Stock Market Predictions Based on Quantified Intermarket Influences

Chandima D. Tilakaratne

(MIT (by Research; Ballarat), MSc, BSc (Colombo))

This thesis is submitted in total fulfilment of the requirements for the degree of PhD

School of Information Technology and Mathematical Sciences,

University of Ballarat

PO Box 663

University of Ballarat

Ballarat, Victoria 3353

Australia

September 2007

Abstract

This study aims to make predictions about the Australian All Ordinary Index (AORD). The following two types of predictions are considered: (1) predicting the direction (up or down) of the Close price; and, (2) predicting whether it is best to buy, hold or sell. A novel approach, which heavily involves global optimization, is adopted for predictions.

This thesis investigates different methods of incorporating intermarket influences for predictions. It proposes a novel method for quantifying stock market influences from a set of potential influential markets on a given dependent market, by maximising the rank correlation between the markets of interest. The possible intermarket influence from the world's major stock markets on the AORD was quantified using this method. The ways of using quantified intermarket influence for predictions were investigated.

The direction of the Close price of the AORD was predicted using feedforward neural networks (FNNs). When predicting whether it is best to buy, hold or sell, to overcome the difficulties caused due to the imbalanced distribution of data (as a result of considering the hold class), this thesis introduces some neural network algorithms. These new algorithms use modified error functions and were trained with a global optimization algorithm.

The results relevant to both types of predictions suggest that the quantified intermarket influences on the AORD can be effectively used. This is an indication of the effectiveness of the proposed approach for prediction.

Statement of Authorship

Except where explicit reference is made in the text of the thesis, this thesis contains no material published elsewhere or extracted in whole or in part from a thesis by which I have qualified for or been awarded another degree or diploma. No other persons work has been relied upon or used without due acknowledgement in the main text and reference on the thesis.

Chandima D. Tilakaratne

24 September 2007

List of Publications

1. Tilakaratne, C. D., Mammadov M. A. & Morris, S. A. (2007). Effectiveness of Using Quantified Intermarket Influence for Predicting Trading Signals of Stock Markets, *Proceedings of the Sixth Australian Data Mining Conference (AusDM 2007)*, *Conferences in Research and Practice in Information Technology (CRPIT)*, 70.

2. Tilakaratne, C. D., Morris, S. A., Mammadov M. A. & Hurst, C. P. (2007). Quantification of Intermarket Influence on the Australian All Ordinary Index Based on Optimization Techniques, *ANZIAM Journal*, 48, C104-C118.

3. Tilakaratne, C. D., Morris, S. A., Mammadov M. A. & Hurst, C. P. (2007). Predicting Stock Market Index Trading Signals Using Neural Networks, *Proceedings of the 14th Annual Global Finance Conference (GFC 2007)*, Melbourne, Australia.

4. Tilakaratne, C. D., Mammadov M. A. & Hurst, C. P. (2006). Quantification of Intermarket Influence Based on the Global Optimization and Its Application for Stock Market Prediction, *Proceedings of the International Workshop on Integrating AI and Data Mining (AIDM'06)*, Hobart, Tasmania, Australia. `http://doi.ieeecomputersociety.org/10.1109/AIDM.2006.14`

5. Tilakaratne, C. D. (2006). A Study of Intermarket Influence on the Australian All Ordinary Index at Different Time Periods. *Proceedings of the 2nd Australian Business and Behavioural Seiences Association (ABBSA) International Conference*, Adelaide, Australia.

6. Pan, H., Tilakaratne, C. & Yearwood, J. (2004). Intermarket Influence Analysis using Asymmetrical Dependence Test and an Application for Australia and G7 Industrial Countries. *Proceedings of the 1st International Workshop on Intelligent Finance*, Melbourne, Australia.

7. Pan, H., Tilakaratne, C., Yearwood, J. & C. Hurst (2004). Asymmetrical Dependence Test for Intermarket Influence Analysis. *Proceedings of the 6th International Conference on Optimization Techniques and Applications*, Ballarat, Australia.

