NEURAL NETWORK FORECASTS OF SINGAPORE PROPERTY STOCK RETURNS USING ACCOUNTING RATIOS

LIU JIAFENG

(M.SC)

A THESIS SUBMITTED FOR THE DEGREE OF

MASTER OF SCIENCE (ESTATE MANAGEMENT)

DEPARTMENT OF REAL ESTATE

NATIONAL UNIVERSITY OF SINGAPORE

2003

ACKNOWLEGEMENTS

I would like to thank the following people:

My supervisor, Dr. Lawrence Chin, for his invaluable advice, guidance and encouragement, without which this work would not have been possible;

My husband, Dai Yuanshun, for his love and support forever;

My parents and brother, for their love and encouragement;

My friends or classmates Xu Min, Li Ying, Gong Yangtao, Sun Hua, Zhu Haihong, etc, who have contributed in one way or another.

TABLE OF CONTENTS

APPENDIXES 97

(23,200 words)

LIST OF TABLES

LIST OF FIGURES

Summary

The return of property stocks is one of the main research areas in property stock performance. Some research supported the notion that accounting information had the ability to predict stock returns. Furthermore, a number of studies have documented the successes of artificial neural networks in forecasting time series and cross sectional financial data. Based on these studies, this work has tried to compare the forecast of Singapore property stock returns by neural networks with that by traditional regressions using accounting ratios as input variables. This work is the first to use neural networks to examine the performance of Singapore property stocks. In Singapore, although there are some works focusing on real estate stock performance, neural network techniques are relatively scarce.

One objective of this work is to provide a practical method for investors or portfolio managers to predict stock returns since accurate forecast of stock returns is vital for the investors to pick stocks. Another objective of this work is to better identify the borderline at which neural networks can outperform traditional regression-based forecasting techniques because the opinions regarding the effectiveness of neural networks are mixed. Moreover, this work includes the Monte Carlo neural network method to improve the performance of neural network models.

This work uses four different methods: OLS neural networks, logit neural networks, OLS and logit regressions to forecast company returns one year ahead. The independent variables are 52 accounting ratios and financial variables. For point prediction models, the dependent variable is a firm's abnormal return, which measured as the firm's actual return over the next fiscal year minus the return on the portfolio of all stocks in this sample during this period. For classification problems, the dependent variable is the probability of a firm having a return above or below the median return of the sample over the upcoming fiscal year. Six years of data are used to estimate the parameters of these models.

The findings indicate that accounting ratios can serve as leading indicators of stock returns in the next year; classification models (logit regression models and logit neural networks) can outperform point estimation models (OLS regression models and OLS neural networks) for this research problem; logit neural networks can outperform all other three alternatives; Monte Carlo neural networks can improve the performance of neural networks in predicting stock returns.

Chapter 1: Introduction

1.1 Background

The performance of property related stocks is a widely researched topic in the real estate literature. In Singapore, there are some studies that focus on real estate stock performance. There are also research examined whether prices of listed property stocks reflected their corporate fundamental values (such as Sing *et al.* (2002); Sing (2001)).

One of the main research areas of property stock performance is the returns of property stocks. Numerous researchers have examined whether real estate investment offers superior return (Sagalyn (1990), Titman and Warga (1986), Liu *et al.* (1995)). Recently, some research investigated the returns of real estate stocks using international data (e.g. Glascock *et al*. (2002), Ling and Naranjo (2002), Ooi and Liow (2002)).

Traditionally, Ordinary Least Square (OLS) and logit regressions are used widely to predict stock returns. For example, some studies (e.g. Ou and Penman (1989), Holthausen and Larcker (1992), Brockman *et al*. (1997)) predicted stock returns using financial statement information by OLS stepwise regression model or logit stepwise regression model. Moreover, some research According to their results, the research verified that annual financial statement information have predictability on the next year's stock returns.

In recent years, artificial neural networks (ANNs) applications in finance for such tasks as pattern recognition, classification, and time series forecasting have dramatically increased. A number of studies have documented the successes and failures of ANNs in forecasting time series and cross sectional financial data.

Based on these studies, this work tries to compare the forecasts of Singapore property stock returns by neural networks with that by traditional regressions using accounting ratios as input variables.

1.2 Objectives of the Work

One objective of this work is to provide a practical method for investors or portfolio managers to predict stock returns. Accurate forecast of stock returns is vital for the investors to pick stocks. According to Elton and Gruber (1991), portfolio managers are generally stock pickers and only occasionally market timers. As stock pickers, their task is to pick (avoid) those stocks that are likely to outperform (under perform) other stocks of comparable risks.

Since neural networks in finance has been increased greatly in recent years and the opinions regarding its effectiveness are mixed, another objective of this work is to better identify the borderline at which neural networks can outperform traditional regression-based forecasting techniques by comparing neural network forecasts of 2-year portfolio returns of Singapore property stocks with the forecasts obtained OLS and logit regression techniques.

Moreover, this work uses Monte Carlo neural network method to improve the performance of neural network models. Also, this work is the first to use neural networks to examine the returns of Singapore property stocks.

1.3 Scope of the Work

Because of the time limitation, this work could only concern Singapore property stocks instead of all Singapore stocks. And due to the data availability, this work only includes 13 of 20 Singapore property stocks. The other 7 Singapore property stocks have many missing variables. The input data are 52 accounting ratios and annual returns for each stock in the sample over the period 1992-2001. The most recent 6 years of data are rolled forward each year to forecast annual returns for 2000 and 2001.

1.4 Methodology

This work uses four different methods: OLS neural networks, logit neural networks, OLS and logit regressions to forecast company returns of one year ahead. The independent variables are 52 accounting ratios and financial variables. For point prediction model, the dependent variable is a firm's abnormal return, which measured as the firm's actual return over the next fiscal year minus the return on the portfolio of all stocks in this sample during this period. For classification problems, the dependent variable is the probability of a firm having a return above or below the median return of the sample over the upcoming fiscal year. Six years of data are used to estimate the parameters of these models. Since previous research Ou and Penman (1989), Holthausen and Larcker (1992), Brockman, Mossman, and Olson (1997) proved that the data set had some predictability, a valid comparison between the regression and ANN forecasting techniques should be possible.

1.4.1 OLS Neural Networks and Logit Neural Networks

The back propagation (BP) is used in this work since it is the most widely used in financial time series forecasting and it is the most common type of neural networks in time series forecasting. In this work, back propagation neural networks which are constructed for point estimation are called OLS neural networks, while back propagation neural networks which are constructed for classification problems are called logit neural networks. The estimation procedures and subsequent trading strategies of neural networks for both OLS neural networks and logit neural networks are similar.

1.4.2 Stepwise OLS Regression and Logit Regression

In this work, six years of data are used to estimate the parameters of the within-sample OLS model and Logit model. The estimation procedures for OLS and logit models are similar. First, all fundamental analysis variables initially enter the regression equation. A parsimonious subset of independent variables is selected in most cases by simply eliminating any explanatory variables with *P*-values greater than certain value (such as 5% or 10%) to ensure the proper number of selected independent variables. For example, the 5% criterion is adjusted downward to 4 or 3% if the step-wise procedure originally selects more than eight independent variables; if the above step-wise procedure selects only one or two variables, the cut-off criterion would be increased to 10% to include more independent variables. In general, models with three to five independent variables tend to give the best forecasts in the validation period.

1.5 Hypotheses

(1) Accounting ratios can serve as leading indicators of stock returns in the next year.

(2) Classification models (logit regression models and logit neural networks) can outperform point estimation models (OLS regression models and OLS neural networks) for problems at hand.

(3) Neural networks can outperform the traditional OLS and logit regression models.

(4) Monte Carlo neural networks can improve the performance of neural networks in predicting stock returns.

1.6 Sources of Data

This work examines the performance of 13 Singapore property companies on Singapore stock market over the period 1992-2001. Financial statement data and stock returns of individual firms over the sample period were extracted from Datastream. This work constructs data set of 52 annual accounting ratios and financial variables for each firm during each of the 10 years in this sample. Table 1.1 lists the 52 potential input variables considered in this work. The variables include most of the 68 accounting ratios in Ou and Penman(1989) (e.g. current and quick ratios, debt-equity ratios, return on equity, etc., as well as annual percentage changes in many ratios). Moreover, since this work only studies property stocks, three ratios commonly used in the property stocks or finance literature (book value to market value, earnings–price, and the price–sales ratio) are also included in accounting ratios.

Table 1.1 Accounting Ratios and Financial Variables Used as Inputs in Models

No	Ratios	No	Ratios
1	current ratio	27	Operating profit before depreciation
2	annual % change in 1	28	Annual % change in 27
3	quick ratio	29	Pre-tax income to sales ratio
4	annual % change in 3	30	Annual % change in 29
5	Inventory turnover ratio	31	Sales to accounts receivable ratio
6	annual % change in 5	32	Sales to inventory ratio
7	inventories to total assets	33	Annual % change in 32
8	Annual % change in 7	34	Sales to net working capital
9	Annual % change in inventory	35	Annual % change in 34
10	Annual % change in sales	36	Sales to fixed assets ratio
11	Annual % change in depreciation	37	Annual % change in total assets
12	Annual % change in dividend per share	38	Annual % change in capital expenses to total assets
13	Depreciation to fixed assets	39	Ratio 38 lagged one year
14	Annual % change in 13	40	Operating income to total assets ratio
15	Net working capital to total assets ratio	41	Annual % change in 40
16	Annual % change in 15	42	Annual % change in long-term debt
17	Total debt to equity ratio	43	Annual % change in net working capital
18	Annual % change in 17	44	Earnings to price ratio
19	Long-term debt to equity ratio	45	Book value to market value ratio
20	Annual % change in 19	46	Price to sales ratio
21	Equity to fixed assets ratio	47	Total debt to market value
22	Annual % change in 21	48	Annual % change in 27
23	Times interest earned ratio	49	Dividend yield
24	Annual % change in 23	50	Annual % change in 49
25	Sales to total assets ratio	51	Total debt total assets
26	Annual % change in 25	52	Annual % change in 51

For each company, accounting ratios are matched with common stock returns. Following Ou and Penman (1989) and Holthausen and Larcker (1992), the accounting data are used with a 3-month lag to ensure that investors actually have access to the data at the time of the investment decision. This means that the accounting ratios are used to forecast 1-year-ahead returns that are calculated from months 4 to 15 following the publication of annual fiscal year accounting data.

The initial sample size for this work is 20 observations; but many accounting ratios, and occasionally returns, are not available for some companies for the whole sample period. To obtain a usable data sample, a set of restrictive filters is imposed on companies. For inclusion in the sample, a company must:

(1) have both annual accounting ratios and return data from 1992 to 2001;

(2) have returns above or below median annual return of the sample at least once during 6 year period for the use of Logit model.

These restrictions resulted in 7 observations being excluded from the sample.

1.7 Organization of This Work

This work is organized into six chapters.

Chapter One provides an overview comprising the background, aim and scope of this work. The hypotheses for the work, sources of data and the methodology are also presented in this chapter.

Chapter Two reviews the relevant literature on (i) some local research on property stock performance, (ii) property stock return, (iii) neural networks in forecasting financial problems, and (vi) traditional regression techniques in stock returns forecast.

Chapter Three explains the stepwise OLS and logit regression models and methodology used to obtain OLS and logit forecasts.

Chapter Four describes the basic knowledge and steps to build the back propagation neural networks and presents the model and architecture of neural networks used to forecast stock returns.

Chapter Five analyzes and compares the results in terms of portfolio profitability for the four forecasting techniques.

Chapter Six summarizes the results and makes recommendations for future research.

Chapter 2 : Literature Review

2.1 Introduction

This work uses several different methods to forecast property stock returns of one year ahead. The thesis reviews not only local research on property stock performance and global property stock return, but also studies on neural networks in forecasting financial problems and traditional regression techniques in stock returns forecast.

This chapter will be presented as follows. Section 2.2 reviews local research on real estate stocks. In Section 2.3, the focus is on research of property stock returns. Section 2.4 reviews research using traditional regression techniques in forecasting stock returns. In Section 2.5, neural network models will be discussed in forecasting financial problems.

2.2 Local Research on Real Estate Stocks

Liow (1997) analyzed the long term performance of Singapore 16 property stocks, and provided comprehensive evidence on the risk-return performance of Singapore property companies over an extended time period from 1975 to 1995. It concluded that property stocks performed no better than the stock market, and poorer than the market on a risk-adjusted basis. Property stocks were also found to be highly correlated with the stock market and their performance was closed tied to the property market. Moreover, property firms failed to provide ex-post inflation protection.

Liow (1998a) empirically examined the sustainable growth of Singapore property investment and development companies and the financial strategies employed by these firms in achieving financial stability and sustaining growth from 1986 to 1995. It found that the actual growth rates of many property firms are higher than their sustainable growth rates and the key financial determinants of sustainable growth for property companies are return on capital, earnings retention and debt-to equity ratio. These firms tended to rely on increasing financial leverage to sustain their high growth. However, the growth did not have a clear impact on the share price performance of the companies and shareholders' returns.

Sing (2001) presented evidence of long-run contemporaneous relationships in Singapore's property stock prices using co-integration methodology. The co-integration tests of the 20 property stocks in Singapore over a 17-year period revealed that 18 percent of the listed property stocks in Singapore established significant long-term pair-wise price convergence relationships. It was possible to predict the long-term price movement of a property stock by observing the price movement of another property stock. The weak form efficient market hypothesis was also not ruled out from the Johansen's multivariate co-integration tests, where not more than 7 out of 17 possible co-integration equations were found to be significant at 5% level. Singapore's property stock market was thus deemed to be highly, though not perfectly, efficient in the weak form.

Sing *et al.* (2002) examined whether prices of 15 sample listed property stocks in Singapore reflected their corporate fundamental values over a ten-year period from June 1989 to June 1999. Proxies for corporate fundamental values used in their study were earnings per share (EPS), dividends per share (DPS) and net asset values (NAV) of the individual property stocks listed in Singapore. It was found that the prices of only 9 of the 15 sample stocks converged in a long run with their fundamental values. The results implied that institutional investors should pay more attention to the underlying performance of stocks, in particularly the EPS and NAV, in their stock selection process.

Moreover, there are studies that examine the relationships of the stock market and property market.

Chan and Sng (1991) analyzed the overall price movement of property in the real estate market and property stocks in the securities market from 1979 to 1988, using property price and share price indices. The results showed that direct investments in real estate have higher quarterly returns and lower level of risk than investments in property stocks. They concluded that the returns are not significantly different and found that property stocks could be used as a proxy for real estate.

Ong (1994), using Structural and Vector Autoregressive Approaches, established that a contemporaneous long term relationship between property stock price index, real estate price index and risk free interest rate existed for the period of analysis from 1976 to 1993. He also concluded that current returns in property stock and the real estate returns were not dependent on the past returns, hence past returns were not a good indication of future returns.

In another study, Ong (1995) re-examined the established practice of using property stocks as a proxy for real estate investment. By applying Co-Integration testing methodology to analyze the property stock and property indices in Singapore from 1977 to 1992, the study showed that there was insufficient evidence to establish a long term, contemporaneous relationship between property stocks and real estate. The co-integration test shows clearly that although the real estate and property stock indices in Singapore are both first-difference stationary (making co-integration test possible), the linear combination between them is not integrated. The error terms from the co-integration are highly auto-correlated and non-stationary.

Ho & Cuervo (1999) looked into the dynamics of private housing prices in Singapore from the first quarter of 1985 to the fourth quarter of 1995. Employing the cointegration analysis, their paper showed that overall private housing price was cointegrated with real gross domestic product, prime lending rate and private housing starts. An error-correction mechanism was also incorporated in the estimation of changes in the overall private housing price to account for the short-run deviations from the equilibrium relationship among these variables.

Liow (1996) investigated the share prices of Singapore property companies in relation to their net asset values from 1980 to 1984. It was found that the share price of most property companies were above the book values of their net tangible assets. In testing the strength of relationships between the share price discount/premium and the property returns, there was evidence of significant co-movement between the two markets' performance.

Liow (1998b) investigated the relationship between property stock and direct property returns in the period 1975 to 1994. The results showed that property stock was highly correlated with the stock market and that property stocks returns led the property market by three to six months.

Although neural networks were studied in models to housing price valuation in Singapore, for example, Tay and Ho (1992, 1994), there are no studies thus far using neural networks to research on property stock performance. Therefore, this research is the first to use neural networks to predict the property stock returns in Singapore.

2.3 Property Stock Returns

One key issue is to examine whether real estate investment offers superior return. Focusing primarily on REITs in the US, earlier studies, such as Sagalyn (1990), concluded that REITs earned positive risk adjusted returns especially from the late 1970s to the mid-1980s. As pointed out by Titman and Warga (1986), these findings were often interpreted as evidence that real estate was a particularly good investment that investors should add to their portfolios.

However, recent studies have questioned the reported abnormal returns. In particular, Liu *et al.* (1995), in a critical review of the literature on real estate performance, suggested that superior real estate performance was an illusion arising from an omission of certain fundamental factors in the estimates of risks. They argued that any evidence that real estate continues to possess superior performance in the long run is likely to suffer from an inadequate or deficient pricing model.

Several studies have also illustrated the importance of using multiple index models instead of single index models to determine the returns of real estate related stocks. In particular, Chan *et al.* (1990) found evidence of excess real estate returns, especially in the 1980s, when a simple CAPM framework was employed. However, when the multifactor model was employed, the excess return evaporated.

Recently, research that examined the performance of real estate stock using international data was carried out. Glascock et al. (2002) used a modified version of Jensen's alpha to measure the excess returns of publicly traded real estate firms in six Asian market economies, namely, Japan, Taiwan, Hong Kong, South Korea, Singapore and Thailand. Their results showed that, except for Taiwan, real estate stocks across the other five Asian markets do not exhibit excess returns behavior. They also noted that the risk characteristics of the real estate stocks changed with market conditions although the effects were not the same across different countries.

In another study, Ling and Naranjo (2002) examined the return performance of 600 publicly trade real estate companies in 28 countries over the 1984 to 1999 time period. Based on single and multifactor specifications, they found substantial variations in mean real estate returns and standard deviations across countries. Using the standard Treynor ratio, they observed substantial variation across countries in excess real estate returns per unit of systematic risk. However, they detected little evidence of abnormal risk-adjusted returns at the country level. Their overall results indicated the existence of a strong worldwide factor in international real estate returns, as well as a highly significant country-specific factor.

Ooi and Liow (2002) investigated the performance of real estate stocks listed in seven emerging markets in Asia, namely Hong kong, Indonesia, Malaysia, Singapore, South Korea, Taiwan and Thailand. Whilst the risk-adjusted returns of real estate stocks vary across the markets and over time, they did not find any evidence of superior returns. Using panel regressions, they examined the determinants of the risk-adjusted returns at the firm level. The empirical evidence suggested that interest rates, market condition, market-to-book value, dividend yield and market diversification had significant influence on the risk-adjusted returns of real estate stocks in Asian. Firm size, leverage, and development exposure, however, did not appear to have any significant impact on the risk-adjusted returns.

2.4 Traditional Regression Techniques in Forecasting Stock Returns (OLS and Logit Regression)

As traditional techniques, OLS and logit regression are used widely in the prediction of stock returns. Here, I will only concentrate on two papers (Ou and Penman (1989), Holthausen and Larcker (1992)) which predicted stock returns using financial statement information by logit regression models. This work is related to these two previous studies.

Ou and Penman (1989) documented the existence of significant abnormal returns to a trading strategy that was based on the prediction of the sign of unexpected annual earnings-per-share (EPS), where unexpected EPS was determined from the assumption that annual EPS follow a random walk (with drift) process. Their prediction model for the sign of unexpected EPS was developed using logit, where the independent variables were traditional financial statement ratios. Ou and Penman's trading strategy took a long (short) position in the common stocks of firms where the prediction model indicated that unexpected earnings were likely to be positive (negative). They documented an average market-adjusted return over the 1973-1983 period associated with this trading strategy of 8.3% for a 12-month holding period and 14.5% for a 24-month holding period. The independent variables used by Ou and Penman were the 68 financial accounting ratios. Ou and Penman (1989, p.328) concluded, based on this result as well as other extensive empirical analyses, that '… financial statements capture fundamentals that are not reflected in prices'. Others who have examined the ability of financial ratios to earn subsequent excess returns include O'Conner (1973), Wansley *et al.* (1983) and Reingnum (1988).

Holthausen and Larcker (1992) examined the ability of accounting information to generate profitable trading strategies by developing a model to directly predict the sign of subsequent one-year excess return measures. They developed logit models, which were based on accounting ratios, to predict three different measures of 12-month excess returns which cumulated from the fourth month following the company's fiscal year-end. The three excess return metrics are: (i) market-adjusted returns, (ii) excess returns computed using the Capital Asset Pricing Model (CAPM), and (iii) size-adjusted returns. They dropped eight of the 68 ratios of Ou and Penman (1989) because there were considerable missing observations in their sample period. Their work is similar to that of Ou and

Penman (1989), but rather than basing trading strategy on a model which predicts unexpected earnings, their trading strategy was based on the prediction of excess return measures directly. The results suggested that a trading strategy based on a model which predicted excess returns directly is able to earn significant abnormal returns in the 1978-1988 period. Their overall results supported the contention of Ou and Penman that financial statement items can be combined into one summary measure to yield insights into the subsequent movement of stock prices.

