Illingworth (1991), A Practical Guide To Neural Networks Publisher: Addison-Wesley

- Nissen, V. Propach, J. (1998) On the robustness of population-based versus point-based optimization in the presence of noise, Evolutionary Computation, IEEE Transactions, Volume: 2 Issue: 3 107 – 119
- Merwin, C. (1942), Financing small corporations: In five manufacturing industries, 1926-36. National Bureau of Economic Research.
- Messier W., J. Hansen (1988), Inducing rules for expert system development: An example using default and bankruptcy data. Management Science 34, 1403–1415.
- McCulloch, W. & Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. Bulletin of Mathematical Biophysics, 7, 115 - 133.
- Muriel P. (2006), Artificial neural networks and bankruptcy forecasting: a state of the art, Journal Neural Computing and Applications archive Volume 15 Issue 2.
- Nii O. Attoh-Okine (1999), Analysis of Learning Rate and Momentum Term in Back propagation Neural network trained to predict pavement performance, Journal of Advances in Engineering Software, Vol. 30, pp. 291-302,
- Nissen, V., Jörn Propach (1998), Optimization with noisy function evaluations, Parallel Problem Solving from Nature — PPSN V, Lecture Notes in Computer Science, 1998, Volume 1498/1998, 159-168
- Nissen, V. Propach, J. (1998), On the robustness of population-based versus point-based optimization in the presence of noise Evolutionary Computation, IEEE Transactions, Volume: 2 Issue: 3
- Odom A, Sharda R (1990), A neural network model for bankruptcy prediction. In: Trippi RR, Turban E (Eds) Neural networks in finance and investing, Probus Publishing, pp 177–186, originally presented at the IJCNN Meetings
- Östermark R. (1999), A Neuro-Genetic Algorithm for Heteroskedastic Time-Series Processes Empirical Tests on Global Asset Returns, Soft Computing - A Fusion of Foundations, Methodologies and Applications, Volume 3, Number 4, 206-220.
- Pao YH (1989), Adaptive Pattern Recognition and Neural Networks, MA: Addison-Wesley
- Parag C. Pendharkar (2007), A comparison of gradient ascent, gradient descent and genetic-algorithm-based artificial neural networks for the binary classification problem, Expert Systems, Volume 24, Issue 2, pages 65–86
- Parag C. Pendharkar (2009) Genetic algorithm based neural network approaches for predicting churn in cellular wireless network services, Expert Systems with Applications, 36 Issue 3.

- Poddig T. (1995), Bankruptcy Prediction: A Comparison with Discriminant Analysis, Neural Networks in the Capital Markets. John Wiley & Sons NY.
- Ramser, J. & Foster, L. (1931), A demonstration of ratio analysis. Bulletin No. 40, University of Illinois, Bureau of Business Research, Urbana, IL.
- Rumelhart D.E. & J. L. McClelland, (1986), Parallel distributed processing: explorations in the microstructure of cognition. Foundations, Volume 1, MIT Press.
- Salchenberger L.M., E.M. Cinar and N.A. Lash (1992), Neural networks: a new tool for predicting thrift failures. Decision Sciences, 23, 899–916.
- Sexton, R. S., Dorsey, R. E., & Johnson, J. D. (1998). Toward a global optimum for neural networks: A comparison of the genetic algorithm and back propagation. Decision Support Systems, 22, 171–185.
- Sexton, R. S., Alidaee, B., Dorsey, R. E., & Johnson, J. D. (1998). Global optimization for artificial neural networks: A tabu search application. European Journal of Operational Research, 106, 570–584.
- Sexton, R. S., Dorsey, R. E., & Johnson, J. D. (1999). Optimization of neural networks: A comparative analysis of the genetic algorithm and simulated annealing. European Journal of Operational Research, 114, 589– 601.
- Sexton, R. S., & Dorsey, R. E. (2000). Reliable classification using neural networks: A genetic algorithm and back propagation comparison. Decision Support Systems, 30, 11–22.
- Sexton, R. S, Jatinder N. D. Gupta (2000): Comparative evaluation of genetic algorithm and back propagation for training neural networks. Information Science, 129(1-4): 45-59
- Sharda R., R.L. Wilson (1993), Performance comparison issues in neural network experiments for classification problems, in: Proceedings of the 26th Hawai International Conference on System Scientists.
- Shaw M. and J. Gentry (1990), Inductive learning for risk classification. IEEE Expert, 47–53.
- Shin K.S. and I. Han (1998), Bankruptcy prediction modeling using multiple neural networks models. Proceedings of Korea Management Science Institute Conference.
- Tam K.Y. and M.Y. Kiang (1992), Managerial applications of neural networks: the case of bank failure predictions, Management Science 38, 926–947.
- Tan, C.N.W. (1996), A study of using artificial neural networks to develop an early warning predictor for credit union financial distress with comparison to the probit model Appeared in Trippi R, Turban E. In: Neural networks in finance and investing, Irwin, Chicago (IL), 329–365.

- Taffler, R., (1982), Forecasting company failure in the UK using Discriminant analysis and financial ratio data. Journal of Royal Statistical Society A, 145, 342-358.
- Tseng F.M., Yi-Chung Hub (2010), Comparing four bankruptcy prediction models: Logit, quadratic interval logit, neural and fuzzy neural networks, Expert Systems with Applications 37, 1846–1853
- Wang C.M., Yin-Fu Huang (2009), Evolutionary-based feature selection approaches with new criteria for data mining: A case study of credit approval data, Expert Systems with Applications, Vol. 36, 5900–5908
- Weiss, G. and F. Provost (2001), The Effect of Class Distribution on Classifier Learning, Technical Report ML-TR-43, Department of Computer Science, Rutgers University.
- West D., (2000), Neural network credit scoring models. Computers and Operations Research 27 1131–1152.
- Werbos Paul J. (1974), Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences. PhD thesis, Harvard University
- Werbos P. (1993), The Roots of the Backpropagation: From Ordered Derivatives to Neural Networks and Political Forecasting, John Wiley & Sons, NY.
- Wilson R.L., R. Sharda (1994), Bankruptcy prediction using neural networks, Decision Support Systems 11, 545–557.
- Whitley D. and T. Starkweather, (1990), GENITOR II: A Distributed Genetic Algorithm, Journal of Experimental and Theoretical Artificial Intelligence, 2, 189-214.
- Winakor, A. & Smith, R., (1935), Changes in the financial structure of unsuccessful industrial corporations. Bulletin No. 5 1, University of Illinois, Bureau of Business Research, Urbana, IL.
- Yang Z.R., M.B. Platt and H.D. Platt (1999), Probability neural network in bankruptcy prediction, Journal of Business Research 44, 67–74.
- Yoh-han Pao (1989), Adaptive Pattern Recognition and Neural Networks, Addison-Wesley
- Zhang et al., (1999), G. Artificial neural networks in bankruptcy prediction: General framework and cross-validation analysis, European Journal of Operational Research 116, 16–32.
- Zurada et al., (1999), Neural networks versus Logit regression models for predicting financial distress response variables. Journal of Applied Business Research 15, 1, pp. 21–30.

URL

- www.bis.org (Bank for international settlements web page)