8. Pan, H.P., Tilakaratne, C. and Yearwood, J. (2003). Predicting the Australian Stock Market Index Using Neural Networks Exploiting Dynamical Swings and Intermarket Influences. *Lecture Notes in Artificial Intelligence*, 2903, 327-338, Springer; Updated version in *Journal of Research and Practice in Information Technology*, 37(1) (2005) 145-160. `https://www.acs.org.au/jrpit/JRPIT_Volumes.html`

Papers Submitted

1. Tilakaratne, C. D., Morris, S. A., Mammadov M. A. & Hurst, C. P., Stock Market Prediction based on Intermarket Influences.

2. Tilakaratne, C. D., Mammadov M. A. & Morris, S. A., New Algorithms for Predicting Trading Signals of Stock Market Indices.

Acknowledgements

I am grateful to my supervisors, Dr. Musa Mammadov, Professor Sidney Morris, and Dr. Cameron Hurst for their support, encouragement, and generosity in sharing their time, knowledge, and expertise throughout my PhD candidature. Also I would like to thank Dr. Heping Pan and Associate Professor John Yearwood for their support and advice at the beginning of this study.

I especially thank Ms. Rosemary Torney for undertaking the job as an editor of my research proposal. I extend my thanks to other staff members of the School of Information Technology and Mathematical Sciences of the University of Ballarat, who supported me in numerous ways.

My special thanks go to the staff of Research and Graduate Studies Office and the International Students Programs of the University of Ballarat.

My appreciation goes to the University of Ballarat for providing financial assistance during the period of my PhD studies. I also appreciate the University of Colombo for granting me study leave to conduct my research.

I am very grateful to Helen and Cecil for their support, encouragement and advice during my stay with them and also for providing me a homely environment to live in. I am also grateful to Mr. Alan Kealy who gave a warm welcomed to Australia.

Finally, I would like to thank my parents, sister, Head and the staff members of the Department of Statistics, University of Colombo, for their support and encouragement during my research.

Contents

List of Figures

List of Tables

Abbreviations

AORD - Australian All Ordinary Index

ARIMA - Autoregressive Integrated Moving Average

DP - Directional Profit

LDS - Discounted Least Squares

FCHI - French CAC 40 Index

FNN - Feedforawrd Neural Network

FTSE - UK FTSE 100 Index

GDAXI - German DAX Index

GMM - Generalised method of moments

GSPC - US S&P 500 Index

HSI - Hong Kong Hang Seng Index

IXIC -US Nasdaq Composite Index

N225 - Japanese Nikkei 225 Index

OLS - Ordinary Least Squares

PNN - Probabilistic Neural Network

SSEC - Chinese SSE Composite Index

STI - Singapore Straits Times Index

SVM - Support Vector Machine

TDP - Time Dependent Directional Profit

TWII - Taiwan TSEC Weighted Index

UK - United Kingdom

USA - United States of America

WDS - Weighted Directional Symmetry

Chapter 1

Introduction

1.1 Background

The background of this study is presented under two subsections: stock market prediction and intermarket influences.

1.1.1 Stock Market Prediction

The profitability of investing and trading in the stock market is directly proportional to its predictability. Therefore, predicting the direction of stock market indices is one of the most important issues in finance. Regarding the possibility of forecasting in this area there have been two major hypotheses widely adopted by financial academicians: the Efficient Market Hypothesis and the Random Walk Hypothesis.

The Efficient Market Hypothesis implies that in liquid markets, where prices are the result of unconstrained demand and supply equilibria, the current price should accurately reflect all the information that is available to the players in the market [2]. Future changes in prices can only be the result of 'news', which by definition is unpredictable.

The Random Walk Hypothesis states that price movements will not follow any patterns or trends and past price movements cannot be used to predict future movements [9].