2.5 Artificial Neural networks in Finance

Artificial neural networks (ANNs) are universal and highly flexible function approximators first used in the fields of cognitive science and engineering. In recent years, neural network applications in finance for such tasks as pattern recognition, classification, and time series forecasting have dramatically increased. ANNs' primary advantage over more conventional econometric techniques lies in their ability to model complex, possibly non-linear processes without assuming any prior knowledge about the underlying data-generating process (see, e.g. Hill et al., 1994; Darbellay & Slama, 2000; Balkin & Ord, 2000; Tacz, 2001). The non-linearity may take the form of a complex non-linear relationship between the independent and dependent variables, the existence of upper or lower thresholds for the influence of independent variables, or differences between forecasting up or down movements of the dependent variable. The fully flexible functional form makes them particularly suited to a financial application where non-linear patterns are clearly present but an adequate structural model is conspicuously absent. The researcher does not need to know the type of functional relationship that exists between the dependent and independent variables (Darbellay & Slama, 2000).

Numerous studies have documented the successes and failures of ANNs in forecasting time series and cross sectional financial data. A good summary of the literature is provided by Adya and Collopy (1998). Moreover, some research also gave some suggestion to improve the performance of ANNs. I will review these literatures in the following.

2.5.1 The Benefits of ANNs in Forecasting

Since construction and implementation of neural network models is considerably more difficult and time consuming than using simpler regression-type models, forecasters may want to build ANNs models only if there is a strong prior belief that additional complexity is warranted (Balkin & Ord, 2000; Darbellay & Slama, 2000). Many published studies generally showed that ANNs dominate traditional forecasting techniques, such as ordinary least squares regression, logit regression, or discriminant analysis.

Tay & Ho (1992) introduced the theory of artificial neural networks (ANN) and discussed its application to the valuation of residential apartments. They also compared the performance of the back propagation neural network (BP) model in estimating sale prices of apartments against the traditional multiple regression analysis (MRA) model. Finally, they concluded that the neural network model was an easy-to-use, black-box alternative to the MRA model.

Refenes *et al.* (1993) tested ANNs in the domain of stock ranking. Comparisons with multiple regression indicated that the proposed network gave better fitness on the test data over multiple regression by an order of magnitude. The network outperformed regression on the validation sample by an average of 36%.

In a study of bankruptcy classification, Udo (1993) reported that ANNs performed, as well as, or only slightly better than, multiple regression although this conclusion was not confirmed by statistical tests.

Wilson and Sharda (1994) and Tam and Kiang (1990, 1992) developed ANNs for bankruptcy classification. Wilson and Sharda (1994) reported that although ANNs performed better than discriminant analyses, the differences were not always significant. The authors trained and tested the network using three sample compositions: 50% each of bankrupt and non-bankrupt firms, 80% of non-bankrupt and 20% of bankrupt firms, and 90% of non-bankrupt and 10% of bankrupt firms. Each sample was tested on a 50/50, 80/20, and 90/10 training set yielding a total of nine comparisons. The ANNs outperformed discriminant analysis on all but one sample combination for which performance of the methods was not statistically different.

Tam and Kiang (1990, 1992) compared the performance of ANNs with multiple alternatives: regression, discriminant analysis, logistic, k Nearest Neighbour, and ID3. They reported that the ANNs outperformed all comparative methods when data from one year prior to bankruptcy was used to train the network. In instances where data for two years before bankruptcy was used to train, discriminant analysis outperformed ANNs. In both instances, an ANN with one hidden layer outperformed a linear network with no hidden layers.

In a similar domain, Salchenberger *et al.* (1992) and Coats and Fant (1992) used ANNs to classify a financial institution as failed or not. Salchenberger et al. (1992) compared the performance of ANNs with logit models. The network performed better than logit models in most instances where the training and testing sample had equal representation of failed or non-failed institutions. The ANNs outperformed logit models in a diluted sample where about 18% of the sample was comprised of failed institutions' data.

Coats and Fant (1993) used the Cascade Correlation algorithm for predicting financial distress. Comparative assessments were made with discriminant analysis. The ANNs outperformed discriminant analysis on samples with large percentages of distressed firms, but failed to do so on those with a more equal mix of distressed and non-distressed firms.

Tay and Ho (1992, 1994) introduced the theory of artificial neural networks (ANN) and discussed its application to the valuation of residential apartments. They also compared the performance of the back propagation neural network (BP) model in estimating sale prices of apartments against the traditional multiple regression analysis (MRA) model. Finally, they concluded that the neural network model was an easy-to-use, black-box alternative to the MRA model.

One previous research, which bears close resemblance to this work, was Olson and Mossman (2002). They compared neural network forecasts of one-year-ahead Canadian stock returns with the forecasts obtained using ordinary least squares (OLS) and logistic regression (logit) techniques. Their results indicated that back propagation neural networks, which considered non-linear relationships between input and output variables, outperformed the best regression alternatives for both point estimation and in classifying firms expected to have either high or low returns. The superiority of the neural network models translated into greater profitability using various trading rules. Classification models out performed point estimation models, but four to eight output categories appeared to give better results for both logit and neural network models than either binary classification models or models with 16 classification categories.

This current research can be differentiated from the Olson and Mossman (2002). Firstly, I employ only 13 Singapore property stocks data and do not test the profitability of different methods under various trading rules. Secondly, I also include The Monte Carlo neural network method to improve the performance of ANNs in forecasting.

2.5.2 Some Failure in ANNs' Forecasting

However, some researchers questioned whether ANNs had been over sold as a miracle forecasting technique and a subsequent strand of literature documents that ANNs often under perform naive financial models, such as the random walk or a buy and hold strategy.

Callen *et al.* (1996) used an ANN model to forecast accounting earnings for a sample of 296 corporations trading on the New York stock exchange. The resulting forecast errors were shown to be significantly larger (smaller) than those generated by the parsimonious Brown-Rozelf and Griffin-Watts (Foster) linear time series models. Their study confirmed the conjecture by Chatfield (1993) that neural network models are context sensitive. In particular, their study shows that neural network models are not necessarily superior to linear time series models even when the data are financial seasonal and non-linear.

Church & Curram (1996) compared the performance of ANNs and econometric model in predicting the decline in the growth rate of consumers' expenditure in the late 1980s. It is found that the neural network models describe the decline in the growth of consumption since the late 1980s as well as, but no better than, the econometric specifications included in the exercise, and are shown to be robust when faced with a small number of data points.

Episcopos and Davis (1995) compared the forecasting performance of EGARCH-M and neural network models for predicting daily US dollar foreign exchange series. They found that both outperform the random walk, but neither was consistently better than the other.

2.5.3 Some Suggestions for the Improvement of ANNs' Forecasting

Some research also provided suggestions to improve the performance of ANNs on finance forecasting. For example, Hill et al. (1994) suggest that ANNs are likely to work best for high frequency financial data and Balkin and Ord (2000) also stress the importance of a long time series of observations to insure optimal results from training neural networks.

Tacz (2001) found that neural networks outperform naive models in forecasting Canadian GDP growth at time horizons of 1 year, but not over shorter intervals. A possible reason is that there is considerable noise in the quarterly observations, so that only longer-run forecast horizons can pick up the non-linear dependence in the data.

Qi (2001) also finds that nonlinearities help in forecasting US GDP and recessions because business cycles may be asymmetric between up and down cycles. For many financial forecasting problems, classification models work better than point prediction (Leung *et al.* 2000; Brooks, 1997) and that contention will be tested in this paper.

Finally, Brooks (1997) and Qi (2001) have pointed out the continually changing nature of financial relationships so that ANNs are more likely to out perform traditional techniques when the input data is kept as current as possible. This can be done by recursive modeling, meaning that the researcher adds new observations and drops the oldest observations each time a new time series forecast is made.

In conclusion, based upon this review of the literature, ANNs are expected to perform better than traditional regression techniques in this forecasting situation, but neither approach dominates the other. Therefore, this research tries to investigate the borderline at which neural networks begin to out perform traditional regression-based forecasting techniques.

Chapter 3: Stepwise OLS Regression and Logit Regression Models Forecasting

3.1 Introduction

Traditionally, OLS or logit regressions are used widely in prediction of stock returns (e.g. Ou and Penman (1989), Holthausen and Larcker (1992), Brockman *et al.* (1997), Olson and Mossman (2002)). This chapter will explain the stepwise regression and logit regression models and methodology used to obtain OLS and logit forecasts.

This chapter is organized as follows. Section 3.2 reviews the basic concepts and limitations of stepwise regression models. For classification problems, Section 3.3 introduces logit regression models. In Section 3.4, stepwise OLS and logit regression models are fitted to the property stock forecast using annual accounting ratios.

3.2 Stepwise Regression Models

3.2.1 Basic Concepts in Stepwise Regression Models

Since there are a larger number of independent variables in this research, I use Stepwise regression models to estimate. I will illustrate this method here. With a lot of independent variables, the computer is programmed to introduce independent variables one at a time in order to determine the sequence that makes R^2 increase the fastest. Only the independent variables that are the most powerful by this criterion are retained in the final model.

An important assumption behind the method is that some input variables in a multiple regressions do not have an important explanatory effect on the response. If this assumption is true, then it is a convenient simplification to keep only the statistically significant terms in the model.

The basic procedure involves (1) identifying an initial model, (2) iteratively "stepping," that is, repeatedly altering the model at the previous step by adding or removing a predictor variable in accordance with the "stepping criteria," and (3) terminating the search when stepping is no longer possible given the stepping criteria, or when a specified maximum number of steps has been reached. The following topics provide details on the use of stepwise model-building procedures.

The Initial Model in Stepwise Regressions

The initial model is designated the model at Step 0. The initial model always includes the regression intercept (unless the No intercept option has been specified.). For the backward stepwise and backward removal methods, the initial model also includes all independent variables specified to be included in the design for the analysis. The initial model for these methods is therefore the whole model.

For the forward stepwise and forward entry methods, the initial model always includes the regression intercept (unless the No intercept option has been specified.). The initial model may also include 1 or more independent variables specified to be forced into the model. If j is the number of independent variables specified to be forced into the model, the first j independent variables specified to be included in the design are entered in the model at Step 0. Any such independent variables that are not eligible would be removed from the model during subsequent Steps.

Independent variables may also be specified to be forced into the model when the backward stepwise and backward removal methods are used. As in the forward stepwise and forward entry methods, any such independent variables that are not eligible would be removed from the model during subsequent Steps.

The Forward Entry Method

The forward entry method is a simple model-building procedure. At each Step after Step 0, the entry statistic is computed for each independent variable eligible for entry in the model. If no independent variable has a value on the entry statistic which exceeds the specified critical value for model entry, then stepping is terminated, otherwise the independent variable with the largest value on the entry statistic is entered in the model. Stepping is also terminated if the maximum number of steps is reached.

The Backward Removal Method

The backward removal method is also a simple model-building procedure. At each Step after Step 0, the removal statistic is computed for each independent variable eligible to be removed from the model. If no independent variable has a value on the removal statistic which is less than the critical value for removal from the model, then stepping is terminated, otherwise the independent variables with the smallest value on the removal statistic is removed from the model. Stepping is also terminated if the maximum number of steps is reached.

The Forward Stepwise Method

The forward stepwise method employs a combination of the procedures used in the forward entry and backward removal methods. At Step 1 the procedures for forward entry are performed. At any subsequent step where 2 or more independent variables have been selected for entry into the model, forward entry is performed if possible, and backward removal is performed if possible, until procedure can be performed and stepping is terminated. Stepping is also terminated if the maximum number of steps is reached.

The Backward Stepwise Method

The backward stepwise method employs a combination of the procedures used in the forward entry and backward removal methods. At Step 1 the procedures for backward removal are performed. At any subsequent step, where 2 or more independent variables have been selected for entry into the model, forward entry is performed or backward removal is performed, until procedure can be performed and stepping is terminated. Stepping is also terminated if the maximum number of steps is reached.

Entry and Removal Criteria

Either critical F values or critical p values can be specified to be used to control entry and removal of independent variables from the model. If p values are specified, the actual values used to control entry or removal of independent variables from the model is 1 minus the specified p values. The critical value for model entry must exceed the critical value for removal from the model. A maximum number of steps can also be specified. If not previously terminated, the stepping stops when the specified maximum number of steps is reached.

3.2.2 Some Limitations of Stepwise Regression Models

Although stepwise regressions (forward or backward) maybe useful for exploring the data, it is a brute-force technique. In letting the computer rather than theory dictates the form of the model, one can end up with theoretically important variables dropped from the model. In general, it seems preferable that model specification be guided by theory instead of some mechanical rule.

Stepwise regressions are additionally frowned on by many statisticians because it violates the logic of hypothesis testing. Instead of testing a theory, one typically concocts an *ad hoc* explanation that seems consistent with a model arrived at mechanically by stepwise regressions. Statistical tests are then inappropriate and invalid. One common problem in multiple regression analysis is the multi-collinearity of the input variables. The input variables may be as correlated with each other as they are with the response. If this is the case, the presence of one input variable in the model may mask the effect of another input. Stepwise regressions used as a canned procedure is a dangerous tool because the resulting model may include different variables depending on the choice of starting model and inclusion strategy.

However, the stepwise regression model is viewed as an effective model if some of the pitfalls that exist in stepwise regressions are avoided. To avoid these pitfalls, Foster and Stine (2002) provided three modifications to standard regressions are required: (1) use interactions to capture non-linearities, (2) use Bonferroni to pick variables to include, and (3) use the sandwich estimator to get robust standard errors. The authors also explained what each of these three modifications is and why they are necessary. If all three of these are done, they end up with a procedure that can be used on almost any data set.

3.3 Logit Regression Models

Logistic regressions, more commonly called logit regressions, are used when the dependent variable is dichotomous (i.e., binary or 0-1). The independent variables may be quantitative, categorical, or a mixture of the two.

3.3.1 The Logistic Function

The basic form of the logistic function is

$$P = \frac{1}{1+e^{-z}} \tag{3.1}$$

where Z is the independent variable and e is the base of the natural logarithm, equal to 2.718828… If I view (3.1) as an estimated model, so the P is an estimated probability. A property of the logistic function, as specified by (3.1), is that when Z becomes infinitely negative, e^{-z} becomes infinitely large, so that P approaches 0. When Z becomes infinitely positive, e^{-z} becomes infinitely small, so that P approaches unity. When $Z = 0$, $e^{-z} = 1$, so that $P = 0.5$.

If the numerator and denominator of the right side of (3.1) are multiplied by e^{z}, the logistic function in (3.1) can be written alternatively as

$$P = \frac{e^{z}}{1+e^{z}} \tag{3.2}$$

3.3.2 The Multivariate Logistic Function

Suppose that Z is a linear function of a set of independent variables:

$$Z_i = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + ... + \beta_n X_n \tag{3.3}$$

Note that Z is not a response variable in this equation. This express can be substituted for Z in the Eq. (3.1):

$$P = \frac{1}{1 + e^{-(\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \dots + \beta_n X_n)}} \tag{3.4}$$

All the basic properties of the logistic function are preserved when this substitution is done. The function still range between 0 and 1 and achieves its maximum rate of change, with respect of change in any of the X_i, at $P = .5$.

3.3.3 The Odds and the Logit of P

The logit of P is derived from the logistic function (3.1). From (3.1) it follows that

$$1 - P = 1 - \frac{1}{1 + e^{-z}} = \frac{e^{-z}}{1 + e^{-z}} \tag{3.5}$$

Dividing (3.1) by (3.5) yields

$$\frac{P}{1 - P} = e^{Z} \tag{3.6}$$

Taking the natural logarithm (base e) of both sides of (3.6), I obtain

$$\log \frac{P}{1 - P} = Z \tag{3.7}$$

The quantity $P/(1-P)$ is called the *odds*, and the quantity $\log(P/(1-P))$ is called the *log odds* or the *logit of* P. The definition of the odds corresponds to everyday usage. For example, one speaks the odds of winning a gamble on a horse race as, say, “75:25”, meaning .75/(1-.75) or, equivalently, 75/(100-75). Alternatively, one speaks of “three-to-one” odds, which is the same as 75:25.

With these definitions, and with the expression in (3.3) substituted for Z, (3.7) can be written alternatively as

$$\log \frac{P}{1-P} = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + ... + \beta_n X_n \tag{3.8}$$

$$\log P = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + ... + \beta_n X \tag{3.9}$$

Eq. (3.8) and (3.9) are in the familiar form of an ordinary multiple regression equation. This is advantageous, because some of the statistical tools developed for multiple regressions can be applied to logit regression too.

3.3.4 Fitting the Logit Regression Models

To fit the logit regression models, I use the method of maximum likelihood. To illustrate this method, let us consider a very simple model:

$$LogitP = a + bX \tag{3.10}$$

which can be written alternatively as

$$P = \frac{1}{1+e^{-(a+bX)}} \tag{3.11}$$

Suppose that this original dependent variable is Y, which for individuals is either 0 or 1. Denote the first individual by subscript 1. For this individual $(X,Y) = (X_1, Y_1)$, then, I have that

$$P_1 = \frac{1}{1+e^{-(a+bX_1)}} \tag{3.12}$$

Similarly, for individual 2,

$$P_2 = \frac{1}{1+e^{-(a+bX_2)}} \tag{3.13}$$

and so on for the remaining individuals in the sample.

Let us consider individual 1 in more detail. I have from (3.12) that

$$\Pr(Y_1 = 1) = P_1 \tag{3.14}$$

where $\Pr$ denotes "probability that". Therefore,

$$\Pr(Y_1 = 0) = 1 - P_1 \tag{3.15}$$

I can combine (3.14) and (3.15) into one formula:

$$\Pr(Y_1) = P_1^{Y_1}(1-P_1)^{(1-Y_1)} \tag{3.16}$$

Similarly

$$\Pr(Y_2) = P_2^{Y_2}(1-P_2)^{(1-Y_2)} \tag{3.17}$$

And so on up to $\Pr(Y_n)$, where n denotes the number of sample cases.

If I assume simple random sampling, these n probabilities are independent. Then the probability, or likelihood, of observing the particular sample data is

$$\begin{aligned}\Pr(Y_1, Y_2, Y_3, ..., Y_n) &= \Pr(Y_1)\Pr(Y_2)\Pr(Y_3)...\Pr(Y_n)\\ &= \prod_{i=1}^{n} \Pr(Y_i)\\ &= \prod P_i^{Y_i}(1-P_i)^{(1-Y_i)}\\ &= \prod [\frac{1}{1+e^{-(a+bX_i)}}]^{Y_i}[1-\frac{1}{1+e^{-(a+bX_i)}}]^{1-Y_i}\\ &= L(a,b)\end{aligned} \tag{3.18}$$

where $\prod$ is the product symbol (analogous to the summation symbol $\sum$), and where $L(a,b)$ indicates that the likelihood function, L, is a function of the unknown parameters,

a and b. Note that X_i and Y_i are observed data and therefore constants in the equation, not unknown parameters.

It is also useful to derive a formula for $\log L$:

$$\log L(a,b) = \sum \{Y_i \log[\frac{1}{1+e^{-(a+bX_i)}}] + (1-Y_i)\log[\frac{1}{1+e^{-(a+bX_i)}}]\} \tag{3.19}$$

In getting from (3.18) to (3.19), I make use of the rules $\log xy = \log x + \log y$ and $\log x^y = y \log x$.

I wish to find the values of a and b that maximize $L(a,b)$. Because $\log L$ is a monotonically function of L, maximizing L is equivalent to maximize $\log L$. I maximize $\log L$ by taking the partial derivative of $\log L$ first with respect to a and second with respect to b, and then equating each derivative separately to zero, yielding two equations in two unknowns, a and b. These equations are then solved for a and b by numerical methods.

3.3.5 Goodness of Fit

In multiple regressions analyses, the traditional indicator of goodness of fit is R^2, which measures the proportion of variation in the dependent variable that is explained by the independent variables. In the case of logit regressions, one could also calculate the proportion of variation in the dependent variable that is explained by the independent variables, but in this case it is impossible for the observed values of the dependent variable, which are either 0 or 1, to conform exactly to the fitted values of P. The maximum value of this proportion depends on the mean and variance of P.

An alternative measure of goodness of fit may be derived from likelihood statistic. Let L_0 denote the likelihood for the fitted intercept model, and let L_1 denote the likelihood for the fitted test model. Define pseudo-R^2 as

$$pseudo - R^2 = \frac{1-(L_0 / L_1)^{2/n}}{1-L_0^{2/n}} \tag{3.20}$$

where n is the sample size. The minimum value of this quantity is zero when the fit is as bad as it can be (when $L_1 = L_0$), and the maximum value is one when the fit is as good as it can be (when $L_1 = 1$). This definition of pseudo-R^2 was suggested by Cragg and Uhler (1970). Unfortunately there is no formal significance test that utilizes this measure.

Another definition suggested by McFadden (1974) is simply

$$pseudo - R^2 = 1-(\log L_0 / \log L_1)^{2/n} \tag{3.21}$$

Eviews presents this measure, denoted by R^2_{McF}. Like R^2, R^2_{McF} also range between 1 and 0. Another comparatively simple measure of goodness of fit is the count R^2, which is defined as

$$\text{count R}^2 = \frac{\text{number of correct predictions}}{\text{total number of observations}} \tag{3.22}$$

There are several difficulties with these measures of pseudo-R^2, some of which have already been mentioned. First, there are several different measures available, which can give rather different numerical results when applied to the same data set. Second, there is little basis for choosing one measure over the other. Third, statistical tests that utilize pseudo-R^2 are not available for any of the measures. For these reasons, many authors do not present values of pseudo-R^2 when reporting results form logit regression analyses. Therefore, this research also does not provide pseudo-R^2.

3.4 Forecasting Using Stepwise OLS Regression Models and Stepwise Logit Model

In this study, six years of data are used to estimate the parameters of the within-sample OLS model and logit model. The independent variables of both models are 52 accounting ratios and financial variables calculated at the beginning of each fiscal year with the 3-month lag as discussed in Chapter 1. The dependent variable in OLS models is a firm's market-adjusted abnormal return, which is measured as the firm's actual return over the next fiscal year minus the return on the portfolio of all stocks in the sample during this period. For logit model, the dependent variable is the probability of a firm having an abnormal return above or below the median of the sample over the upcoming fiscal year.

The estimation procedures for OLS and logit models are similar. First, the dependent variable is estimated using step-wise OLS or logit regressions, depending upon the type of forecast desired (point estimation or classification). All fundamental analysis variables initially enter the regression equation. A parsimonious subset of independent variables is selected in most cases by simply eliminating any explanatory variables with *P*-values greater than 5% for the 6-year estimation period. The exceptions are best illustrated by some examples. In some estimation periods, return on sales and return on equity have coefficients of similar magnitude and significance, but they have opposite signs. This indicates co-linearity, so one of the variables is dropped and step-wise regression is re-applied to the remaining variables. Also, based upon forecasting results in the validation year, the 5% criterion is adjusted downward to 4 or 3% if the step-wise procedure originally selects more than eight independent variables. If the above stepwise procedure selects only one or two variables, the cut-off criterion is increased to 10% to include more independent variables. In general, models with three to five independent variables tend to give the best forecasts in the validation period.

3.4.1 Forecasting Using Stepwise OLS Regression Models

The model uses stepwise OLS regressions to select independent variables and estimate abnormal returns over 6-year periods. Abnormal returns for each company observation are:

$$R_{it} = r_{it} - r_{mt} = \alpha + \sum_{j=1}^{n} \beta_j V_{jt-1} + \varepsilon_{it} \tag{3.23}$$

where R_{it} is the market-adjusted abnormal return for stock *i* in fiscal year *t* calculated as the actual return for stock *i* (r_{it}) minus the mean return (r_{mt}) as it is measured by the average return for all property stocks in this sample in period *t*, β_j are the estimated OLS coefficients, α is a constant, and ε_{it} are error terms for each regression. The V_{jt-1} represent the *n* statistically significant inputs selected from the all potential beginning-of-the-fiscal-year input variables denoted by the time subscript $t-1$ to indicate that they are known at the end of the previous period and not updated during fiscal year t. One-year ahead abnormal returns (R_{t+1}) are forecasted by substituting the next year's beginning fiscal year V_{jt} ratios into Eq.(3.23), while keeping α and β_j the same as in the former period. This procedure is used to forecast the returns for year 2000 and 2001.

3.4.2 Forecasting Using Stepwise Logit Regression Models

The second model of this research is stepwise logit regressions, where the probability (P_{it}) that firm i in year t will have an abnormal return above that of the median firm is given by:

$$\log(P_{it} / 1 - P_{it}) = \alpha + \sum_{j=1}^{n} \beta_j V_{jt-1} + \varepsilon_{it} \tag{3.24}$$

Notation and methodology are the same as in Eq. (3.23). Any number of classification categories can be used, such as four, eight, and 16 different ranking classes, but this study only considers two different ranking classes based upon the magnitude of each stock's annual abnormal returns. According to Eq.(3.4),

$$P_{it} = \frac{1}{1 + e^{-z_{it}}}$$

where $z_{it} = \alpha + \sum_{j=1}^{n} \beta_j V_{jt-1} + \varepsilon_{it}$, then, I could calculate the probability (P_{it}). Similarly, one-year ahead probability ($P_{i(t+1)}$) that firm i will have an abnormal returns are forecasted by substituting the next year's beginning fiscal year V_{jt} ratios into Eq.(3.24), while keeping α and β_j the same as in the former period.

3. 5 Summary

This chapter reviews the basic concepts and limitations of stepwise regression models and logit regression models. And then, stepwise OLS regression models and logit regression models are fitted to the property stock forecast using annual accounting ratios.

Chapter 4 : Neural Networks in Forecasting Stock Returns

4.1 Introduction

Artificial neural networks (ANN) have attracted much interest in financial engineering as the cost of computing technology has declined. Typical applications in finance include portfolio selection /diversification, risk rating of mortgages and fixed income investments, index construction, simulation of market behavior, identification of economic explanatory variables, and economy forecasting. Back propagation (BP) neural networks are the most common type of neural networks in time series forecasting. So, this chapter will focus on the application of BP neural networks.

This chapter is organized as follows. Section 4.2 reviews the basic concepts and strengths and weaknesses of neural networks. Since most neural networks used in economic analyses are BP neural networks, Section 4.3 introduce the methods of the BP neural networks building. In Section 4.4, the BP neural networks are fitted to the property stock forecast using annual accounting ratios.

4.2 Basic Concepts and Strengths & Weakness of ANNs

Before looking at the application process, let us establish some basic concepts. ANNs are defined as an information processing technology inspired by studies of the brain and nervous system (Medsker *et al.* 1996). It is a data model that has the ability to learn to recognize complicated patterns without being programmed with specific preconceived rules.

4.2.1 Some Basic Concepts

In building an artificial neural network, the builder must be familiar with some concepts. The key concepts are as follow:

Processing Elements:

An ANN is composed of artificial neurons (neurodes, or nodes); these are processing elements. Each of the neurons receives input(s), processes the input(s), and delivers a single output(s). This process is shown in Fig. 4.1. The inputs can be raw data or output of other processing elements. The outputs can be the final product, or it can be an input to another neurons. Between inputs and outputs, weights express the relative importance of each input to a processing element. Through the repeated adjustments of weights, the network learns.

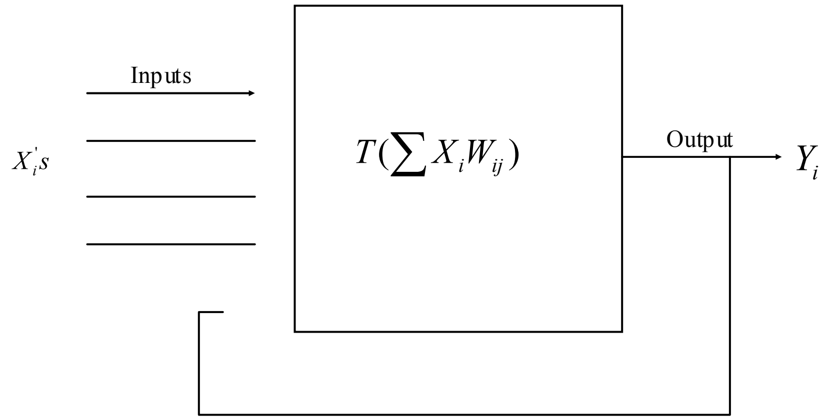

Fig.4.1 A Neural Processing Element

Network Structure:

Similar to biological networks, an ANN can be organized in several different ways (topologies); that is, the neurons can be interconnected in different ways. Three representative architectures are associative memory systems, hidden layer and double layer. In this study, I use hidden layer structure.

Learning

An ANN learns from its mistakes. The usual process of learning (or training) involves three tasks: compute outputs, compare outputs with desired answers, and adjust the weights and repeat the process. The learning process usually starts by setting the weights randomly. The difference between the actual output and the desired output is called error. The objective is to minimize error, or even better to reduce it to zero. The reduction of error is done by incrementally changing the weights.

Summation Function

The summation function determines the weighted average of all the input elements to each processing element. A summation function multiplies the input values (Xs) by the weights (Ws) and totals them together for a weighted sum, Y. For N inputs i into one processing element j, I have

$$Y_i = \sum_{i=1}^{n} X_i W_{ij} \tag{4.1}$$

Transformation Function:

The summation function computes the internal stimulation, or activation level, of the neurons. Based on this level, the neuron may or may not produce an output. The relationship between the internal activation level and the output may be linear or nonlinear. Such relationships are expressed by a transformation function. Therefore, transformation functions are mathematical formulas that determine the output of a processing neuron. They are also referred to as transfer, squashing, activation, or threshold functions. The purpose of the transfer function is to prevent outputs from reaching very large values which can 'paralyze' neural networks and thereby inhibit training. Although there are some kinds of transformation functions such as the hyperbolic tangent, step, ramping and arc tan, the majority of current neural network models use the sigmoid (S-shaped) function,

$$Y_t = \frac{1}{1+e^{-y}} \tag{4.2}$$

where Y_t is the transformed value of Y.

Learning algorithms:

An important consideration in ANN is the appropriate use of algorithms for learning (or training). Such algorithms are called learning algorithms (or paradigms), and more than a hundred of them are known. The taxonomy of these algorithms has been proposed by Lippman (1989), who distinguishes between two major categories based on the input format: binary-valued input or continuous-valued input. Each of these can be further divided into two categories: supervised learning and unsupervised learning.

Supervised learning uses a set of inputs for which the appropriate (desired) output are known. In unsupervised learning, only input stimuli are shown to the networks. The network organizes itself internally, so that each hidden processing element responds strategically to a different set of input stimuli.

4.2.2 ANN Strengths and Weaknesses

Neural Network technology has significant advantages over traditional rule in some applications. Neural networks are less sensitive to error term assumptions and they can tolerate noise, chaotic components, and heavy tails better than most other methods (Master, 1993). Other advantages include greater fault tolerance. Since there are many processing nodes, each with primarily local connections, damage to a few nodes or links does not bring the system to a halt. Moreover, neural networks have good adaptability compared to expert systems due to the large number of interconnected processing elements that can be ‘trained’ to learn new patterns (Lippman, 1999). By learning through interaction with their environment, ANNs are particularly well suited for pattern recognition.

However, neural networks have their own weaknesses. Since the scope of training is always to some extent limited by economics and time, networks that contradict accepted theory run the risk of lacking generality, functioning well only on data with a structure similar to that of the training set. Furthermore, some neural network systems lack explanation facilities. Justifications for results are difficult to obtain because the connection weights do not usually have obvious interpretations. Also, most neural networks can not guarantee an optimal solution to a problem, a completely certain solution, or sometimes even lack repeatability with the same input data.

4.3 Back propagation Neural networks Building

The BP neural network is used in this study since it is the most widely used in financial time series forecasting and it is the most common type of neural network in time series forecasting. Also, the BP network is the most common multi-layer network estimated to be used in 80% of all applications (Caudill, 1992). Other neural networks less common in time series forecasting include recurrent networks, probabilistic networks and fuzzy neural networks. Hornik et al. (1989) showed that the standard BP network using an arbitrary transfer function can approximate any measurable function in a very precise and satisfactory manner, if a sufficient number of hidden neurons are used.

Hecht-Nielsen (1989) also demonstrated that a three-layer BP network can approximate any continuous mapping.

Therefore, the objective of this part is to provide the overview of a step by step methodology to design a neural network for forecasting financial time series data. First, the architecture of a BP neural network is briefly discussed. This is followed by an explanation of an eight-step procedure for designing a neural network including a discussion of tradeoffs in parameter selection and some common pitfalls.

4.3.1 Architecture of a BP Neural networks

BP neural networks consist of a collection of inputs and processing units known as neurons. The neurons in each layer are fully interconnected by connection strengths called weights which, along with the network architecture, store the knowledge of a trained network. In addition to the processing neurons, there is a bias neuron connected to each processing unit in the hidden and output layers. The bias neuron has a value of positive one and serves a similar purpose as the intercept in regression models. The neurons and bias terms are arranged into layers; an input layer, one or more hidden layers, and an output layer. The number of hidden layers and neurons within each layer can vary depending on the size and nature of the data set.

BP networks are a class of feed forward neural networks with supervised learning rules. Feed forward refers to the direction of information flow from the input to the output layer. Inputs are passed through the neural networks once to determine the output. Supervised learning is the process of comparing each of the network's forecasts with the known correct answer and adjusting the weights based on the resulting forecast error to minimize the error function.

Neural networks are similar to linear and non-linear least squares regression and can be viewed as an alternative statistical approach to solving the least squares to minimize the sum of squared errors. The bias term is analogous to the intercept term in a regression equation. The number of input neurons is equal to the number of independent variables while the output neuron(s) represent the dependent variables(s). Linear regression models may be viewed as a feed forward neural network with no hidden layers and one output neurons to the single output neuron are analogous to the coefficients in a linear least squares regression. Networks with one hidden layer resemble nonlinear regression models. The weights represent regression curve parameters.

4.3.2 Steps in Designing a Neural Network Forecasting Model

Kaastra and Boyd (1996) presented an eight-step design methodology, including variable selection, data collection, data preprocessing, training, testing and validation sets, neural network paradigm, evaluating criteria, neural network training and implement. In this study I follow their suggestions to build the neural networks. The procedure is usually not a single-pass one, but may require visiting previous steps especially between training and variable selection.

Step 1: Variable Selection

Success in designing a neural network depends on a clear understanding of the problem. Knowing which input variables are important in the market being forecasted is critical. However, economic theory can help in choosing variables which are likely important predictors and the very reason for relying on a neural network is for its powerful ability to detect complex nonlinear relationships among a number of different variables. At this point in the design process, the concern is about the raw data from which a variety of indicators will be developed. These indicators will form the actual inputs to neural networks. In this study, following the Ou and Penman (1989) methodology, the data set of about 50 annual accounting ratios and financial variables are constructed for each firm during each of the 10 years in this sample.

Step 2: Data Collection

The researcher must consider not only cost and availability when collecting data for the variables chosen in the first step, but also the quality of the data. All data should be checked for errors by examining day-to-day changes, ranges, logical consistency (e.g. high greater than or equal to close, open greater or equal to low) and missing observations. Missing observations, which often exist, can be handled in a number of ways. All missing observations can be dropped or a second option is to assume that the missing observations remain the same by interpolating or averaging from nearby values.

Dedicating an input neuron to the missing observations by coding it as a one if missing and zero otherwise is also often done.

Step 3: Data Preprocessing

Data preprocessing refers to analyzing and transforming the input and output variables to minimize noise, highlight important relationships, detect trends, and flatten the distribution of the variable to assist the neural network in learning the relevant patterns. Since neural networks are pattern matchers, the representation of the data is critical in designing a successful network. The data preprocessing for presenting to the neural network needs for the dataset to be stationary although normality is not imperative. The input and output variables for which the data was collected are rarely fed into the network in a raw form. At the very least, the raw data must be scaled between the upper and lower bounds of the transfer functions (usually between zero and one or negative one and one).

Two of the most common data transformations in both traditional and neural network forecasting are first differencing and taking the natural log of a variable. First differencing, or using changes in a variable, can be used to remove a linear trend from the data. Logarithmic transformation is useful for data which can take on both small and large values and is characterized by an extended right hand tail distribution. Another popular data transformation is to use ratios of the input variables. Ratios highlight important relationships (e.g. hog/corn, financial statement ratios) while at the same time conserving degrees of freedom because fewer input neurons are required to code the independent variables. In this study, I transform the raw data into various accounting ratios as the independent variables.

Step 4: Training, Testing and Validation Sets

A common practice is to divide the time series into three distinct sets called training, testing and validation (out-of sample) sets. The training set is the largest set and is used by the neural networks to learn the patterns present in the data. The testing set, ranging in size from 10% to 30% of the training set, is used to evaluate the generalization ability of a supposedly trained network. The researcher would select the network(s) which perform best on the testing set. A final check on the performance of the trained networks is made using the validation set. The size of the validation set chosen must strike a balance between obtaining a sufficient sample size to evaluate a trained network and having enough remaining observations for both training and testing. The validation set should consist of the most recent contiguous observations.

It is recommended that the training and testing sets be scaled together since the purpose of the testing set is to determine the ability of the network to generalize. However, by no means, should the validation set be scaled with either the training or testing sets since this biases the integrity of the validation set as a final and independent check on the neural networks. In actual use, the researcher has no way of knowing the exact range of future values, but only has a reasonable estimate based on the range of the training and /or testing sets. In this study, training and testing sets are data from 1993 to 1998 and the validation set is data of 1999.

Step 5: Neural Network Paradigms

There are an infinite number of ways to construct a neural network. Neurodynamics and architecture are two terms used to describe the way in which a neural network is organized. The combination of neurodynamics and architecture define the neural network's paradigm. Neurodynamics describe the properties of an individual neuron such as its transformation function and how the inputs are combined. A neural network's architecture defines its structure including the number of neurons in each layer and the number and type of interconnections.

The number of input neurons is one of the easiest parameters to select once the independent variables have been preprocessed because each independent variable is represented by its own input neuron. This section will address the selection of the number of hidden layers, hidden layer neurons, output neurons and the transfer functions.

The hidden layer(s) provide the network with its ability to generalize. In theory, a neural network with one hidden layer with a sufficient number of hidden neurons is capable of approximating any continuous function. In practice, neural networks with one and occasionally two hidden layers are widely used and have performed very well. Increasing the number of hidden layers also increases computation time and the danger of over fitting which leads to poor out-of-sample degrees of freedom. In other words, it has relatively few observations in relation to its parameters and therefore it is able to memorize individual points rather than learn the general patterns. Therefore, it is recommended that all neural networks should start with preferably one or at most two hidden layers. If a four-layer neural network (i.e. two hidden layers) proves unsatisfactory after having tested multiple hidden neurons using a reasonable number of randomly selected starting weights, then the research should modify the input variables a number of times before adding a third hidden layer. Both theory and virtually all empirical work to date suggest that networks with more than four layers will not improve the results (Kaastra & Boyd, 1996).

Despite its importance, there is no ‘magic’ formula for selecting the optimum number of hidden neurons. Therefore, researchers fall back on experimentation. However, some rules of thumb have been advanced. Katz (1992) indicated that an optimal number of hidden neurons will generally be found between one-half to three times the number of input neurons. Ersoy (1990) proposed doubling the number of hidden neurons until the network’s performance on the testing set deteriorates. Selecting the ‘best’ number of hidden neurons involves experimentation. Three methods often used are the fixed, constructive and destructive. Regardless of the method used to select the range of hidden neurons to be tested, the rule is to always select the network that performs best on the testing set with the least number of hidden neurons. When testing a range of hidden neurons it is important to keep all other parameters constant. Changing any parameter in effect creates a new neural network with a potentially different error surface which would needlessly complicate the selection of the optimum number of hidden neurons.

Deciding on the number of output neurons is somewhat more straightforward since there are compelling reasons to always use only one output neuron. Neural networks with multiple outputs, especially if these outputs are widely spaced, will produce inferior results as compared to a network with a single output (Master, 1993). A neural network trains by choosing weights such that the average error over all output neurons is minimized. For example, a neural network attempting to forecast one month ahead and six month ahead cattle future prices will concentrate most of its effort on reducing the forecast with the largest error which is likely the six-month forecast. As a result, a relatively large improvement in the one-month forecast will not be made if it increases the absolute error of the six-month forecast by an amount greater than the absolute improvement of the one-month forecast. The solution is to have the neural networks specialize by using separate networks for each forecast. Specialization also makes the trial and error design procedure somewhat simpler since each neural network is smaller and fewer parameters need to be changed to fine tune the final model.

Linear transformation functions are not useful for nonlinear mapping and classification. Levich and Thomas (1993) and Kao and Ma (1992) found that financial markets are nonlinear and have memory suggesting that nonlinear transformation functions are more appropriate. Transformation functions such as the sigmoid are commonly used for time series data because they are nonlinear and continuously differentiable which are desirable properties for network learning. In this work, the transformation function applied is sigmoid.

Step 6: Evaluation Criteria

The most common error function minimized in neural networks is the sum of squared errors. Other error functions offered by software vendors include least absolute deviations, least fourth powers, asymmetric least squares and percentage differences. These error functions may not be the final evaluation criteria since other common forecasting evaluation methods such as the mean absolute percentage error (MAPE) are typically not minimized in neural networks.

In the case of commodity trading systems, the neural network forecasts would be converted into buy/sell signals according to a predetermined criterion. For example, all forecasts greater than 0.8 or 0.9 can be considered buy signals and all forecasts less than 0.2 or 0.1 as sell signals. The buy/sell signals are then fed into a program to calculate some type of risk adjusted return and the networks with the best risk adjusted return (not the lowest testing set error) would be selected. Low forecast errors and trading profits are not necessarily synonymous since a single large trade forecasted incorrectly by the neural networks could have accounted for most of the trading system's profits.

Step 7: Neural networks Training

Training a neural network to learn patterns in the data involves iteratively presenting it with examples of the correct known answers. The objective of training is to find the set of weights between the neurons that determine the global minimum of the error function. Unless the model is over fitted, this set of weights should provide good generalization. The BP network uses a gradient descent training algorithm which adjusts the weights to move down the steepest slope of the error surface. Finding the global minimum is not guaranteed since the error surface can include many local minima in which the algorithm can become ‘stuck’. A momentum term and five to ten random sets of starting weights can improve the chances of reaching a global minimum. This section will discuss when to stop training a neural network and the selection of learning rate and momentum values.