According to these hypotheses, the possibility of predicting the movements in financial markets is very low. However, Fama [19] suggested that stock market price movement may show a weak form of efficient market hypothesis. Furthermore, many recent studies (for example [12, 16, 22, 45, 64, 67, 70, 74, 75, 100]) aimed at predicting the movements of stock market indices.

Until recently, the major forecasting methods used for financial prediction have been either technical or fundamental. Fundamental analysis studies the effect of supply and demand on the value of security while technical analysis believes that the stock market moves in trends and these trends can be found and exploited [9]. Fundamental methods rely on fundamental economic data such as retail sales, gold price, the industrial production index, and foreign currency exchange rates etc. as input variables to predict stock market indices whereas, technical methods rely mainly on market activity data and derivatives such as moving averages, momentum, relative strength index, etc. as input variables [59].

However, many of the techniques used by financial analysts are empirical in nature. These techniques have not been shown to be statistically valid, and may lack a rational explanation for their use [100]. Stock markets are influenced by many interrelated factors including the effects of economic, political and even psychological factors. These varied and diverse factors interact with each other in a complex fashion, and it is therefore very difficult to accurately forecast the movement of stock markets.

Recently, Murphy [58] suggested that intermarket technical analysis is a promising method for stock market prediction. If the performance of two markets are interrelated, then one market will be termed an *intermarket* of other [73].

1.1.2 Intermarket influences

In the last two decades, many changes such as, liberalisation of financial markets, improvements of information and communication technologies, and developments of trading facilities, have taken place in the world. Due to these changes, the scope of selection for financial investors and traders is increased. This enlargement of the scope of selection causes an integration of financial markets of many different countries of the world [25, 55, 66, 99]. As a result of this integration, the behaviour of the world's major stocks are interrelated. As mentioned in the previous section (Section 1.1.1), when the performance of two markets are interrelated, one market is called an intermarket of other [73].

If the lagged prices (or a derivative thereof) of a stock market index have a significant impact on the current price (or derivative of price) of a given stock market, then we define such an impact as the influence from the former market on the latter. Furthermore, if one market influences another, we call such an influence an *intermarket influence*. There may be a set of intermarkets that influence the behaviour of a target market. Therefore, we define *Intermarket Influence Analysis* as the study of relationships between the current price (or a derivative of price) of a dependent market and the lagged price (or a derivative thereof) of one or more influential markets [83, 84, 85]. The way one market influences another for a perceivable time period will be called an 'intermarket influence pattern'. This pattern may vary from one pair of markets to another, or it may vary from one time period to another for the same pair of markets. Intermarket influences (either positive or negative) may be reflected on the price itself, and/or one or more derivative properties of price such as trend (linear or non-linear), volatility [1], etc. of the target market.

Currently intermarket influence is an important consideration among investors and decision makers. Discovering intermarket influence patterns is useful in many applications such as market prediction, portfolio optimization and management, option pricing, and risk management. Intermarket technical analysis is a relatively recent area of technical analysis practised by professional analysts [60].

Poddig and Rehkugler [66] argued that most of the financial markets of major developed countries must be regarded as highly integrated and therefore, the traditional approach of modelling (an unique model with respect to asset or asset class of interest) would ignore valuable information. Instead of this traditional approach, a non-linear

[1] This is a measure of the dispersion in a probability density of stock market price returns.

analysis of integrated financial market is essential to understand the behaviour of these markets. Taking this matter into account, they proposed a model named the 'intermarket model'. They defined this model as the direct approach of modelling a system of interlinked markets by the use of a system of interdependent equations. Their results suggested that the intermarket model outperforms isolated (unique) market models.

The suggestions and the findings done by the past studies [60, 66] give a strong indication of the importance of taking the behaviour of foreign stock market indices into account, when studying the behaviour of a selected stock market index.

1.2 Motivation for the Study

Recent studies [63, 65, 66] have shown that the intermarket influences improve forecast accuracy. Furthermore, Olson and Mossaman [63] showed that during periods when macroeconomic variables are changing, correlations among interrelated markets pick up the changing market conditions faster than the lagged macroeconomic variables.