Number of Training Iterations

There are two schools of thought regarding the point at which training should be stopped. The first stresses the danger of getting trapped in a local minimum and the difficulty of reaching a global minimum. The researcher should only stop training until there is no improvement in the error function based on a reasonable number of randomly selected starting weights (Master, 1993). The point at which the network does not improve is called convergence. The second view advocates a series of train-test interruptions (Deboeck, 1994 and Mendelsohn, 1993). Training is stopped after a predetermined number of iterations and the network’s ability to generalize on the testing set is evaluated and training is resumed. Generalization is the idea that a model based on a sample of the data is suitable for forecasting the general population. The network for which the testing set error bottoms out is chosen since it is assumed to generalize the best.

The criticism of the train-test procedure is that additional train-test interruptions could cause the error on the testing set to fall further before rising again or it could even fall asymptotically. In other words, the research has no way of knowing if additional training could improve the generalization ability of the network especially since weights are randomized. The advantage of the convergence approach is that one can be more confident that the global minimum was reached. Replication is more difficult for the train-test approach given that starting weights are usually randomized and the mean correlation can fluctuate wildly as training proceeds. However, both schools of thought agree that generalization on the validation set is the ultimate goal and both use testing sets to evaluate a large number of networks. The point, at which the two approaches depart, centers on the notion of overtraining versus over fitting. The convergence approach states that there is no such thing as overtraining, only over fitting. Over fitting is simply a symptom of a network that has too many weights. The solution is to reduce the number of hidden neurons (or hidden layers if there is more than one) and /or increase the size of the training set. The train-test approach attempts to guard against over fitting by stopping training based on the ability of the network to generalize.

Learning Rate and Momentum

A BP network is trained using a gradient descent algorithm which follows the contours of the error surface by always moving down the steepest slope. The objective of training is to minimize the total squared errors, defined as follows:

$$E = \frac{1}{2}\sum_{h}^{M} E_h = \frac{1}{2}\sum_{h}^{M} \sum_{i}^{N} (t_{hi} - O_{hi})^2 \tag{4.1}$$

where E is the total error of all patterns, E_h represents the error on pattern h, the index h ranges over the set of input patterns, and i refers to the ith output neuron. The variable

t_{hi} is the desired output for the i th output neuron when the h th pattern is presented, the O_{hi} is the actual output of the i th output neuron when pattern h is presented. The learning rule to adjust the weight between neuron i and j is defined as:

$$\delta_{hi} = (t_{hi} - O_{hi})O_{hi}(1 - O_{hi}) \tag{4.2}$$

$$\delta_{hi} = O_{hi}(1 - O_{hi})\sum_{k}^{N}\delta_{hk}w_{jk} \tag{4.3}$$

$$\Delta w_{ij}(n+1) = \varepsilon(\delta_{hi}O_{hj}) \tag{4.4}$$

where n is the presentation number, δ_{hi} is the error signal of neuron i for pattern h, and ε is the learning rate. The learning rate is a constant of proportionality which determines the size of the weight changes. The weight change of a neuron is proportional to the impact of the weight from that neuron on the error. The error signal for an output neuron and a hidden neuron are calculated by Eq. (4.3) and (4.4), respectively.

During training, a learning rate that is too high is revealed when the error function is changing wildly without showing a continued improvement. On the other hand, a very small learning rate is evident when there is very little or no improvement in the error function and more training time. In either case, the researcher must adjust the learning rate during training or ‘brainwash’ the network by randomizing all weights and changing the learning rate for the new run through the training set. One method to increase the learning rate and thereby speed up training time without leading to oscillation is to include a momentum term in the BP learning rule. The momentum term determines how past weight changes affect current weight changes. The modified BP training rule is defined as follows:

$$\Delta w_{ij}(n+1) = \varepsilon(\delta_{hi} O_{hj}) + \alpha \Delta w_{ij}(n) \tag{4.5}$$

where α is the momentum term, and all other terms are as previously defined.

The momentum term suppresses side-to-side oscillations by filtering out high-frequency variations. Each new search direction is a weighted sum of the current and the previous gradients. Such a two-period moving average of gradients filters out rapid fluctuations in the learning rate. Momentum values that are too great will prevent the algorithm from following the twists and turns in weight space. McClelland and Rumelhart (1986) indicate that the momentum term is especially useful in error spaces containing long ravines that are characterized by steep, high walls and a gently sloping floor. Without a momentum term, a very small learning rate would be required to move down the floor of the ravine which would require excessive training time. By dampening the oscillations between the ravine walls, the momentum term can allow a higher learning rate to be used.

Most neural network software programs provide default values for learning rate and momentum that typically work well. Initial learning rates used in previous work vary widely from 0.1 to 0.9. Common practice is to start training with a higher learning rate such as 0.7 and decrease as training proceeds. Many neural network programs will automatically decrease the learning rate and increase momentum values as convergence is reached.

Step 8: Implementation

The implementation step is listed as the last one, but in fact requires careful consideration prior to collecting data. Data availability, evaluation criteria, and training times are all shaped by the environment in which the neural networks will be deployed. Most neural network software vendors provide the means by which trained networks can be implemented either in the neural network program itself or as an executable file. If not, a trained network can be easily created in a spreadsheet by knowing its architecture, transformation functions, and weights. Care should be taken that all data transformations, scaling, and other parameters remain the same from testing to actual use.

An advantage of neural networks is their ability to adapt to changing market conditions through periodic retraining. Once deployed, a neural network's performance will degrade over time unless retraining takes place. However, even with periodic retraining, there is no guarantee that network performance can be maintained as the independent variables selected may have become less important.

It is recommended that the frequency of retraining for the deployed network should be the same as used during testing on the final model. However, when testing a large number of networks to obtain the final model, less frequent retraining is acceptable in order to keep training times reasonable. A good model should be robust with respect to retraining frequency and will usually improve as retraining takes place more often.

4.3 Forecasting Property Stock Return Using the Monte Carlo BP Neural Networks

Six years of data are used to estimate the parameters of the within-sample neural network models. The independent variables are the accounting ratios and financial variables calculated at the beginning of each fiscal year with the 3-month lag as discussed in Chapter 1. In point-prediction models, the dependent variable is a firm's abnormal return. For classification problems, the dependent variable is the probability of a firm having a return above or below the median of the sample over the upcoming fiscal year.

4.3.1 Architecture of BP Neural Networks in Forecasting

The architecture of OLS neural networks and Logit neural networks are the same. By iterating within sample and examining the error terms, the neural network readjusts the input weights (β_{hj}) and output weights (ϕ_h) to minimize within-sample root mean square error for point prediction models. It maximizes the classification rate of correct predictions for classification models. I used the extended delta bar delta learning rule to back propagation the mean squared error from the output layer back to each neuron in the input layer. For this rule, the input weights (which are analogous to OLS or logit parameters) are adjusted at each of k iterations by

$$\beta_{hj}^{(k)} = \beta_{hj}^{(k-1)} + \Delta\beta_{hj}^{(k)}$$

where the change in input weight at each iteration is

$$\Delta\beta_{hj}^{(k)} = L_c e_h^{(k)} V_{jt-1} + m\Delta\beta_{hj}^{(k-1)}$$

Iterations continue until the portion of the error back propagated to each individual neuron (e_h^k) based on $\beta_{hj}^{(k-1)}$ parameter values reaches some tolerable level, or until the maximum number of iterations is reached. The learning rate coefficient (L_c) was set equal to 0.3 and the momentum coefficient (m) was set equal to 0.4 for all models. These coefficients control the speed of gradient decent around a local optimum.

In addition, an F' offset of 0.1 was added to the $F(.)$ squashing function at the each iteration to avoid saturation or closing down the learning process too soon. For all time periods, the number of hidden layers was fixed at one. The number of hidden neurons was set using a rule commonly used by other researchers, where H is the sum of input plus output variables divided by two. In event of a non-integer result, H was always around upward. In this study, all neural network models are estimated using Matlab 6.1. An epoch size of 30 was set for all models. This means that 30 training set observations were selected randomly and examined for error. Input weights were then adjusted and another 30 random observations were selected with weights changed using the extended delta bar delta learning rule. Based on the results from the validation sample, a maximum of 10,000 iterations provide a reasonable tradeoff between the problems of under-fitting and over-fitting the data.

4.3.2 The Model of OSL Neural Networks and Logit Neural Networks

Back propagation neural networks can be constructed for the point estimation and classification problems. I define neural networks for point estimation as OLS neural networks and for classification problems as logit neural networks. The estimation procedures and subsequent trading strategies for both point prediction and classification problems are similar in neural networks.

OSL Neural Networks

OSL neural networks corresponding to point- prediction problems can be represented by Eq. (4.6) as follows:

$$R_{it} = r_{it} - r_{mt} = \alpha + \sum_{h=1}^{H} \phi_h F(\gamma_h + \sum_{j=1}^{n} \beta_{hj} V_{jt-1}) + \varepsilon_{it} \tag{4.6}$$

where R_{it} is the abnormal return for stock i in fiscal year t calculated as the firm's actual return (r_{it}) minus the return on the portfolio of all stocks in the sample (r_{mt}) as measured by the average return for stocks in the sample in period t. H is the total number of hidden units or neurons in the hidden layer between inputs and outputs. γ_h is input threshold terms, β_{hj} are weights from the accounting ratio inputs to the hidden layer, while the ϕ_h parameters are weights from the hidden layer to the output layer. V_{jt-1} represents the inputs in the beginning of the fiscal year and denotes by the time subscript $t-1$ to indicate that they are known at the end of the previous period and not updated during the fiscal year t. α is a constant, and ε_{it} are error terms. Basically independent variables are put through a transformation or squashing function as represented by $F(.)$. For this purpose, following the research of Olson and Mossman (2002), this is the hyperbolic tangent function constrained to lie within the interval from -1 to +1.Denoting

$$z = (\gamma_h + \sum_{j=1}^{n} \beta_{hj} V_{jy-1})$$

in Eq.(4.6), the hyperbolic transformation or activation function is

$$F(z) = (\exp(z) - \exp(-z)) / (\exp(z) + \exp(-z))$$

Logit Neural Networks

Similarly, logit neural networks corresponding to classification problems can be represented as follows:

$$Log(P_{it} /(1 - P_{it})) = \alpha + \sum_{h=1}^{H} \phi_h F(\gamma_h + \sum_{j=1}^{n} \beta_{hj} V_{jt-1}) + \varepsilon_{it} \quad (4.7)$$

where P_{it} is that firm i in year t will have an abnormal return above that of the median firm. And the right-hand side of Eq.(4.7) is the same as that of Eq.(4.6). However, Eq.(4.7) is used to estimate probabilities for an abnormal return to lie within a two-class condition.

4.4.3 The Monte Carlo Neural Networks

To improve the performance of neural networks models, this study uses the Monte Carlo neural networks method. According to the characteristics of neural networks, most neural networks can not guarantee an optimal solution to a problem, a completely certain solution, or sometimes even lack repeatability with the same input data. Therefore, I implement the Monte Carlo method to decrease the variance of results.

The steps of this Monte Carlo process are shown in the following: first, a neural network is trained for an observation company with the input data from year 1993 to 1999 period. Then I simulated a predicted abnormal return of that company in 2000 with the input data of year 2000. Second, I repeated these training and simulation process 30 times with the same input data. Here, I assume that the distribution of output of the neural networks follows a normal distribution. Then, I averaged 30 predicted abnormal returns as that company's abnormal return in year 2000. Similarly, I can obtain that company's abnormal return in year 2001.

4.5 Summary

This chapter reviews the basic concepts, strengths and weaknesses of neural networks and introduces the BP neural networks which are the most popular neural networks used in economic analyses. Then, the BP neural networks are fitted to the property stock forecast using annual accounting ratios.

Chapter 5: Comparison and Analysis

5.1 Introduction

After four different methodologies (OLS neural networks, logit neural networks, OLS and logit regressions) were developed to forecast the company returns of one year ahead, this chapter shows and analyzes the results in terms of portfolio profitability for the four forecasting techniques.

This chapter is organized as follows. Section 5.2 presents the empirical results of stepwise OLS regressions and stepwise Logit regressions. Section 5.3 shows the empirical results of OLS neural networks and Logit neural networks. In Section 5.4, the results of four techniques are compared and analyzed in term of portfolio profitability. Section 5.5 concludes the chapter.

5.2 Empirical Results of Regressions

Both the stepwise OLS regression models and logit regression models presented in 3.3.1 and 3.3.2 were estimated by SPSS. The results of in terms of predictive accuracy of the abnormal returns are shown for each company in observation in this subsection. In this Chapter, the error of predicted abnormal return is equal to actual abnormal return minus predicted abnormal return.

5.2.1 Results of Stepwise OLS Regressions

Table 5.1 summarizes the predicted abnormal returns and error of all 13 companies using OLS forecasting techniques for the years 2000 and 2001.

Table 5.1 Predicted Abnormal Return Results by OLS Regressions

Name of Companies	2000	Error	2001	Error
BONVEST HOLDINGS	0.1427	-0.2566	0.0121	-0.0786
BUKIT SEMBAWANG EST	-0.1511	0.1880	-0.2407	0.5427
CHEMICAL INDL. (FE)	1.7614	-1.8318	0.3597	-0.1966
CITY DEVELOPMENTS	-0.0491	-0.0109	0.1532	0.0077
HONG FOK CORPORATION	0.3034	-0.1930	0.0400	-0.2085
KEPPEL LAND	-0.6539	0.6235	0.0190	0.0775
MCL LAND	0.3614	-0.3948	-1.8293	2.1950
ORCHARD PARADE HDG	-0.0519	-0.1876	-0.3150	0.4155
SINGAPORE LAND	0.5428	-0.1467	0.6006	-0.3792
UNITED OVERSEAS LAND	0.4330	0.1189	0.5234	-0.1634
WING TAI HOLDINGS	-0.2213	-0.0047	0.7001	-0.4378
CAPITALAND	-0.2637	0.2653	0.0212	-0.0710
MARCO POLO DEV	0.0648	0.1629	0.4861	-0.2010
Average	0.1340	-0.1286	0.0618	0.0703
Max	1.7614	0.6235	0.7001	2.1950
Min	-0.6539	-1.8318	-1.8293	-0.5177

From Table 5.1, the average errors of predicted abnormal returns are -0.1286 and 0.0703 for year 2000 and 2001 respectively. However, the range of errors of predicted abnormal returns are from -1.8318 to 0.6235 in 2000 and from -0.5177 to 2.1950 in 2001.

5.2.2 Results of Stepwise Logit Regressions

Table 5.2 summarizes the predicted abnormal returns and error of all 13 companies using OLS forecasting techniques for the years 2000 and 2001.

Table5.2 Predicted Abnormal Return Results by Logit Regressions

Name of Companies	2000	Error	2001	Error
BONVEST HOLDINGS	1.0000	-1.0000	0.0000	0.0000
BUKIT SEMBAWANG EST	0.0000	1.0000	1.0000	0.0000
CHEMICAL INDL. (FE)	0.0000	0.0000	1.0000	0.0000
CITY DEVELOPMENTS	1.0000	-1.0000	1.0000	0.0000
HONG FOK CORPORATION	1.0000	0.0000	0.0133	-0.0133
KEPPEL LAND	0.0000	0.0000	No Value	No Value
MCL LAND	1.0000	-1.0000	0.1471	0.8529
ORCHARD PARADE HDG	0.0035	-0.0035	0.0000	1.0000
SINGAPORE LAND	1.0000	0.0000	0.5247	0.4753
UNITED OVERSEAS LAND	0.9954	0.0046	1.0000	0.0000
WING TAI HOLDINGS	1.0000	-1.0000	0.9973	0.0027
CAPITALAND	0.0000	1.0000	0.0000	0.0000
MARCO POLO DEV	0.8712	0.1288	1.0000	0.0000

As shown in Table 5.2, it is observed that the logit regression technique is not applicable to the data of Keppel Land Company in 2001. Even though P-value is adjusted downward to 90%, the stepwise regressions still selected no independent variables. This means that the data set of Keppel Land Company do not include independent variables which can explain its dependent variable with proper P-value for the estimation period, 1995 to 2000.

Moreover, in logit regressions, for the error of predicted abnormal returns, the value closer to 0, the better are the results. According to this characteristic, it is found the error of 7 out of 13 companies is less than 0.2 in 2000 while 9 out of 12 is less than 0.1 in 2001.

5.3 Results of the Monte Carlo Neural Networks

As explained in 4.4.3 where neural networks could improve the stability of the performance of neural networks by reducing the errors of outputs, this point will be shown in this subsection. The results of all observation companies using The Monte Carlo neural networks are presented in Appendix 1-14.

For example, I will only examine the predicted abnormal return results of MCL land Company. From Table 5.3, for OLS neural networks, the average errors of predicted abnormal returns are only 0.0602 and 0.3386 for year 2000 and 2001 respectively. However, the range of errors of predicted abnormal returns are from -0.9867 to 0.9294 in 2000 and from -0.5801 to 1.3003 in 2001. Moreover, for logit neural networks, the average errors of predicted abnormal returns are only -1.4823 and 0.8818 for year 2000 and 2001 respectively, while the range of errors of predicted abnormal returns are from -1.9451 to -1.0110 in 2000 and from 0.0356 to 0.9986 in 2001. This similarity could be found in the predicted results of all other companies using The Monte Carlo neural networks.

Table 5.3 Predicted Abnormal Return Results of MCL LAND by The Monte Carlo Neural networks

Methods	OLS Neural networks				Logit Neural networks			
Number	2000	Error	2001	Error	2000	Error	2001	Error
1	0.5163	-0.5498	-0.3736	0.7393	0.3586	-1.3586	0.0095	0.9905
2	-0.0089	-0.0245	0.3989	-0.0332	0.0648	-1.0648	0.0015	0.9986
3	-0.6233	0.5898	-0.1404	0.5062	0.9384	-1.9384	0.0280	0.9720
4	-0.3103	0.2768	0.2001	0.1656	0.0123	-1.0123	0.1573	0.8428
5	-0.2472	0.2137	-0.0044	0.3701	0.0120	-1.0120	0.0108	0.9892
6	-0.5309	0.4974	-0.7330	1.0987	0.5406	-1.5406	0.1221	0.8779
7	0.3563	-0.5309	0.3502	0.0155	0.5821	-1.5821	0.0069	0.9932
8	-0.6209	0.5874	0.4784	-0.1127	0.2338	-1.2338	0.3119	0.6882
9	0.8757	-0.9092	0.5710	-0.2052	0.3463	-1.3463	0.0342	0.9659
10	-0.9628	0.9294	0.0416	0.3241	0.0202	-1.0202	0.1276	0.8725
11	0.2471	-0.2805	-0.1586	0.5244	0.8591	-1.8591	0.0126	0.9874
12	-0.7638	0.7304	0.0420	0.3237	0.7677	-1.7677	0.0445	0.9555
13	0.7874	-0.8208	0.0428	0.3230	0.3052	-1.3052	0.0080	0.9920
14	0.8535	-0.8870	0.1637	0.2020	0.3257	-1.3257	0.4352	0.5649
15	-0.1022	0.0688	-0.2727	0.6384	0.4814	-1.4814	0.0705	0.9295
16	-0.8085	0.7751	0.4565	-0.0907	0.4814	-1.4814	0.0220	0.9781
17	-0.1700	0.1365	0.1385	0.2273	0.5624	-1.5624	0.0372	0.9628
18	0.1650	-0.1985	-0.0959	0.4617	0.9291	-1.9291	0.0790	0.9210
19	-0.1855	0.1521	-0.5097	0.8754	0.3133	-1.3133	0.3119	0.6882
20	-0.6946	0.6611	-0.0618	0.4275	0.8540	-1.8540	0.0345	0.9655
21	0.9533	-0.9867	0.7627	-0.3970	0.2090	-1.2090	0.9645	0.0356
22	-0.8705	0.8371	-0.5525	0.9182	0.1861	-1.1861	0.0126	0.9874
23	-0.7947	0.7613	0.9458	-0.5801	0.9103	-1.9103	0.0556	0.9444
24	-0.0054	-0.0280	0.7075	-0.3418	0.8990	-1.8990	0.0488	0.9512
25	0.8620	-0.8955	0.0350	0.3307	0.9138	-1.9138	0.0175	0.9825
26	-0.8340	0.8006	0.5179	-0.1522	0.0110	-1.0110	0.0173	0.9827
27	0.2235	-0.2569	-0.1894	0.5552	0.0322	-1.0322	0.0091	0.9909
28	-0.3897	0.3562	-0.5128	0.8785	0.6262	-1.6262	0.3226	0.6774
29	0.0089	-0.0424	-0.5007	0.8664	0.7470	-1.7470	0.1565	0.8435
30	0.1249	-0.1583	-0.9345	1.3003	0.9451	-1.9451	0.0782	0.9218
Average	-0.0983	0.0602	0.0271	0.3386	0.4823	-1.4823	0.1182	0.8818
Max	0.9533	0.9294	0.9458	1.3003	0.9451	-1.0110	0.9645	0.9986
Min	-0.9628	-0.9867	-0.9345	-0.5801	0.0110	-1.9451	0.0015	0.0356

5.3.1 Results of OLS Neural networks

Table 5.4 shows the predicted abnormal returns and errors of all observation companies using The Monte Carlo OLS neural networks forecasting techniques for the years 2000 and 2001.

Table 5.4 Predicted Abnormal Return Results by OLS Neural Networks

Name of Companies	2000	Error	2001	Error
BONVEST HOLDINGS	0.0017	-0.1157	-0.0259	-0.0406
BUKIT SEMBAWANG EST	0.2079	-0.1709	0.1392	0.1629
CHEMICAL INDL. (FE)	-0.0229	-0.0476	0.1141	0.0490
CITY DEVELOPMENTS	0.1086	-0.1687	0.0049	0.1561
HONG FOK CORPORATION	0.0763	0.0341	0.1486	-0.3171
KEPPEL LAND	-0.0783	0.0479	0.0953	0.0012
MCL LAND	-0.0983	0.0649	0.0271	0.3386
ORCHARD PARADE HDG	-0.0525	-0.1869	0.1050	-0.0045
SINGAPORE LAND	-0.1080	0.5041	0.0964	0.1250
UNITED OVERSEAS LAND	0.1021	0.4499	0.2353	0.1246
WING TAI HOLDINGS	-0.0188	-0.2072	-0.1310	0.3933
CAPITALAND	0.0856	-0.0841	-0.0723	0.0225
MARCO POLO DEV	-0.0312	0.2589	0.0530	0.2321
Average	0.0132	0.0291	0.0607	0.0956
Max	0.2079	0.5041	0.2353	0.3933
Min	-0.1080	-0.2072	-0.1310	-0.3171

From Table 5.4, the average errors of predicted abnormal returns are 0.0291 and 0.0956 for year 2000 and 2001 respectively. Moreover, the range of errors of predicted abnormal returns are from 0.5041 to -0.2072 in 2000 and from 0.3933 to -0.3171. Compared to the results of OLS regression models in Table 5.1, the results of OLS neural networks are generally better.

5.3.2 Results of Logit Neural Networks

Table 5.5 shows the predicted abnormal returns and error of all observation companies using The Monte Carlo logit neural networks forecasting techniques to forecast the Singapore property stock returns 1-year-ahead.

Table 5.5 Predicted Abnormal Return Results by Logit Neural Networks

Name of Companies	2000	Error	2001	Error
BONVEST HOLDINGS	0.0000	0.0000	0.2431	-0.2431
BUKIT SEMBAWANG EST	0.4686	0.5314	0.7668	0.2332
CHEMICAL INDL. (FE)	0.9800	-0.9800	0.5046	0.4954
CITY DEVELOPMENTS	0.5415	-0.5415	0.4943	0.5057
HONG FOK CORPORATION	0.4950	0.5050	0.4026	-0.4026
KEPPEL LAND	0.1046	-0.1046	0.4873	0.5127
MCL LAND	0.4823	-0.4823	0.1182	0.8818
ORCHARD PARADE HDG	0.5550	-0.5550	0.5730	0.4270
SINGAPORE LAND	0.4016	0.5984	0.3934	0.6066
UNITED OVERSEAS LAND	0.9978	0.0022	0.9953	0.0047
WING TAI HOLDINGS	0.4475	-0.4475	0.5559	0.4441
CAPITALAND	0.7181	0.2819	0.6756	-0.6756
MARCO POLO DEV	0.5826	0.4174	0.8036	0.1964

5.4 Comparison and Analysis

This research compares the forecasting performance of back propagation neural networks with traditional regression techniques using fundamental accounting data as an input to forecast 1-year-ahead Singapore property stock returns. Forecasts are judged primarily on the basis of trading rule profit. Profitability is what matters to most market participants; so trading rule profit is a better measure of the usefulness of a forecasting model than statistical measures of forecast error (e.g. mean squared error, mean absolute error, or mean absolute percentage error).

In this study, trading rule profit is the portfolio return, or the returns in percentage from a stock selection strategy using four different forecasting techniques, including OLS regressions, logit regressions, OLS neural networks and logit neural networks. According to the forecast of 2000 and 2001 year return by each different technique, I will select property stocks with positive abnormal returns or higher probability to out perform median stocks to construct an equal weighted portfolio. Then, each portfolio is assumed to be held for a two-year period. The holding or selling the stocks of the portfolio is adjusted yearly according to the forecast of 1-year-ahead return suggested by each technique. At the end, the annual abnormal return of each portfolio (R_{2000p}, R_{2001p}) in 2000 and 2001 is calculated, which consists of the average abnormal returns for all selected stocks (R_{2000i}).

$$R_{2000p} = \frac{1}{n}\sum_{i=1}^{n} R_{2000i} \tag{5.1}$$

Then, the total abnormal return for 2 year holding period of the portfolio (R_{TP}) is equal to

$$R_{TP} = (1 + R_{2000p})(1 + R_{2001p}) \tag{5.2}$$

Table 5.6 Real Abnormal Returns of All Observation Companies

Name of Companies	2000(R_{2000i})	2001(R_{2001i})
BONVEST HOLDINGS	-0.1140	-0.0665
BUKIT SEMBAWANG EST	0.0370	0.3020
CHEMICAL INDL. (FE)	-0.0704	0.1631
CITY DEVELOPMENTS	-0.0601	0.1609
HONG FOK CORPORATION	0.1104	-0.1685
KEPPEL LAND	-0.0304	0.0965
MCL LAND	-0.0335	0.3657
ORCHARD PARADE HDG	-0.2394	0.1005
SINGAPORE LAND	0.3961	0.2214
UNITED OVERSEAS LAND	0.5520	0.3600
WING TAI HOLDINGS	-0.2260	0.2623
CAPITALAND	0.0016	-0.0498
MARCO POLO DEV	0.2277	0.2851

5.4.1 Portfolios Constructed by OLS Regressions

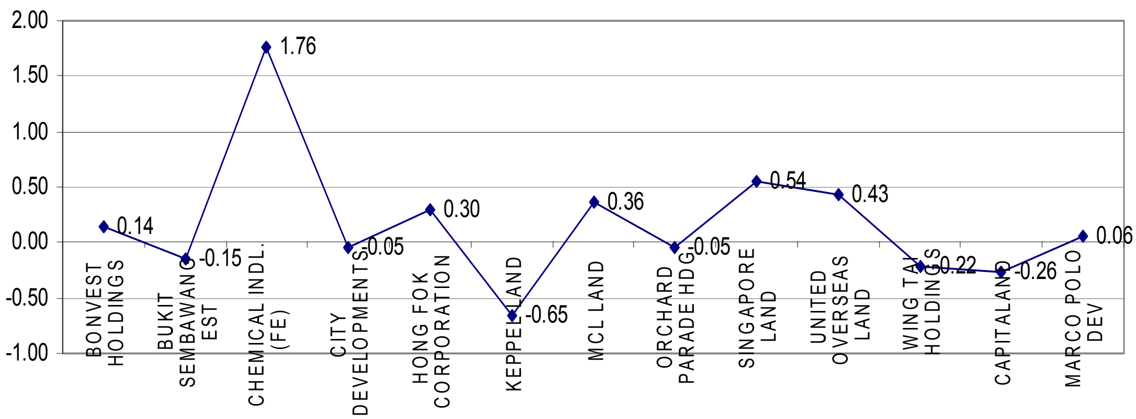

Fig.5.1 Predicted Abnormal Return Results in 2000 by OLS Regressions

I select the stocks with positive predicted abnormal returns to construct a portfolio. From the Fig.5.1, I will select 7 stocks out of 14 observations, including

Bonvest Holdings, Chemical INDL. (FE), Hong Fok Corporation, MCL Land, Singapore Land, United Overseas Land and Marco Polo Development.

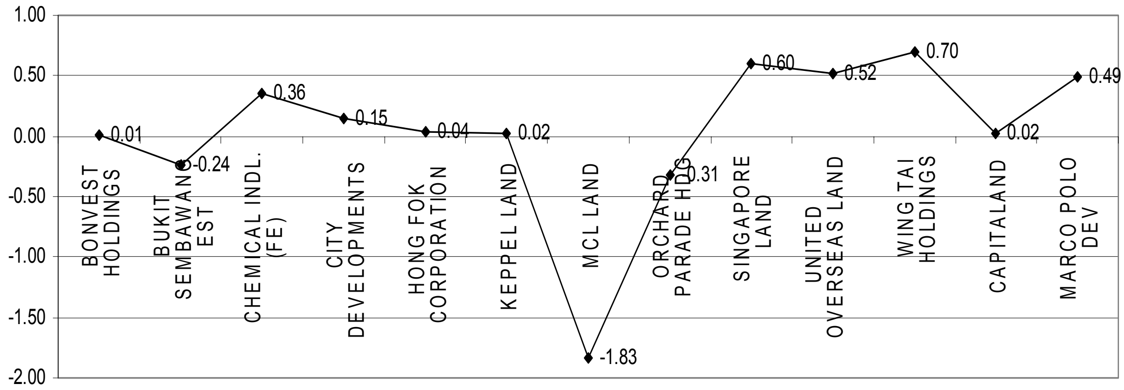

Fig.5.2 Predicted Abnormal Return Results in 2001 by OLS Regressions

Similarly, according to Fig.5.2, for year 2001 the portfolio can be constructed by these several stocks: Bonvest Holdings, Chemical INDL. (FE), City Developments, Hong Fok Corporation, Keppel Land, Singapore Land, United Overseas Land, Wing Tai Holdings, Capitaland and Marco Polo Development.

5.4.2 Portfolios Constructed by Logit Regressions

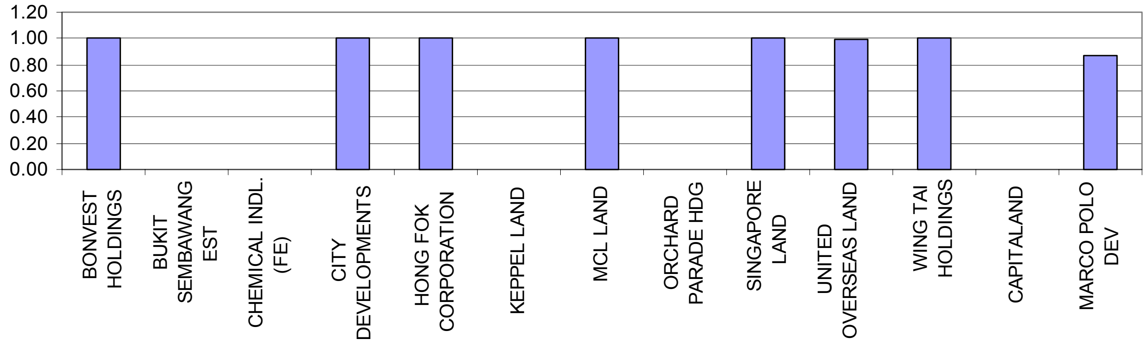

Fig.5.3 Predicted the Probability of Abnormal Results in 2000 by Logit Regressions

For logit regression models, I select the stocks with probability that out perform the median firm higher than 0.6. According to this rule, the equal weighted portfolio in 2000 can be constructed by these companies, Bonvest Holdings, City Developments, Hong Fok Corporation, MCL Land, Singapore Land, United Overseas Land, Wing Tai Holdings and Marco Polo Development.

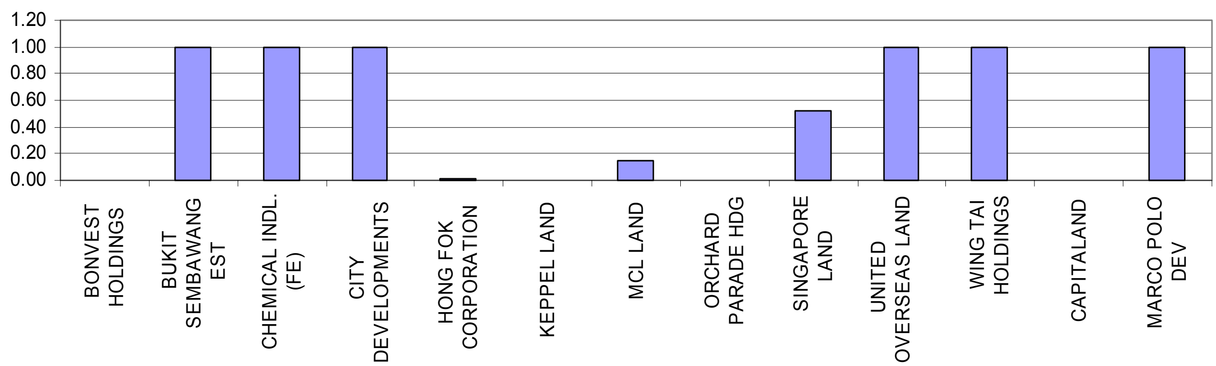

Fig.5.4 Predicted the Probability of Abnormal Results in 2001 by Logit Regressions

Likely, the portfolio in 2001 could select these stocks: Bukit Sembawang Est, Chemical INDL. (FE), City Developments, United Overseas Land, Wing Tai Holdings and Marco Polo Development.

5.4.3 Portfolios Constructed by OLS Neural networks

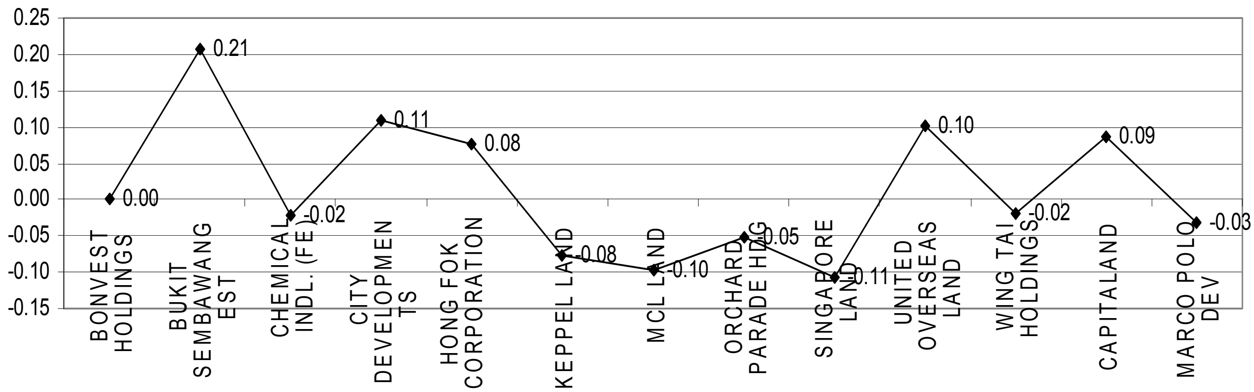

Fig.5.5 Predicted Probability Abnormal Results in 2000 by OLS Neural networks

As for OLS regression models, I select the stocks with positive predicted returns to form portfolio for OLS neural network models. Therefore, according to Fig.5.5, Bukit Sembawang Est, City Developments, Hong Fok Corporation, United Overseas Land and Capitaland are selected for the portfolio in 2000.

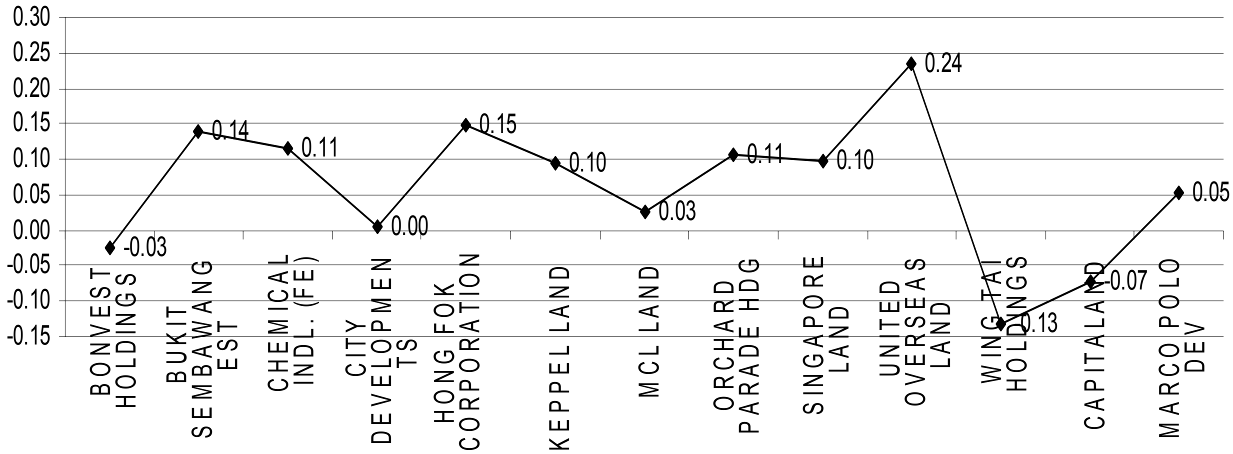

Fig.5.6 Predicted Probability Abnormal Results in 2001 by Logit Neural networks

Similarly, according to the Fig.5.5, the portfolio in 2001 selects the stocks: Bukit Sembawang EST, Chemical INDL. (FE), Hong Fok Corporation, Keppel Land, MCL Land, Orchard Parade HDG, Singapore Land, United Overseas Land and Marco Polo Development.

5.4.4 Portfolios Constructed by Logit Neural networks

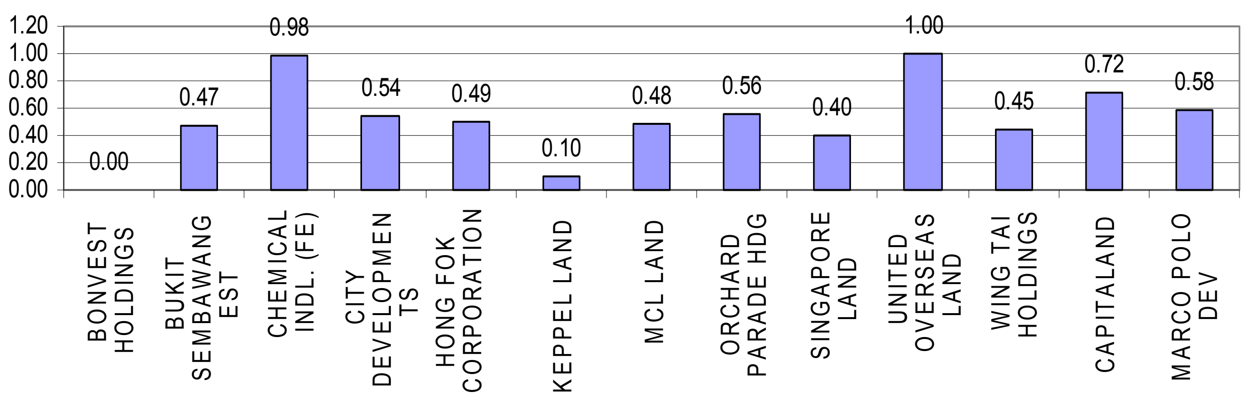

Fig.5.7 Predicted Probability Abnormal Results in 2000 by Logit Neural networks

Using logit regression models, I select the stocks with probability bigger than 0.6, which means that the firms have 0.6 probability to out perform the median firm. According to this rule, the equal weighted portfolio in 2000 can be constructed by three companies, Chemical INDL. (FE), United Overseas Land and Capitaland.

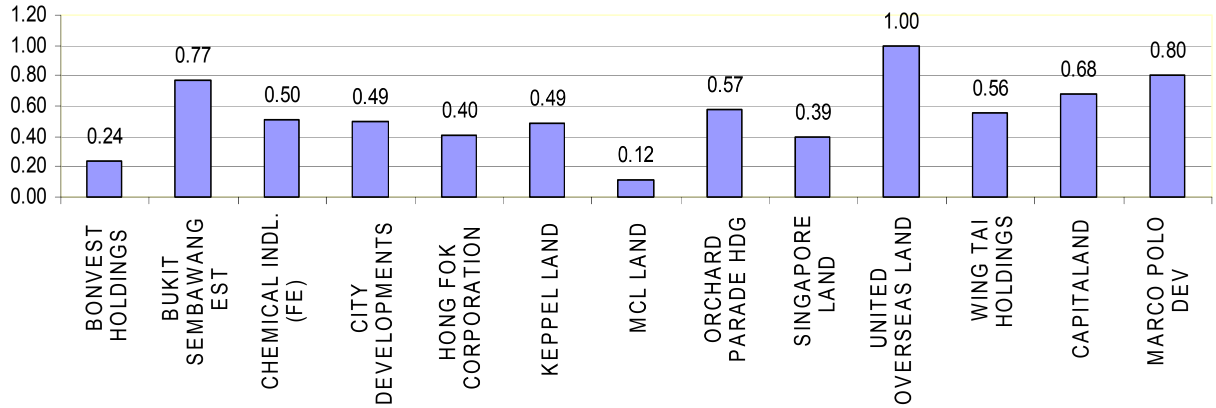

Fig.5.8 Predicted Probability Abnormal Results in 2001 by Logit Neural Networks

Similarly, the portfolio in 2001 select companies, Bukit Sembawang Est, United Overseas Land, Capitaland and Marco Polo Development.