If a set of stock markets are interrelated, each stock market in this set can be considered as a part of a single system. The influence from one integrated stock market on a dependent market may include the influence from one or more stock markets on the former. Therefore, in order to estimate the direct influence from one market to another, intermarket influence needs to be quantified. However, no techniques for the quantification of intermarket influences were introduced in the literature.

Discovering and formalising intermarket influence patterns is likely to prove extremely useful in many applications such as market predictions. Some past studies (for example [26, 63, 65, 66]) incorporated the possible influence from one or more foreign stock markets together with other factors, to predict a selected stock market index. However these studies did not consider the quantified intermarket influences from the considered foreign stock markets. Therefore, stock market predictions using quantified intermarket influences as input variables, potentially provides a great opportunity for a PhD research.

1.3 Research Objectives

The objective of this study is to make predictions about the Australian All Ordinary Index (AORD) using the intermarket influences from the world's major stock market indices. The prediction is twofold:

1. Predicting the direction (up or down) of the Close price of day $(t+1)$ of the AORD;

2. Predicting whether it is best to buy, hold or sell (trading signals) on day $(t+1)$.

Such predictions are beneficial for short-term traders, since they can make the correct investment decitions by looking at the predictions.

1.3.1 Research Problems

The previous section (Section 1.2) highlighted the importance of the quantification of intermarket influences. As mentioned in Section 1.2, no quantification technique for quantifying intermarket influences, is available in the literature. Hence, a technique needs to be introduced before starting to quantify intermarket influences on the AORD.

After quantifying the intermaket influence on the AORD, an investigation can be carried out to identify how the quantified intermarket influences can be effectively used for predictions. The existing methods and algorithms can be employed to do these predictions. However, these methods and algorithms may not be sufficient to address the prediction problem of interest, specifically the predicting the three trading signals. Therefore, their appropriateness for addressing the prediction problem of interest, needs to be examined and also their drawbacks need to be identified. Then the new algorithms for predicting the trading signals can be developed by overcoming these drawbacks.

Taking above matters into account, the main objective of the study can be elaborated to following research problems:

1. Develop a technique for quantifying intermarket influence from a set of potential influential markets on a given target market.

2. Quantify the intermarket influences from the world's major stock market indices on the AORD by applying the technique proposed in 1.

3. Predict the direction of the Close price of the AORD, and investigate whether and how the quantified intermarket influences can effectively be used for directional prediction.

4. Predict whether it is best to buy, hold or sell shares (trading signals), with the help of the existing methods and algorithms, and investigate whether and how the quantified intermarket influence can effectively be used for such predictions.

5. Investigate the efficiency of the methods and algorithms used in 4 and identify their shortcomings. Develop new algorithms which predict trading signals with higher accuracy.

1.4 Significance of the Study

This study is significant due to following reasons:

- It proposes a technique for quantifying intermarket influences from a set of potential influential markets on a given target market;

- It applies this proposed technique to quantify the intermarket influence from the world's major stock market indices on the AORD;

- It introduces new algorithms for predicting whether it is best to buy, hold or sell shares;

- Uses the quantified intermarket influences for directional prediction and predicting whether it is best to buy, hold or sell shares.

1.5 Outline of the Thesis

This thesis consists of eight chapters. The next chapter (Chapter 2) reviews the literature relevant to this study. Chapter 3 discusses the methodology used to achieve the objectives. Chapter 4 proposes a technique for quantifying intermarket influences. This chapter also presents and discuses the quantification results related to the AORD. Chapter 5 aims to predict the direction of the Close price of the AORD with a special view of investigating the effectiveness of using quantified intermarket influences for directional prediction. The next chapter (Chapter 6) focuses on predicting whether it is best to buy, hold or sell, with a special view of investigating the effectiveness of using quantified intermarket influences for such predictions. This chapter uses the existing methods and algorithms for such predictions and it investigates the appropriateness of these methods and algorithms for addressing the prediction problem of interest: predicting the three trading signals. Chapter 7 focuses on developing algorithms for predicting whether it is best to buy, hold or sell, by addressing the issues arise when doing such predictions. It also investigates the efficiency of using the quantified intermarket influences to improve the prediction accuracy. Finally, Chapter 8 presents the conclusions and suggests future research directions.