5.4.5 Comparison of the Performance of 4 Portfolios

As mentioned before, forecasts of these four techniques are judged primarily on the basis of trading rule profit. Profitability is what matters to most market participants; so trading rule profit is a better measure of the usefulness of a forecasting model than statistical measures of forecast error (e.g. mean squared error, mean absolute error, or mean absolute percentage error).

Table 5.7 shows the abnormal return results of 4 portfolios constructed according to different forecasting techniques.

Table 5.7 the Abnormal Returns of Portfolios in 2 Year Holding Period

Forecasting Techniques	2000* (R_{2000p})	2001* (R_{2001p})	Total Return in 2 Years(R_{TP})**
OLS Regressions	0.1526	0.1264	1.2984
Logit Regressions	0.1066	0.2556	1.3894
OLS Neural Networks	0.1282	0.1532	1.3011
Logit Neural Networks	0.1610	0.2243	1.4215

* is calculated according to equation 5.1; ** is calculated according to equation 5.2.

A feature of logit estimation is that it makes a nonlinear transformation of the input data that decreases the influence of outliers. Therefore, if outliers and noisy data are a problem, logit should outperform OLS; while if the magnitude of actual past returns are important, OLS should outperform logit.

For this forecasting problem, classification estimation seems better suited than point estimation. The total returns in 2 year period of logit regressions and logit neural networks models are 1.3894 and 1.4215 respectively. These are better than those of OLS regressions and OLS neural network models, although logit models use less information than the OLS model because the dependent variable takes on a value of zero or one, instead of the whole range of possible positive and negative market-adjusted abnormal returns. These results are also consistent to the findings in Olson and Mossman (2002).

Moreover, the results also indicate that the logit neural networks, which consider non-linear relationship between input and output variables, out performs the corresponding logit regressions. Total return in 2 year period for portfolio by logit neural networks is 1.4215, which is larger compared to that of 1.3849 for the logit model. Therefore, this superiority of logit neural network models translates into greater profitability in the trading rules.

5.5 Summary

Similar to previous results using Canadian stocks (Olson and Mossman, 2002), this study shows that classification models outperform point estimation models for the research problems. Moreover, in classification models logit neural networks outperform the logit regression models.

Therefore, it have been shown that investors should prefer classification models to point estimation models and choose logit neural network models instead of traditional logit models. Moreover, the results supported the Monte Carlo neural networks which improve the stability of the neural network performance by reducing the errors of outputs.

Chapter 6 Summary and Conclusion

6.1 The Significance of this Work

Similar to previous results using US stocks and Canadian stocks (Brockman, Mossman, and Olson (1997); Olson and Mossman (2002)), this work also shows that fundamental analysis leads to abnormal returns in Singapore property and accounting ratios could be leading indicators of stock returns in the next year.

Moreover, like previous study of Olson and Mossman (2002), this work indicates that classification models outperform point estimation models for problems at hand. In classification models logit neural networks out perform the logit regression models. Therefore, investors are advised to prefer classification models to point estimation models and choose logit neural network models instead of traditional logit models to pick stocks. Then, the investors can pick (avoid) those stocks that are likely to out perform (under perform) other stocks.

Furthermore, this work helps to better identify the borderline at which neural networks can out perform traditional regression-based forecasting techniques. Based on the review of the literature, ANNs might be expected to perform better than traditional OLS and logit regression techniques in forecasting stock returns, but neither approach dominates the other. This work compares neural network forecasts of one-year-ahead Singapore property stock returns with the forecasts obtained using OLS regression and logit regression techniques. Then, it is found logit neural networks out perform all other three methods, OLS neural networks, OLS regression and logit regression.

This work uses Monte Carlo neural network method to improve the stability of the performance of neural networks by reducing the variance of outputs. According to the characteristics of neural networks, most neural networks can not guarantee an optimal solution to a problem or sometimes even lack repeatability with the same input data. The findings indicate that the Monte Carlo neural networks reduce the errors of output by averaging the predicted output value.

In addition, this work is the first to use neural networks to examine the performance of Singapore property stocks. In Singapore, although there are some works focusing on real estate stock performance (Liow (1997 and 1998a); Sing (2001); Sing *et al.*(2002); Chan and Sng (1991); Ong (1994 and 1995)), neural network techniques are used relatively scarcely.

6.2 The Limitation of this Work

Applying the techniques of the research of Olson and Mossman (2002) in Singapore real estate stock, this work uses six years data to estimate the parameters of models. However, due to the availability of the data, I can only get the data set from 1992 to 2001, so the portfolios were constructed for only 2 years period. It is better if a longer period portfolio could be constructed to investigate the performance of neural networks and traditional regressions. Moreover, due to the time limitation, this work only studied Singapore property stocks. It is preferable if this thesis could include all Singapore stocks into this sample to compare the performance of four techniques.

6.3 Recommendation for Future Works

The Monte Carlo neural network method can be further studied on how much time of Monte Carlo is preferable for logit neural networks and OLS neural networks respectively. In this work, it seems logit neural networks converge faster than OLS neural networks using Monte Carlo method (see Appendix 1 to 13).

Moreover, the techniques used here represent but a small sample of infinitely many alternatives that could potentially lead to superior results. Changes could be made at the data collection stage by adding additional variables; different significance levels could be used to select explanatory variables, and alternative neural network architectures could be considered.

Furthermore, as mentioned a little in the limitation of this work, if time and data available, future studies could include as many as possible Singapore stocks and construct portfolios with longer period to compare neural networks and regression techniques.

Bibliography

Adya, M., and Collopy, F., How Effective are Neural Networks at Forecasting and Prediction? A Review and Evaluation, *Journal of Forecasting*, 1998, Vol.17, pp.481–495.

Altman, D. G. and P. K. Andersen, Bootstrap Investigation of The Stability of a Cox Regression Model, *Statistics in Medicine* 1989,Vol. 8, pp.771-783.

Balkin, S. D., & Ord, J. K., Automatic Neural Network Modeling for Univariate Time Series, *International Journal of Forecasting*, 2000, Vol.16, pp.509–515.

Brockman, P., Mossman, C., & Olson, D., What's the value of fundamental analysis? *Canadian Investment Review*, 1997, Vol.6, pp.10–15.

Brooks, C., Linear and Non-linear (non-) Forecastability of High Frequency Exchange Rates, *Journal of Forecasting*, 1997, Vol.16, pp.125–145.

Callen, J. L., Kwan, C., Yip, P., & Yuan, Y., Neural Network Forecasting of Quarterly Accounting Earnings, *International Journal of Forecasting*, 1996, Vol.12, pp.475–482.

Caudill, M., The View From Now, *AI Expert*, 1992, June, pp.24-31.

Chan, K.C., Hendershott, C.P., and Sanders, A., Risk and Return on Real Estate Evidence from Equity REITs, *Journal of the American Real Estate and Urban Economics Association,* 1990, Vol.18, pp.431-452.

Chan, S.L. and Sng, G.N., Real Estate Versus Property Stock Investment, *Securities Industry Review*, 1991, Vol.17(2), pp.9-15.

Chatfield, C., Neural Network: Forecasting Breakthrough or Passing Fad, *International Journal of Forecasting*, 1993, Vol.9, pp.1–3.

Chatfield, C., Positive or Negative, *International Journal of Forecasting*, 1995, Vol.11, pp.501–502.

Church, K., & Curram, S., Forecasting Consumers' Expenditure: a Comparison between Econometric and Neural Network Methods, *International Journal of Forecasting*, 1996, Vol.12, pp.255–267.

Coats, P.K. and Fant, L.F., A Neural Network Approach to Forecasting Financial Distress, *The Journal of Business Forecasting*, 1992, Winter, pp.9-12.

Coats, P.K. and Fant, L.F., Recognizing Financial Distress Patterns Using a Neural Network Tool, *Financial Management*, 1993, 22(3), pp.142-155.

Copas, J. B., Regression, Prediction and Shrinkage with Discussion, *Journal of the Royal Statistical Society*, 1983, 45, pp.311-354.

Cragg, J.G. and Uhler, R., The Demand for Automobiles, *Canadian Journal of Economics*, 1970, Vol.3, pp.386-406.

Darbellay, G. A., & Slama, M., Forecasting the Short-term Demand for Electricity? Do Neural Networks Stand a Better Chance? *International Journal of Forecasting*, 2000, Vol.16, pp.71-83.

Derksen, S. and Keselman, H. J., Backward, Forward and Stepwise Automated Subset Selection Algorithms: Frequency of Obtaining Authentic and Noise Variables, *British Journal of Mathematical and Statistical Psychology*, 1992, Vol.45, pp.265-282.

Deboeck, G.J., *Trading on the Edge: Neural, Genetic, and Fuzzy Systems for Chaotic Financial markets*, Wiley, New York, 1994.

Elton, E.J. and Gruber, M.J., Modern Portfolio Theory and Investment Analysis, New York: John Wiley & Sons, Inc., 1991, p.561.

Episcopos, A., and Davies, J., Predicting Returns on Canadian Exchange Rates with Artificial Neural Networks and EGARCH-M Models, *Neural Computing and Applications*, 1995, Vol.5, pp.1–11.

Ersoy, O., *Tutorial at Hawaii International Conference on Systems Sciences*, 1990, Jan.

Foster, F. R. and Robert A. Stine, Variable Selection in Data Mining: Building a Predictive Model for Bankruptcy, Department of Statistics, The Wharton School of the University of Pennsylvania Philadelphia, PA 19104-6340, October 29, 2002

Glascock J.L., Lu C.L., So R.W., REIT Returns and Inflation: Perverse or Reverse Causality Effects? *Journal Real Estate Finance*, 2002, 24 (3), pp.301-317.

Gorr,W., Research Prospective on Neural Network Forecasting, *International Journal of Forecasting*, 1994, Vol.10, pp.1–4.

Haefke, C., & Helmenstein, C., Forecasting Austrian IPOs: an Application of Linear and Neural Network Error-correction Models, *Journal of Forecasting*, 1996, Vol.15, pp.237–251.

Hecht-Nielsen, R. *Neurocomputing*, Addison Wesley, Menlo Park, CA, 1989.

Hill, T., Marquez, L., O'Conner, M., and Remus, W., Artificial Neural Network Models for Neural Network Forecasting of Quarterly Accounting Earnings, *International Journal of Forecasting*, 1994, Vol.10, pp.5–15.

Ho, K. H., Cuervo, J. C., A Cointegration Approach to the Price Dynamics of Private Housing. A Singapore case study, *Journal of Property Investment & Finance*, 1999, Vol. 17 (1), pp. 35-60.

Holthausen, R. W., & Larcker, D. F., The Prediction of Stock Returns Using Financial Statement Information," *Journal of Accounting and Economics*, 1992, Vol.15, pp.373–411.

Hornik, K., Stinchcombe, M. and White, H., Multilayer Feedforward Networks are Universal Approximators, *Neural Networks*, 1989, 2, pp.359-366.

Kaastra, I. and Boyd, M., Designing a Neural Network for Forecasting Financial and Economic Time Series, *Neurocomputing*, 1996, 10, pp.215-236.

Kane, G. D., Richardson, F. M., & Meade, N. L., Rank Transformations and The Prediction of Corporate Failure, *Contemporary Accounting Research*, 1998,Vol.15, pp.145–166.

Kao, G.W. and Ma, C.K., Memories, Heteroscedasticity and Prices Limit in Currency Futures Markets, *Journal Futures Markets*, 1992, 12, pp.672-692.

Kapplin, 1995 (eds) *Alternative Real Estate Investment*, Kluwer Academic Publisher, Norwall, M.A., 1995, pp.59-82.Katz, J.O., Developing Neural Network Forecasters for Trading, *Technical Analysis of Stocks and Commodities*, 1992, April, pp.58-70.

Katz, S., Lilien, S., and Nelson, B., Stock Market Behavior around Bankruptcy Model Distress and Recovery Predictions, *Financial Analysts Journal*, 1985, Vol.36, pp.70-74.

Kryzanowski, L., Galler, M., & Wright, D., Using Artificial Neural Networks to Pick Stocks, *Financial Analysts Journal*, 1993, Vol.49, pp.21–27.

Kuan, C., & Lim, T. Forecasting Exchange Rates Using Feed Forward and Recurrent Neural Networks, *Journal of Applied Econometrics*, 1994, Vol.10, pp.347–364.

Leung, M. T., Daouk, H., & Chen, A. S., Forecasting Stock Indices: a Comparison of Classification and Level Estimation Models, *International Journal of Forecasting*, 2000, Vol.16, pp.173–190.

Levich, R.M. and Thomas, L.R., The Significance of Technical Trading Rule Profits in the Foreign Exchange Market: A Bootstrap Approach; in *Strategic Currency investing-*

Trading and Hedging in the Foreign Exchange Market, Probus, Chicaga, 1993, pp.336-365.

Ling, D.C. and Naranjo, A., Commercial Real Estate Return Performance: A Cross-Country Analysis, *Journal of Real Estate Finance and Economics*, 2002, Vol.24(1/2), pp.119-142.

Liow, K.H., Property Companies' Share Price Discounts and Property Market Returns-The Singapore Evidence, *Journal of Property Finance*, 1996, Vol.7(4), pp.64-77.

Liow, K.H., An Empirical Investigation of Corporate Growth of Property Companies, *Journal of Financial Management of Property and Construction*, 1998a, 3(3), pp. 5-16.

Liow, K.H., Relationship between Singapore Property Stock and Real Estate Returns, *Journal of Real Estate and Construction*, 1998b, Vol.8, pp.1-16.

Liow, K.H., The Historical Performance of Singapore Property Stocks, *Journal of Property Finance*, 1997, Vol.8(2), pp.111-125.

Lippman, R.P., Review of Neural Networks for Speech Recognition, *Neural Computation1*, 1999(1), pp.1-38.

Liu, C.H., Grissom, T.V., and D.J. Hartzell, Superior Real Estate Investment Performance: Egnima or Illusion? A Critical Review of the Literature, in Schwartz, A.L. and S.D.

Mantel, N., Why Step down Procedures in Variable Selection, *Technometrics*, 1970, Vol.12, pp.621-625.

Master, T., *Practical Neural Network Recipes in C++*, Academic Press, New York, 1993.

McClelland, J.L. and Rumelhart, D.E., and the PDP Group, *Parallel Distributed Processing. Explorations in the Microstructure of Cognition, Vol.1: Foundations*, Cambridge, MA, MIT Press, 1986.

McFadden, D., The Measurement of Urban Travel Demand, *Journal of Public Economics*, 1974, Vol. 3, pp.303-328.

Medsker, L., Trippi, R.R., and Turban, E., Neural Network Fundamentals for Financial Analysts, *Neural Networks in Finance and Investing*, 1996, pp.1-24.

Mendelsohn, L., Training Neural Networks, *Technical Analysis of Stocks and Commodities*, 1993, Nov., pp.40-48.

O'Conner, M., On the Usefulness of Financial Statement Analysis and the Prediction of Stock Return, *Accounting Review*, 1973, Vol.48, pp.339-352.

Olson, D. and Mossman, C., Neural Network Forecasts of Canadian stock Returns Using Accounting Ratios, *International Journal of Forecasting*, 2002, Jan., pp.1-13.

Ong, S.E., Singapore Real Estate and Property Stocks- a Co-Integration Test, *Journal of Property Research*, 1995, Vol.12(1), pp.29-39.

Ong, S.E., Structural and Vector Autoregressive Approaches to Modeling Real Estate and Property Stock Prices in Singapore, *Journal of Property Finance*, 1994, Vol.5(4), pp.4-18.

Ooi, T.L and Liow, K.H., Risk-adjusted Performance on Real Estate Stocks: Evidence from Emerging Markets in Asia, *Annual Conference of American Real Estate and Urban Economics Association*, Jan. 3-5, 2003, Washington D.C.

Ou, J. A., & Penman, S. H., Financial Statement Analysis and the Prediction of Stock Returns, *Journal of Accounting and Economics*, 1989, Vol.11, pp.295–329.

Qi, M., Predicting US Recessions with Leading Indicators via Neural Network Models, *International Journal of Forecasting*, 2001,Vol.17, pp.383–401.

Refenes, A. N., Azema-Barac, M. and Zapranis, A. D., Stock Ranking: Neural Networks vs Multiple Linear Regression, *IEEE International Conference on Neural Networks*, 1993, pp.1419-1426.

Refenes, A., Zapranis, A., & Francis, G., Stock Performance Modeling Using Neural Networks: a Comparative Study with Regression Models, *Neural Networks*, 1994, Vol.7, pp.375–388.

Reinganum, M., The Anatomy of a Stock Market Winner, *Financial Analysts Journal*, 1988, Vol.39, pp.16-28.

Roecker, Ellen B, Prediction Error and Its Estimation for Subset--Selected Models, *Technometrics*, 1991, Vol.33, pp.459-468.

Sagalyn, L.B., Real Estate Risk and The Business Cycle: Evidence form Security Markets, *Journal of Real Estate Research*, 1990. Vol.5, pp.203-219.

Salchenberger, L.M., Cinar, E.M. and Lash, N.A., Neural Networks: a New Tool for Predicting thrift Failures, *Decision Sciences*, 23(1992), pp.899-916.

Sing, T.F., Is property stock market efficient in the weak form? Singapore's evidence, *Journal of Financial Management of Property and Construction*, 2001,Vol. 6 No. 1, pp. 3-18.

Sing, T.F., K.H. Liow and Chan, W.J., Mean Reversion of Singapore Property Stock Prices towards Their Fundamental Values, *Journal of Property Investment & Finance*, 2002,Vol. 20(4), pp.374-387.

Tacz, G., Neural Network Forecasting of Canadian GDP Growth, *International Journal of Forecasting*, 2001, Vol.17, pp.57–69.

Tam, K.Y. and Kiang, M.Y., Predicting Bank Failures: a Neural Network Approach", *Applied Artificial Intelligence*, 4(1990), pp.265-282.

Tay, D.P.H. and Ho, D.K.K., Artificial Inteligence and the Mass Appraisal of Residential Apartment, *Journal of Property Valuation & Investment,* 1992, 10(2) pp.525-540.

Tay, D.P.H. and Ho, D.K.K., Intelligent Mass Appraisal*, Journal of Property Tax Assessment & Administration*, 1994, 1(1), pp. 5-25.

Tibshirani, R., Regression Shrinkage and Selection via the Lasso, *Journal of the Royal Statistical Society*, 1996, Vol.58, pp.267-288.

Titman, S. and Warga, A., Risk and the Performance of Real Estate Investment Trusts: A Multiple Index Approach, *Journal of American Real estate and Urban Economics Association*, 1986, Vol.14 (3), pp.414-431.

Udo, G., Neural Network Performance on the Bankruptcy Classification Problem, *Proceedings of the 15th Annual Conference on Computers and Industrial Engineering*, 25(1993), 1-4, 377-380.

Wansley, J., Roenfeldt, R., and Cooley, P., Abnormal Returns from Merger Profiles, *Journal of Financial and Quantitative Analysis*, 1983, Vol.18, pp.149-162.

Wilson, R.L. and Sharda, R., Bankruptcy Prediction Using Neural Networks, Decision Support Systems, 11(1994), pp.545-557.

APPENDIXES

Appendix 1 Neural Network Results of Bonvest Holdings

Methods	OLS Neural Networks				Logit Neural Networks			
Number	2000	Error	2001	Error	2000	Error	2001	Error
1	-0.9331	0.8191	-0.1607	0.0942	0.6919	-1.6919	0.2278	-1.2278
2	-0.7706	0.6566	0.3383	-0.4048	0.1206	-1.1206	0.1835	-1.1835
3	0.9641	-1.0780	-0.1750	0.1085	0.3392	-1.3392	0.1825	-1.1825
4	-0.3528	0.2388	0.1374	-0.2039	0.0036	-1.0036	0.3315	-1.3315
5	-0.7433	0.6293	-0.1661	0.0996	0.4822	-1.4822	0.1705	-1.1705
6	0.0355	-0.1495	-0.4587	0.3922	0.3392	-1.3392	0.2278	-1.2278
7	-0.6617	0.5477	0.0190	-0.0855	0.4822	-1.6919	0.1850	-1.1850
8	-0.3528	0.2388	-0.1565	0.0900	0.4822	-1.6919	0.2690	-1.2690
9	0.0184	-0.1323	-0.1661	0.0996	0.4822	-1.6919	0.1050	-1.1050
10	0.3554	-0.4694	-0.4773	0.4108	0.0445	-1.0445	0.0485	-1.0485
11	0.9753	-1.0893	-0.1034	0.0369	0.0335	-1.0335	0.4090	-1.4090
12	-0.9790	0.8650	0.5066	-0.5731	0.5177	-1.5177	0.1315	-1.1315
13	-0.8825	0.7685	0.8595	-0.9260	0.9116	-1.9116	0.2225	-1.2225
14	0.9554	-1.0694	0.0445	-0.1110	0.9910	-1.9910	0.3285	-1.3285
15	-0.3647	0.2507	-0.4456	0.3791	0.1695	-1.1695	0.2205	-1.2205
16	0.8533	-0.9673	-0.3873	0.3208	0.9820	-1.9820	0.2818	-1.2818
17	0.9152	-1.0292	-0.1661	0.0996	0.9980	-1.9980	0.4595	-1.4595
18	-0.2426	0.1286	0.2885	-0.3550	0.3236	-1.3236	0.5685	-1.5685
19	0.4226	-0.5366	-0.0338	-0.0327	0.6642	-1.6642	0.1821	-1.1821
20	-0.6732	0.5592	0.1239	-0.1904	0.7462	-1.7462	0.2465	-1.2465
21	-0.7851	0.6711	0.0190	-0.0855	0.9360	-1.9360	0.0055	-1.0055
22	0.3200	-0.4339	0.3383	-0.4048	0.8480	-1.8480	0.4188	-1.4188
23	0.9041	-1.0180	0.0192	-0.0857	0.0013	-1.0013	0.1275	-1.1275
24	-0.4201	0.3061	0.2881	-0.3546	0.9435	-1.9435	0.2470	-1.2470
25	0.8710	-0.9850	-0.7877	0.7212	0.6000	-1.6000	0.2995	-1.2995
26	0.3382	-0.4521	0.2638	-0.3303	0.9877	-1.9877	0.5947	-1.5947
27	0.9353	-1.0493	0.0445	-0.1110	0.2750	-1.2750	0.2065	-1.2065
28	-0.4141	0.3001	-0.0720	0.0055	0.2785	-1.2785	0.2875	-1.2875
29	-0.3295	0.2155	-0.3873	0.3208	0.5047	-1.5047	0.0180	-1.0180
30	0.0930	-0.2070	0.0748	-0.1413	0.4639	-1.4639	0.1060	-1.1060

Appendix 2 Neural Network Results of Bukit Semawang EST

Methods	OLS Neural Networks				Logit Neural Networks			
Number	2000	Error	2001	Error	2000	Error	2001	Error
1	0.0071	0.0299	-0.2788	0.5808	0.3502	0.6499	0.7585	0.2415
2	0.9580	-0.9211	0.1232	0.1788	0.7694	0.2307	0.8458	0.1543
3	0.1864	-0.1495	0.2354	0.0667	0.2664	0.7337	0.7962	0.2038
4	0.3508	-0.3138	0.1514	0.1507	0.4640	0.5360	0.8835	0.1166
5	-0.0392	0.0762	0.2416	0.0604	0.4554	0.5447	0.9575	0.0425
6	0.0026	0.0344	0.1960	0.1060	0.3875	0.6125	0.8010	0.1990
7	0.2648	-0.22780	-0.0664	0.3685	0.2815	0.7185	0.9590	0.0410
8	0.1982	-0.1613	0.1751	0.1269	0.3506	0.6495	0.5142	0.4858
9	0.7687	-0.7317	-0.0500	0.3520	0.3830	0.6170	0.5995	0.4005
10	-0.0877	0.1247	0.0205	0.2815	0.8642	0.1358	0.9800	0.0200
11	0.3764	-0.3394	0.0563	0.2457	0.3104	0.6896	0.5987	0.4014
12	0.0872	-0.0503	0.4790	-0.1770	0.7758	0.2242	0.8695	0.1305
13	-0.4387	0.4756	0.8251	-0.5231	0.4000	0.6000	0.8566	0.1434
14	0.0640	-0.0271	0.3981	-0.0960	0.2310	0.7690	0.9575	0.0425
15	0.3602	-0.3233	-0.1478	0.4499	0.2880	0.7120	0.7592	0.2409
16	0.4185	-0.3816	-0.0136	0.3157	0.3815	0.6185	0.8030	0.1970
17	0.4542	-0.4173	0.6063	-0.3043	0.7460	0.2540	0.9005	0.0995
18	0.2528	-0.2159	0.3595	-0.0574	0.9285	0.0715	0.5575	0.4425
19	0.0682	-0.0313	-0.2749	0.5769	0.5655	0.4345	0.6005	0.3995
20	0.2885	-0.2516	0.1467	0.1553	0.3500	0.6500	0.7685	0.2315
21	0.6210	-0.5841	0.0765	0.2256	0.2675	0.7325	0.8319	0.1681
22	0.6215	-0.5845	0.0004	0.3016	0.2650	0.7350	0.8460	0.1540
23	0.4212	-0.3842	-0.0839	0.3859	0.1268	0.8732	0.7415	0.2585
24	0.3602	-0.3233	0.4016	-0.0995	0.9165	0.0835	0.6200	0.3800
25	-0.4568	0.4938	-0.0098	0.3118	0.3552	0.6448	0.8550	0.1450
26	-0.6818	0.7187	0.4215	-0.1195	0.4950	0.5050	0.8800	0.1200
27	-0.1271	0.1641	0.0165	0.2855	0.9319	0.0681	0.6220	0.3781
28	0.4328	-0.3959	-0.2779	0.5799	0.3565	0.6435	0.5300	0.4700
29	0.1492	-0.1122	0.0847	0.2173	0.3565	0.6435	0.6500	0.3500
30	0.3548	-0.3178	0.3629	-0.0609	0.4398	0.5603	0.6600	0.3400

Appendix 3 Neural Network Results of Chemical INDL. (FE)

Methods	OLS Neural Networks				Logit Neural Networks			
Number	2000	Error	2001	Error	2000	Error	2001	Error
1	0.9249	-0.9953	0.5390	-0.3759	1.0000	-2.0000	0.3123	0.6877
2	0.3536	-0.4240	-0.5339	0.6969	1.0000	-2.0000	0.9959	0.0041
3	-0.4571	0.3867	0.7486	-0.5855	1.0000	-2.0000	0.0229	0.9772
4	0.3431	-0.4135	0.5981	-0.4350	1.0000	-2.0000	0.0810	0.9190
5	0.8288	-0.8992	0.9433	-0.7803	1.0000	-2.0000	0.2139	0.7861
6	0.0676	-0.1380	0.5226	-0.3595	1.0000	-2.0000	0.9580	0.0420
7	-0.6502	0.5798	0.4490	-0.2859	1.0000	-2.0000	0.0450	0.9550
8	0.1245	-0.1949	0.2637	-0.1007	1.0000	-2.0000	0.0263	0.9737
9	0.1338	-0.2042	-0.2991	0.4622	1.0000	-2.0000	0.4335	0.5665
10	-0.8264	0.7560	-0.2985	0.4616	0.4029	-1.4029	0.9755	0.0245
11	0.7518	-0.8222	0.9386	-0.7756	1.0000	-2.0000	0.9526	0.0474
12	0.7600	-0.8304	0.9967	-0.8336	1.0000	-2.0000	0.1145	0.8855
13	-0.7183	0.6479	-0.7913	0.9544	1.0000	-2.0000	0.1441	0.8560
14	-0.1615	0.0911	-0.1399	0.3030	1.0000	-2.0000	0.5445	0.4555
15	-0.1369	0.0665	-0.5355	0.6986	1.0000	-2.0000	0.7475	0.2525
16	0.1672	-0.2376	0.5216	-0.3585	1.0000	-2.0000	0.7850	0.2151
17	-0.1348	0.0644	0.9809	-0.8178	1.0000	-2.0000	0.7050	0.2950
18	0.8852	-0.9556	-0.9181	1.0812	1.0000	-2.0000	0.0285	0.9715
19	0.1227	-0.1931	-0.0585	0.2216	1.0000	-2.0000	0.9730	0.0270
20	0.9336	-1.0040	-0.5362	0.6993	1.0000	-2.0000	0.3081	0.6920
21	0.0073	-0.0777	0.9435	-0.7804	1.0000	-2.0000	0.9935	0.0065
22	-0.8407	0.8776	0.8645	-0.7014	1.0000	-2.0000	0.0121	0.9879
23	-0.3158	0.3528	-0.8345	0.9975	1.0000	-2.0000	0.8795	0.1206
24	0.1488	-0.1118	-0.5737	0.7368	1.0000	-2.0000	0.7260	0.2740
25	0.0225	0.0145	-0.15798	0.3211	1.0000	-2.0000	0.0475	0.9525
26	-0.4108	0.4478	0.1599	0.0032	0.9979	-1.9979	0.9900	0.0100
27	-0.9601	0.9971	0.1290	0.0341	1.0000	-2.0000	0.9600	0.0400
28	-0.7184	0.7554	-0.6380	0.8011	1.0000	-2.0000	0.2335	0.7665
29	-0.2643	0.3013	0.1410	0.0221	0.9995	-1.9995	0.0822	0.9179
30	-0.6658	0.7027	-0.2737	0.4367	0.9995	-1.9995	0.8460	0.1540

Appendix 4 Neural Network Results of City Development

Methods	OLS Neural Networks				Logit Neural Networks			
Number	2000	Error	2001	Error	2000	Error	2001	Error
1	0.4018	-0.4619	0.1393	0.0216	0.7670	-1.7670	0.5126	0.4874
2	0.1738	-0.2339	-0.1450	0.3060	0.5751	-0.0336	0.4855	0.5145
3	0.0176	-0.0777	-0.4390	0.5999	0.7414	-0.7414	0.5073	0.4928
4	0.0480	-0.1081	-0.4543	0.6153	0.9053	-0.9053	0.4010	0.5990
5	0.0704	-0.1305	-0.0327	0.1937	0.5890	-0.5890	0.0860	0.9140
6	0.5756	-0.6356	-0.0849	0.2459	0.9369	-0.9369	0.4802	0.5199
7	0.2519	-0.3119	0.0499	0.1110	0.8226	-0.8226	0.6187	0.3813
8	-0.0641	0.0041	0.3372	-0.1762	0.5558	-0.5558	0.4070	0.5930
9	-0.5123	0.4522	-0.4797	0.6407	0.5785	-0.5785	0.4310	0.5690
10	-0.3703	0.3102	-0.4593	0.6202	0.6526	-0.6526	0.6640	0.3361
11	0.1568	-0.2169	0.1307	0.0302	0.4387	-0.4387	0.4005	0.5995
12	0.0713	-0.1314	0.0823	0.0786	0.5872	-0.5872	0.4431	0.5570
13	0.0758	-0.1359	0.1007	0.0602	0.5088	-0.5088	0.1600	0.8400
14	0.6784	-0.7384	0.2916	-0.1306	0.4040	-0.4040	0.6608	0.3392
15	-0.3913	0.3313	0.2153	-0.0543	0.5211	-0.5211	0.3095	0.6905
16	0.0399	-0.1000	-0.2978	0.4587	0.5585	-0.5585	0.6893	0.3108
17	0.0397	-0.0997	0.1105	0.0504	0.7852	-0.7852	0.7245	0.2755
18	0.6155	-0.6756	0.0647	0.0962	0.7575	-0.7575	0.4589	0.5411
19	0.2743	-0.3344	0.3680	-0.2071	0.6812	-0.6812	0.7500	0.2500
20	-0.2153	0.1553	-0.0032	0.1641	0.5725	-0.5725	0.0000	1.0000
21	0.0971	-0.1572	0.1066	0.0543	0.1390	-0.1390	0.5530	0.4470
22	0.0675	-0.1275	-0.1865	0.3474	0.1433	-0.1433	0.9250	0.0750
23	0.1569	-0.2170	-0.0031	0.1641	0.6232	-0.6232	0.6393	0.3607
24	-0.0828	0.0227	0.0449	0.1161	0.3775	-0.3775	0.4965	0.5035
25	0.0480	-0.1081	-0.0565	0.2175	0.4870	-0.4870	0.9095	0.0905
26	0.7665	-0.8266	0.2481	-0.0872	0.4427	-0.4427	0.5674	0.4326
27	0.3522	-0.4123	-0.0131	0.1741	0.3770	-0.3770	0.6284	0.3716
28	-0.0641	0.0041	0.8715	-0.7106	0.0272	-0.0272	0.3360	0.6640
29	-0.0922	0.0321	-0.6519	0.8129	0.5788	-0.5788	0.4142	0.5858
30	0.0713	-0.1314	0.2916	-0.1306	0.1096	-0.1096	0.1694	0.8307

Appendix 5 Neural Network Results of Capitaland

Methods	OLS Neural Networks				Logit Neural Networks			
Number	2000	Error	2001	Error	2000	Error	2001	Error
1	0.9461	-0.9446	-0.1746	0.1249	0.1120	0.8880	0.6330	-1.6330
2	0.0509	-0.0494	-0.1205	0.0707	0.7778	0.2222	0.8875	-1.8875
3	0.8144	-0.8129	-0.0074	-0.0424	0.9127	0.0873	0.8065	-1.8065
4	0.2303	-0.2287	-0.0063	-0.0435	0.7189	0.2811	0.8717	-1.8717
5	-0.9142	0.9158	-0.6998	0.6501	0.6667	0.3334	0.8521	-1.8521
6	0.1451	-0.1435	0.1051	-0.1549	0.2742	0.7259	0.9470	-1.9470
7	0.7518	-0.7503	0.3848	-0.4346	0.9370	0.0630	0.6250	-1.6250
8	-0.7991	0.8006	-0.0156	-0.0342	0.4170	0.5830	0.5937	-1.5937
9	0.7022	-0.7006	-0.0077	-0.0421	0.7424	0.2577	0.7740	-1.7740
10	-0.1590	0.1606	-0.2597	0.2099	0.9755	0.0245	0.6785	-1.6785
11	0.9766	-0.9751	-0.5901	0.5403	0.6667	0.3333	0.3439	-1.3439
12	-0.9005	0.9021	-0.7896	0.7398	0.8060	0.1940	0.3840	-1.3840
13	0.4004	-0.3988	0.2813	-0.3311	0.7390	0.2610	0.9190	-1.9190
14	0.7705	-0.7690	-0.2057	0.1560	0.9147	0.0854	0.4105	-1.4105
15	-0.5255	0.5271	-0.1712	0.1214	0.0800	0.9200	0.6705	-1.6705
16	-0.7426	0.7441	0.7804	-0.8302	0.9969	0.0031	0.8500	-1.8500
17	0.6651	-0.6636	0.1261	-0.1759	0.7396	0.2604	0.6020	-1.6020
18	0.1463	-0.1447	-0.1205	0.0707	0.8580	0.1420	0.4327	-1.4327
19	-0.9835	0.9851	0.4337	-0.4835	0.6183	0.3818	0.6331	-1.6331
20	-0.4400	0.4415	-0.1712	0.1214	0.5793	0.4207	0.3588	-1.3588
21	-0.1362	0.1377	0.1261	-0.1759	0.6577	0.3423	0.6620	-1.6620
22	-0.8136	0.8152	-0.0074	-0.0424	0.8537	0.1463	0.8450	-1.8450
23	-0.0689	0.0705	0.0832	-0.1330	0.9606	0.0395	0.7932	-1.7932
24	0.9867	-0.9852	-0.6998	0.6501	0.9665	0.0335	0.7500	-1.7500
25	-0.2375	0.2391	0.2046	-0.2544	0.8005	0.1995	0.4260	-1.4260
26	0.7174	-0.7159	-0.5739	0.5241	0.9745	0.0255	0.8780	-1.8780
27	0.3292	-0.3276	-0.0156	-0.0342	0.9373	0.0628	0.6380	-1.6380
28	0.6933	-0.6918	-0.0077	-0.0421	0.0850	0.9150	0.7450	-1.7450
29	-0.5964	0.5980	-0.0411	-0.0086	0.9186	0.0814	0.7095	-1.7095
30	0.5589	-0.5574	-0.0075	-0.0423	0.8558	0.1442	0.5480	-1.5480

Appendix 6 Neural Network Results of Hong Fok Corporation

Methods	OLS Neural Networks				Logit Neural Networks			
Number	2000	Error	2001	Error	2000	Error	2001	Error
1	-0.9255	1.0359	-0.1248	-0.0437	0.2304	0.7696	0.2014	0.7986
2	-0.6583	0.7688	-0.0842	-0.0843	0.7406	0.2594	0.1238	0.8762
3	0.9821	-0.8717	0.8806	-1.0491	0.5295	0.4705	0.2234	0.7766
4	0.8816	-0.7712	-0.9802	0.8117	0.8395	0.1605	0.0624	0.9377
5	0.6153	-0.5049	0.8810	-1.0495	0.9543	0.0458	0.2476	0.7525
6	0.6987	-0.5882	-0.0902	-0.0783	0.5920	0.4080	0.2155	0.7845
7	-0.3815	0.4919	-0.2127	0.0442	0.8029	0.1972	0.1752	0.8249
8	0.4781	-0.3677	0.4428	-0.6113	0.0230	0.9770	0.2180	0.7820
9	0.9975	-0.8871	-0.1364	-0.0321	0.8211	0.1789	0.4975	0.5025
10	-0.0001	0.1105	-0.2100	0.0415	0.0110	0.9890	0.5466	0.4534
11	-0.8858	0.9962	0.8716	-1.0401	0.1515	0.8485	0.9240	0.0760
12	0.1651	-0.0547	-0.0316	-0.1369	0.9932	0.0068	0.8294	0.1707
13	-0.3263	0.4367	-0.9157	0.7472	0.4984	0.5016	0.7456	0.2544
14	0.2043	-0.0939	0.0262	-0.1946	0.0587	0.9414	0.0353	0.9647
15	-0.3333	0.4437	0.7859	-0.9544	0.0081	0.9920	0.6385	0.3615
16	-0.9743	1.0847	0.3006	-0.4691	0.0576	0.9425	0.7175	0.2825
17	-0.9314	1.0418	-0.3517	0.1832	0.2332	0.7669	0.1375	0.8625
18	0.2908	-0.1804	-0.4814	0.3130	0.7406	0.2594	0.6135	0.3865
19	0.1283	-0.0179	-0.1008	-0.0676	0.6845	0.3155	0.9500	0.0500
20	-0.2323	0.3427	0.8784	-1.0469	0.0389	0.9612	0.4755	0.5245
21	-0.0224	0.1329	0.8280	-0.9965	0.8395	0.1605	0.4124	0.5876
22	-0.5553	0.6657	0.8721	-1.0405	0.9543	0.0458	0.4998	0.5002
23	0.9607	-0.8503	0.5111	-0.6796	0.2060	0.7940	0.0150	0.9850
24	0.2627	-0.1523	-0.3504	0.1819	0.9303	0.0697	0.6080	0.3920
25	0.9664	-0.8560	-0.0450	-0.1235	0.8029	0.1972	0.5855	0.4145
26	0.3190	-0.2086	-0.1089	-0.0596	0.0250	0.9750	0.2759	0.7242
27	0.8595	-0.7491	0.9929	-1.1614	0.9988	0.0012	0.0455	0.9545
28	0.8636	-0.7532	0.6930	-0.8615	0.9977	0.0023	0.0145	0.9855
29	-0.2951	0.4055	0.2036	-0.3721	0.0102	0.9898	0.5169	0.4832
30	-0.8622	0.9726	-0.4858	0.3173	0.0762	0.9238	0.5270	0.4731

Appendix 7 Neural Network Results of Keppel Land

Methods	OLS Neural Networks				Logit Neural Networks			
Number	2000	Error	2001	Error	2000	Error	2001	Error
1	0.5025	-0.5330	0.1535	-0.0570	0.0212	-1.0212	0.4848	0.5152
2	-0.7311	0.7007	-0.8764	0.9729	0.0195	-1.0195	0.4523	0.5477
3	0.4821	-0.5125	0.2182	-0.1217	0.1349	-1.1349	0.4848	0.5152
4	0.3148	-0.3452	0.0114	0.0851	0.0220	-1.0220	0.4848	0.5152
5	-0.1996	0.1692	-0.2132	0.3097	0.0910	-1.0910	0.4848	0.5152
6	-0.0840	0.0536	0.1439	-0.0475	0.0041	-1.0041	0.4848	0.5152
7	-0.6080	0.5775	0.3562	-0.2597	0.0383	-1.0383	0.4195	0.5805
8	-0.1753	0.1449	0.4636	-0.3671	0.1389	-1.1389	0.9059	0.0941
9	0.1311	-0.1615	0.8832	-0.7867	0.1440	-1.1440	0.4848	0.5152
10	-0.1396	0.1091	0.3565	-0.2600	0.5422	-1.5422	0.4848	0.5152
11	-0.2645	0.2341	0.2740	-0.1775	0.3218	-1.3218	0.4848	0.5152
12	0.0932	-0.1236	-0.1395	0.2360	0.3109	-1.3109	0.4848	0.5152
13	0.0277	-0.0582	0.7426	-0.6461	0.0067	-1.0067	0.4848	0.5152
14	-0.6505	0.6200	-0.5004	0.5969	0.0338	-1.0338	0.4848	0.5152
15	0.7415	-0.7719	0.4219	-0.3254	0.0500	-1.0500	0.4848	0.5152
16	0.1680	-0.1984	-0.2022	0.2987	0.0280	-1.0280	0.4848	0.5152
17	-0.0306	0.0002	0.2964	-0.1999	0.3385	-1.3385	0.2375	0.7625
18	0.1617	-0.1921	0.2339	-0.1375	0.0613	-1.0613	0.4847	0.5153
19	-0.3976	0.3672	-0.1148	0.2113	0.0660	-1.0660	0.4848	0.5152
20	-0.3404	0.3099	-0.7180	0.8145	0.2580	-1.2580	0.4848	0.5152
21	-0.1261	0.0957	0.1908	-0.0943	0.0292	-1.0292	0.4848	0.5152
22	0.3653	-0.3957	0.2950	-0.1985	0.2071	-1.2071	0.4848	0.5152
23	-0.1723	0.1419	0.4797	-0.3832	0.0254	-1.0254	0.4848	0.5152
24	-0.3334	0.3030	0.7388	-0.6423	0.0897	-1.0897	0.4848	0.5152
25	-0.0483	0.0178	-0.1451	0.2416	0.0849	-1.0849	0.4848	0.5152
26	-0.1931	0.1627	-0.5976	0.6941	0.0119	-1.0119	0.4848	0.5152
27	-0.2495	0.2191	0.6451	-0.5486	0.0212	-1.0212	0.4848	0.5152
28	-0.5439	0.5134	0.5453	-0.4489	0.0195	-1.0195	0.4848	0.5152
29	-0.2118	0.1814	-0.5907	0.6872	0.0047	-1.0048	0.4848	0.5152
30	0.1628	-0.1933	-0.4928	0.5893	0.0136	-1.0136	0.4848	0.5152

Appendix 8 Neural Network Results of Marco Polo DEV.