Chapter 2

Literature Review

2.1 Introduction

This chapter reviews the literature related to stock market predictions with a special view on the directional prediction and the prediction of trading signals. The methods and algorithms used for the predictions and the measures used for evaluating the predictions by the past studies will be discussed. Also the shortcomings of these algorithms and how past studies overcame these drawbacks will be investigated. In addition to the above matters, the input features used by past studies for predictions will be reviewed.

2.2 Stock Market Predictions

Before the 1980s, attempts to model financial market data in order to predict future market directions were unsuccessful due to the inherent complexity of the data. The efficient market hypothesis claims that financial markets are a random time series and, therefore unpredictable on the basis of any amount of publicly available knowledge [46]. However, Fama [19] suggested that stock market price movement may show a weak form of efficient market hypothesis.

Until the late 1980s, most quantitative approaches used to test this hypothesis were based on linear time series modelling [93]. Chenoweth et al. [13] stated that it is very hard to find statistically significant market inefficiencies using standard linear time series modelling, since such linear approaches are not capable of identifying dynamic or non-linear relationships in financial data. Weiss and Kulikowski [91] suggested that an appropriate nonparametric machine-learning technique might be able to discover more complex non-linear relationships through supervised learning from examples. Such new approaches to financial modelling have been developed during the last two decades. Many recent studies (for instance [12, 16, 22, 45, 64, 66, 67, 70, 74, 75, 100]) used non-linear modelling techniques to stock market returns.

A majority of previous studies (for example [16, 23, 67, 75]) have specifically aimed at predicting the price levels (that is the value) of the stock market indices. Recently there has been a growing interest in prediction of the direction (up and down) of stock market indices [81]. When predictiong the price level, the error (deviation of the predicted value from the actual value) is taken as the measure of accuracy, whereas the number of times that correct direction was predicted is the main concern for directional predictions.

Some studies (for example [11, 98]) have suggested that trading strategies guided by forecasts on the direction of price change may be more effective and may lead to higher profits. Laboratory based experiments conducted by O'Connor et al. ([62] cited in [44]) demonstrated the usefulness of predicting the direction of change in price levels, that is the importance of ability to classify the future return as a gain or a loss. Leung et al. ([44] cited in [81]) found that the classification models based on the direction of stock returns outperformed the models based on the level of stock return in terms of both predictability and profitability .

In terms of practical applications (that is the higher predictability and the profitability), it is worth to focus on predicting the direction of a given stock market index, rather than predicting its level.

2.2.1 Directional Prediction

The literature reveals that there are two types of study which focus on prediction of the direction of stock markets. One type (say **Type A**) focused on predicting the future direction (up or down) of stock market indices (for instance [11, 26, 34, 35, 44, 74, 104]). The other type (say **Type B**) focused on predicting the price levels of the stock market indices. In the latter case, the prediction accuracy was evaluated by sign or direction accuracy of the predictions (for example [21, 23, 44, 45, 65, 66, 68, 95, 100]). All of these studies considered only two classes: either upward/downward trend or positive/negative sign.

Classification Models and Evaluation Measures Used by Type A Studies

Chen et al. [11] employed a probabilistic neural network (PNN) to forecast the sign (positive or negative) of the 3-month, 6-month and 12-month excess returns of the market index of the Taiwan Stock Exchange. They compared the predictive strength of the PNN with those of the Generalised methods of moments (GMM) with Kalman filter and random walk models. They used the hit rate (that is the percentage of predictions with the correct sign) to evaluate the predictions obtained by these three methods. The results evidenced that PNN outperformed other two models in all three prediction targets.

To forecast the weekly movement direction (up or down) of the Japanese NIKKEI 225 Index, Huang et al. [26] used a support vector machine (SVM) with a radial basis function as the kernel. Furthermore, they compared the performance of the SVM with those of linear discriminant analysis, quadratic discriminant analysis and elman backpropagation neural networks. The hit rate (in this case, the hit rate is the percentage of predictions with the correct direction) was used to evaluate the performance. Their results showed that the SVM outperforms the other classification methods.

Kim [34] compared the prediction ability of the SVM against the that of backpropagation (feedforward) neural networks and case base reasoning to predict the direction of change in the daily Korean composite stock price index (KOSPI). A radial basis function was used as the kernel function of these SVMs. He also used the hit rate as the evaluation measure of the prediction performance. The results suggested that the SVM outperforms other two models.

An arrayed probabilistic network (APN) was used by Kim and Chun [35] to predict the fractional change (up or down) of the Singapore Stock price index, from its current value. This APN is a combination of a number of PNNs and each of these PNNs classifies the change as belonging to a particular range or not. The final decision is taken by examining the outputs of individual PNNs. They compared the results of this APN with those of the recurrent neural networks, backpropagation neural networks and case base reasoning, by using a 'mistake chart' as well as the hit rate. This mistake chart plots Type II error versus Type I error of the predictions. By means of the hit rate, the APN outperforms the other models. However, case base reasoning tended to outperform the APN as well as other models when mistakes (Type I and II errors) were taken into consideration.

To predict stock trends (2%, 5% and 10% up move of the stock closing price, within the following 22 working days), Saad et al. [74] exploited three types of neural networks, time delay neural networks (TDNNs), recurrent neural networks (RNNs) and probabilistic neural networks (PNNs). They tested different values for the lost incurred from misclassification, L (Section 3.2.2) and also for the standard deviation, σ, of the gaussian distribution. The percentage of false alarms (the percentage of cases wrongly classified as an upward trend) was used as the measure of evaluation of the predictions. PNN gave a low false alarm percentage even for the stocks with low predictability.

Zemke [104] predicted whether the index value of the Warsaw Stock Exchange (WSE) one trading week ahead will be up or down, in relation to the current value. This study used backpropagation neural networks, naive bayesian classifier, and k-nearest neighbour genetic algorithms as classification techniques. When using the backpropagation neural networks, the up movement and the down movement were scaled as 0.8 and 0.2, respectively. The prediction performance of the classification models considered was evaluated by the hit rate. The results suggested that the k-nearest neighbour method outperforms the other classification models.

Unlike the studies discussed above, the study done by Leung et al. [44] focused on classifying the next day's direction (based on probability) as well as estimating the next day's price level of the stock market indices (the later part is discussed in the next subsection in detailed). As the classification models, they employed discriminant analysis, logit model, probit model and PNN. They used data from three stock market indices, namely the US S&P 500 Index, the UK FTSE 100 Index and the Japanese Nikkei 225 Index.

Leung et al. [44] used two measures to evaluate the performance of the predictive models: (1) the percentage of forecasts with the correct sign (hit rate); and, (2) the rate of return obtained by performing trading simulations. Two different trading strategies were employed for classification and level estimation models.

According to their results, the PNN gave the highest hit rate for the US S&P 500 Index and the UK FTSE 100 Index, while discriminant analysis produced the highest hit rate for the Japanese Nikkei 225 Index. PNN also yielded the highest rate of return for the US and the UK markets while that for the Japanese market was obtained by the discriminant analysis.

The above literature reveals that the most common algorithms used for the classification of future movement of stock market indices are PNN, SVM and feedforward (backpropagatoion) neural network (FNN). Also it is noteworthy that theses algorithms, particularly PNN and SVM, outperform traditional statistical models (such as discriminant analysis, random walk, logit and probit models), and also FNN models.