Methods	OLS Neural Networks				Logit Neural Networks			
Number	2000	Error	2001	Error	2000	Error	2001	Error
1	-0.3052	0.5329	-0.2640	0.5491	0.5234	0.4766	0.1854	0.8146
2	0.0410	0.1868	0.1153	0.1698	0.7396	0.2604	0.9440	0.0560
3	0.3184	-0.0907	0.1777	0.1074	0.2663	0.7338	0.8333	0.1667
4	0.9711	-0.7434	-0.0305	0.3156	0.5657	0.4343	0.8330	0.1670
5	-0.3494	0.5771	-0.0869	0.3720	0.6406	0.3595	0.8690	0.1310
6	0.0111	0.2166	0.3375	-0.0524	0.4516	0.5485	0.7118	0.2883
7	-0.4561	0.6838	-0.3527	0.6378	0.3758	0.6242	0.9677	0.0324
8	0.5994	-0.3717	0.0729	0.2122	0.8731	0.1270	0.8923	0.1078
9	0.9046	-0.6768	0.1375	0.1476	0.8474	0.1527	0.5294	0.4707
10	0.0023	0.2254	-0.1567	0.4418	0.0287	0.9713	0.4558	0.5442
11	0.2126	0.0151	0.3443	-0.0592	0.9625	0.0375	0.9116	0.0884
12	0.2083	0.0194	0.0353	0.2497	0.6815	0.3186	0.9221	0.0779
13	-0.8564	1.0841	0.1743	0.1108	0.7036	0.2965	0.8280	0.1720
14	0.3595	-0.1318	0.3607	-0.0756	0.6667	0.3334	0.4987	0.5014
15	0.5215	-0.2938	0.1792	0.1059	0.3631	0.6369	0.8429	0.1572
16	-0.2065	0.4343	0.4021	-0.1170	0.7784	0.2216	0.9272	0.0729
17	0.1706	0.0572	0.1695	0.1156	0.9051	0.0950	0.9317	0.0684
18	-0.6902	0.9180	-0.3001	0.5851	0.5274	0.4727	0.7555	0.2445
19	-0.3270	0.5547	-0.0941	0.3792	0.7592	0.2408	0.7500	0.2500
20	-0.1219	0.3496	0.4934	-0.2084	0.1068	0.8932	0.9662	0.0338
21	0.3871	-0.1594	0.2542	0.0309	0.7449	0.2551	0.8550	0.1450
22	-0.3146	0.5424	-0.5679	0.8530	0.5936	0.4064	0.9280	0.0720
23	-0.1530	0.3807	0.4277	-0.1426	0.0632	0.9368	0.9530	0.0471
24	0.1871	0.0407	-0.1996	0.4846	0.7014	0.2986	0.5304	0.4696
25	-0.0354	0.2632	0.3638	-0.0787	0.5047	0.4954	0.9310	0.0690
26	-0.0674	0.2952	0.0176	0.2674	0.5047	0.4954	0.8253	0.1747
27	0.0137	0.2140	0.0891	0.1959	0.4651	0.5350	0.7222	0.2778
28	-0.8757	1.1035	0.6386	-0.3535	0.9337	0.0663	0.8330	0.1670
29	-0.9441	1.1719	-0.8738	1.1589	0.5754	0.4246	0.9757	0.0243
30	-0.1419	0.3696	-0.2743	0.5594	0.6266	0.3734	0.9980	0.0020

Appendix 9 Neural Network Results of MCL Land

Methods	OLS Neural Networks				Logit Neural Networks			
Number	2000	Error	2001	Error	2000	Error	2001	Error
1	0.5163	-0.5498	-0.3736	0.7393	0.3586	-1.3586	0.0095	0.9905
2	-0.0089	-0.0245	0.3989	-0.0332	0.0648	-1.0648	0.0015	0.9986
3	-0.6233	0.5898	-0.1404	0.5062	0.9384	-1.9384	0.0280	0.9720
4	-0.3103	0.2768	0.2001	0.1656	0.0123	-1.0123	0.1573	0.8428
5	-0.2472	0.2137	-0.0044	0.3701	0.0120	-1.0120	0.0108	0.9892
6	-0.5309	0.4974	-0.7330	1.0987	0.5406	-1.5406	0.1221	0.8779
7	0.3563	-0.5309	0.3502	0.0155	0.5821	-1.5821	0.0069	0.9932
8	-0.6209	0.5874	0.4784	-0.1127	0.2338	-1.2338	0.3119	0.6882
9	0.8757	-0.9092	0.5710	-0.2052	0.3463	-1.3463	0.0342	0.9659
10	-0.9628	0.9294	0.0416	0.3241	0.0202	-1.0202	0.1276	0.8725
11	0.2471	-0.2805	-0.1586	0.5244	0.8591	-1.8591	0.0126	0.9874
12	-0.7638	0.7304	0.0420	0.3237	0.7677	-1.7677	0.0445	0.9555
13	0.7874	-0.8208	0.0428	0.3230	0.3052	-1.3052	0.0080	0.9920
14	0.8535	-0.8870	0.1637	0.2020	0.3257	-1.3257	0.4352	0.5649
15	-0.1022	0.0688	-0.2727	0.6384	0.4814	-1.4814	0.0705	0.9295
16	-0.8085	0.7751	0.4565	-0.0907	0.4814	-1.4814	0.0220	0.9781
17	-0.1700	0.1365	0.1385	0.2273	0.5624	-1.5624	0.0372	0.9628
18	0.1650	-0.1985	-0.0959	0.4617	0.9291	-1.9291	0.0790	0.9210
19	-0.1855	0.1521	-0.5097	0.8754	0.3133	-1.3133	0.3119	0.6882
20	-0.6946	0.6611	-0.0618	0.4275	0.8540	-1.8540	0.0345	0.9655
21	0.9533	-0.9867	0.7627	-0.3970	0.2090	-1.2090	0.9645	0.0356
22	-0.8705	0.8371	-0.5525	0.9182	0.1861	-1.1861	0.0126	0.9874
23	-0.7947	0.7613	0.9458	-0.5801	0.9103	-1.9103	0.0556	0.9444
24	-0.0054	-0.0280	0.7075	-0.3418	0.8990	-1.8990	0.0488	0.9512
25	0.8620	-0.8955	0.0350	0.3307	0.9138	-1.9138	0.0175	0.9825
26	-0.8340	0.8006	0.5179	-0.1522	0.0110	-1.0110	0.0173	0.9827
27	0.2235	-0.2569	-0.1894	0.5552	0.0322	-1.0322	0.0091	0.9909
28	-0.3897	0.3562	-0.5128	0.8785	0.6262	-1.6262	0.3226	0.6774
29	0.0089	-0.0424	-0.5007	0.8664	0.7470	-1.7470	0.1565	0.8435
30	0.1249	-0.1583	-0.9345	1.3003	0.9451	-1.9451	0.0782	0.9218

Appendix 10 Neural Network Results of Orchard Parade HDG

Methods	OLS Neural Networks				Logit Neural Networks			
Number	2000	Error	Number	2000	Error	Number	2000	Error
1	-0.8962	0.6567	0.6802	-0.5797	0.7132	-1.7132	0.4279	0.5722
2	-0.8325	0.5931	-0.0986	0.1992	0.5631	-1.5631	0.0310	0.9691
3	-0.5418	0.3024	0.2834	-0.1829	0.3927	-1.3927	0.7893	0.2108
4	0.7684	-1.0078	0.1016	-0.0011	0.8540	-1.8540	0.8875	0.1125
5	-0.3454	0.1060	0.1484	-0.0478	0.3908	-1.3908	0.6338	0.3663
6	-0.0642	-0.1752	0.3224	-0.2219	0.5275	-1.5275	0.6105	0.3895
7	0.3495	-0.5889	0.5669	-0.4664	0.2461	-1.2461	0.3532	0.6468
8	-0.0957	-0.1437	-0.3098	0.4103	0.2166	-1.2166	0.6159	0.3841
9	0.2694	-0.5088	-0.6917	0.7922	0.5286	-1.5286	0.6730	0.3270
10	0.0730	-0.3124	0.3180	-0.2175	0.7885	-1.7885	0.1820	0.8180
11	-0.7322	0.4928	-0.9304	1.0309	0.6624	-1.6624	0.9322	0.0679
12	-0.7193	0.4799	0.2032	-0.1027	0.9397	-1.9397	0.6373	0.3628
13	0.1977	-0.4371	0.9403	-0.8398	0.2410	-1.2410	0.7790	0.2211
14	0.2441	-0.4835	0.8577	-0.7572	0.2214	-1.2214	0.5833	0.4167
15	-0.3813	0.1418	0.7315	-0.6310	0.2777	-1.2777	0.6824	0.3177
16	0.4902	-0.7296	-0.6018	0.7023	0.6169	-1.6169	0.6680	0.3320
17	0.5451	-0.7845	0.1118	-0.0113	0.2500	-1.2500	0.6030	0.3971
18	0.0041	-0.2436	0.7368	-0.6363	0.6684	-1.6684	0.7560	0.2440
19	0.5192	-0.7586	-0.1458	0.2463	0.4842	-1.4842	0.3633	0.6368
20	0.3025	-0.5419	-0.8202	0.9207	0.5105	-1.5105	0.2892	0.7108
21	-0.7647	0.5253	0.2716	-0.1711	0.8861	-1.8861	0.7841	0.2159
22	-0.2083	-0.0312	-0.9404	1.0409	0.5304	-1.5304	0.3140	0.6861
23	-0.5667	0.3273	0.7649	-0.6644	0.8274	-1.8274	0.2778	0.7222
24	0.6440	-0.8834	0.2351	-0.1346	0.3258	-1.3258	0.6646	0.3355
25	0.0310	-0.2704	0.0304	0.0701	0.8092	-1.8092	0.8266	0.1734
26	-0.0423	-0.1972	-0.7446	0.8451	0.5521	-1.5521	0.1898	0.8102
27	0.3560	-0.5954	0.8772	-0.7767	0.5286	-1.5286	0.4826	0.5174
28	-0.4056	0.1662	0.3323	-0.2318	0.3549	-1.3549	0.6452	0.3548
29	0.1694	-0.4088	-0.0923	0.1928	0.8037	-1.8037	0.5253	0.4747
30	0.0573	-0.2967	0.0123	0.0883	0.9397	-1.9397	0.9841	0.0160

Appendix 11 Neural Network Results of Singapore Land

Methods	OLS Neural Networks				Logit Neural Networks			
Number	2000	Error	2001	Error	2000	Error	2001	Error
1	-0.6515	1.0476	-0.1679	0.3893	0.4918	0.5082	0.1639	0.8362
2	-0.1417	0.5378	0.6513	-0.4299	0.4025	0.5976	0.9754	0.0246
3	0.4153	-0.0192	0.1206	0.1007	0.7887	0.2113	0.3711	0.6289
4	0.3107	0.0854	-0.5711	0.7925	0.5371	0.4629	0.2655	0.7345
5	-0.4711	0.8672	-0.0071	0.2285	0.1356	0.8644	0.2510	0.7490
6	-0.3189	0.7150	0.3397	-0.1184	0.5176	0.4824	0.0639	0.9362
7	-0.3275	0.7236	0.7822	-0.5608	0.5309	0.4691	0.1168	0.8832
8	-0.6833	1.0794	-0.0953	0.3167	0.3626	0.6374	0.1873	0.8128
9	-0.2037	0.5998	-0.0269	0.2483	0.0347	0.9654	0.8947	0.1053
10	-0.6875	1.0836	0.7213	-0.4999	0.2484	0.7517	0.4138	0.5863
11	0.5283	-0.1322	0.9990	-0.7777	0.1272	0.8729	0.3005	0.6995
12	-0.1224	0.5185	0.1538	0.0676	0.2416	0.7584	0.2788	0.7212
13	0.0752	0.3209	0.4778	-0.2564	0.6423	0.3577	0.5195	0.4805
14	-0.5735	0.9696	-0.6492	0.8706	0.1923	0.8077	0.4475	0.5525
15	-0.3208	0.7169	0.4858	-0.2645	0.2908	0.7092	0.1533	0.8468
16	0.2720	0.1241	0.1114	0.1100	0.3210	0.6790	0.8023	0.1977
17	0.3705	0.0256	-0.8010	1.0223	0.3332	0.6668	0.5465	0.4535
18	-0.0615	0.4576	0.7431	-0.5218	0.2785	0.7215	0.1395	0.8606
19	0.1018	0.2943	0.4910	-0.2697	0.8714	0.1286	0.3831	0.6169
20	-0.2799	0.6760	0.4840	-0.2627	0.5394	0.4606	0.5547	0.4453
21	-0.1417	0.5378	0.5722	-0.3508	0.4863	0.5137	0.5110	0.4890
22	0.0571	0.3390	-0.2523	0.4737	0.6997	0.3003	0.3033	0.6967
23	0.0764	0.3197	-0.7502	0.9715	0.3270	0.6730	0.6557	0.3444
24	0.2341	0.1620	0.6298	-0.4084	0.4380	0.5620	0.3482	0.6518
25	-0.6833	1.0794	-0.0707	0.2921	0.2130	0.7870	0.1979	0.8021
26	0.6233	-0.2272	-0.7822	1.0035	0.2729	0.7271	0.3837	0.6164
27	-0.6383	1.0344	-0.2207	0.4420	0.6470	0.3530	0.6935	0.3065
28	0.0002	0.3959	0.1858	0.0356	0.5515	0.4485	0.3133	0.6867
29	0.1234	0.2727	0.1874	0.0340	0.3669	0.6331	0.5676	0.4325
30	-0.1224	0.5185	-0.8505	1.0718	0.1585	0.8415	0.0000	1.0000

Appendix 12 Neural Network Results of United Overseas Land

Methods	OLS Neural Networks				Logit Neural Networks			
Number	2000	Error	Number	2000	Error	Number	2000	Error
1	-0.7756	1.3276	-0.1611	0.5211	0.9999	0.0001	1.0000	0.0000
2	-0.7515	1.3035	0.8752	-0.5153	0.9992	0.0008	1.0000	0.0000
3	-0.2710	0.8230	-0.1777	0.5376	0.9999	0.0001	0.9867	0.0133
4	0.2964	0.2556	-0.5861	0.9461	0.9993	0.0007	0.9994	0.0006
5	-0.7976	1.3496	0.4311	-0.0711	0.9997	0.0003	1.0000	0.0000
6	0.4435	0.1085	0.3171	0.0429	1.0000	0.0000	1.0000	0.0000
7	0.8906	-0.3387	0.4590	-0.0990	0.9997	0.0004	0.9990	0.0010
8	0.8005	-0.2485	0.6912	-0.3312	0.9996	0.0004	0.9992	0.0008
9	0.3415	0.2105	0.2752	0.0848	0.9983	0.0017	0.9998	0.0002
10	0.5937	-0.0418	0.5785	-0.2185	1.0000	0.0000	0.9995	0.0006
11	0.9231	-0.3712	-0.1081	0.4680	1.0000	0.0000	0.9989	0.0011
12	0.1788	0.3731	0.7753	-0.4153	0.9998	0.0002	0.9984	0.0016
13	-0.0801	0.6321	0.6941	-0.3341	0.9995	0.0005	0.9939	0.0061
14	-0.0158	0.5678	0.8029	-0.4429	0.9989	0.0011	0.9939	0.0061
15	-0.1953	0.7473	0.1163	0.2437	0.9955	0.0046	0.9983	0.0017
16	-0.5952	1.1471	-0.4651	0.8250	0.9959	0.0041	0.9928	0.0072
17	-0.8559	1.4079	-0.1675	0.5274	0.9983	0.0017	0.9923	0.0077
18	-0.2174	0.7694	0.9522	-0.5922	1.0000	0.0000	0.9884	0.0116
19	0.7206	-0.1686	0.4623	-0.1024	0.9999	0.0001	0.9839	0.0161
20	-0.5335	1.0855	0.7276	-0.3677	0.9951	0.0049	0.9939	0.0061
21	0.9731	-0.4211	0.4372	-0.0773	1.0000	0.0000	0.9984	0.0016
22	-0.1693	0.7213	-0.7085	1.0685	0.9999	0.0001	0.9977	0.0023
23	-0.3234	0.8754	0.1122	0.2477	0.9973	0.0027	1.0000	0.0000
24	-0.3361	0.8881	0.6055	-0.2455	0.9996	0.0004	0.9993	0.0008
25	-0.2648	0.8168	0.6168	-0.2568	0.9993	0.0008	0.9973	0.0028
26	0.8665	-0.3145	-0.5288	0.8888	1.0000	0.0000	0.9917	0.0083
27	0.9138	-0.3618	-0.4732	0.8331	0.9963	0.0037	0.9978	0.0022
28	0.8334	-0.2815	-0.2604	0.6205	0.9999	0.0001	0.9838	0.0163
29	0.3864	0.1656	0.2315	0.1285	0.9770	0.0230	0.9772	0.0229
30	0.0838	0.4682	0.5350	-0.1750	0.9863	0.0137	0.9994	0.0006

Appendix 13 Neural Network Results of Wing Tai Holdings

Methods	OLS Neural Networks				Logit Neural Networks			
Number	2000	Error	Number	2000	Error	Number	2000	Error
1	0.1890	-0.4150	-0.9254	1.1877	0.6444	-1.6444	0.7710	0.2291
2	-0.2856	0.0596	0.5966	-0.3343	0.1360	-1.1360	0.9820	0.0180
3	0.1948	-0.4207	-0.9436	1.2059	0.3530	-1.3530	0.0763	0.9238
4	0.1148	-0.3408	0.8939	-0.6316	0.6069	-1.6069	0.9465	0.0535
5	-0.3387	0.1127	-0.3996	0.6619	0.8200	-1.8200	0.0289	0.9711
6	-0.3836	0.1576	-0.8325	1.0948	0.3530	-1.3530	0.0026	0.9975
7	-0.1493	-0.0766	0.6818	-0.4195	0.3582	-1.3582	0.4501	0.5499
8	0.1368	-0.3628	0.2958	-0.0335	0.0776	-1.0776	0.9712	0.0289
9	0.7377	-0.9636	0.9838	-0.7215	0.2993	-1.2993	1.0000	0.0000
10	0.0707	-0.2967	0.0543	0.2081	0.3434	-1.3434	0.9991	0.0010
11	0.0395	-0.2655	0.9688	-0.7065	0.7190	-1.7190	0.5036	0.4965
12	-0.2376	0.0116	-0.1803	0.4426	0.1580	-1.1580	0.9427	0.0574
13	-0.0293	-0.1967	-0.5032	0.7655	0.2832	-1.2832	0.8813	0.1187
14	0.6298	-0.8558	-0.8564	1.1180	0.7253	-1.7253	0.4400	0.5600
15	0.0169	-0.2429	-0.9787	1.2410	0.5042	-1.5042	0.0389	0.9612
16	-0.2444	0.0184	0.4774	-0.2150	0.3300	-1.3300	0.6398	0.3603
17	-0.1036	-0.1224	-0.8486	1.1109	0.6471	-1.6471	0.4611	0.5389
18	0.0728	-0.2987	-0.7034	0.9658	0.2855	-1.2855	0.0135	0.9865
19	-0.0095	-0.2165	-0.5152	0.7775	0.6608	-1.6608	0.0498	0.9503
20	0.2993	-0.5253	0.7935	-0.5312	0.2964	-1.2964	0.9820	0.0180
21	-0.9451	0.7191	0.3366	-0.0743	0.4703	-1.4703	0.9720	0.0280
22	-0.4246	0.1987	0.9760	-0.7137	0.4635	-1.4635	0.9664	0.0337
23	0.1374	-0.3634	-0.9810	1.2433	0.2585	-1.2585	0.7435	0.2565
24	0.1303	-0.3563	-0.9439	1.2062	0.5823	-1.5823	0.0110	0.9890
25	-0.2095	-0.0165	-0.9958	1.2582	0.4795	-1.4795	0.3780	0.6220
26	-0.0685	-0.1575	-0.1795	0.4419	0.3029	-1.3029	0.8750	0.1250
27	-0.3592	0.1332	-0.9567	1.2190	0.6860	-1.6860	0.6591	0.3409
28	-0.6486	0.4226	-0.0496	0.3119	0.4173	-1.4173	0.1235	0.8765
29	0.3504	-0.5764	0.2822	-0.0199	0.5089	-1.5089	0.0975	0.9025
30	0.7528	-0.9788	0.5222	-0.2599	0.6541	-1.6541	0.6710	0.3290