The hit rate (the percentage of predictions with the correct direction/sign) seems to be the most common measure used for evaluation of prediction performance of the models applied.

Level Estimation Models and the Evaluation Measures Used by Type B Studies

As noted in the previous sub section, Leung et al. [44] focused on predicting the price level of the stock market indices, in addition to the classification of trading signals. They used an adoptive exponential smoothing, vector autoregressive model with a Kalman filter, multivariate transfer function and a multi-layered feedforward neural network, as the level estimation models. The same measures (as those used to evaluate the classification results) were used to evaluate the performance of the models applied.

When comparing the performance of the level estimation models, the multi-layered FNNs produced the highest rates of return for all three markets. However, both the hit rate and the rate of return produced by the best classification models are higher than those corresponding to the best level estimation model. Hence they suggested that *the classification models are better than the level estimation models in terms of the predictability and the profitability.*

In order to investigated the profitability of a technical trading rule based on an artificial neural network, Fernando et al. [21] used the FNN to predict the relative return of the General Index of the Madrid Stock Market, 250 days ahead. If the predicted value is greater than zero, they considered the corresponding trading signal as a buy signal; otherwise it was considered as a sell signal. They used several measures to evaluate the forecast accuracy: hit rate, total return, ideal profit and Sharpe ratio (mean return of the trading strategy divided by its standard deviation). They suggested that the FNN trading strategy was more profitable than the buy-and-hold strategy during 'bear' and 'stable' market periods.

Gencay and Stengos [23] examined the prediction ability of the FNN against linear regression and GARCH-M models in terms of the sign accuracy of the predictions. They applied these models to predict the daily return series of the Dow Jones Industrial Average Index. Their results suggested that FNN models outperformed the other two models.

Although not common, genetic algorithms (GA) were also applied by the past studies for stock market prediction. For instance, Mahfoud and Mani [45] applied a combined GA and neural network model as well as GA and neural network as separate models to model the relative returns of the stocks traded in S&P 500 Index. The prediction ability of these models was evaluated in terms of directional correctness. They found that the combined model outperformed either algorithm individually.

Forecasts related to the Australian stock market are very rare. Pan et al. [65] investigated the predictability of FNN models for forecasting the direction of the AORD. They predicted the Close price index of the AORD and used sign correctness percentage as the measure of evaluation.

They argued that if the next day's relative return is zero or approximately zero, then there is no substantial difference between current day's and next day's Close prices, irrespective of the sign. To fix this problem, they introduced a threshold which helps to represent a 'no change' region. When the sign of the actual and the predicted values are different, they checked whether the absolute value of the difference between the actual and the predicted values, is less than this threshold. If so, they considered that the signs of the both values to be the same.

To compare the predictive accuracy of the stock returns produced by the neural networks with those obtained by the linear predictive models, Qi and Maddala [68] forecasted S&P 500 index returns using the FNN. Linear regression and random walk models were considered as the linear models to be compared. Several measures, including the direction accuracy (proportion of times the upward or downward movement is correctly predicted) and the proportion of times the sign (positive or negative) is correctly forecasted, were used as the measures of evaluation. FNN outperformed the two linear model in terms of the direction accuracy and the sign accuracy.

Wood and Dasgupta [95] also employed FNNs to predict the direction (trend) of the Morgan Stanley US Capital Market Index, one month ahead. The predictive power of the neural network models were compared with those of the linear regression and ARIMA models. These models predicted the value of the index and this predicted value was substituted in the criterion of identifying the correctly predicted direction. The hit rate was used as the measure of evaluation of predictive power. Their results suggest that, in terms of directional prediction, the FNN model outperforms the two alternative models.

An approach similar to Wood and Dasgupta [95] was used by Yao et al. [100] to test the forecasting ability of neural networks. They also employed FNN and ARIMA models to forecast the value of the Kuala Lumpur Stock Exchange Index (KLCI). The forecasts were evaluated by hit rates (in terms of accuracy of gradients and signs) as well as rate of returns obtained by performing trading simulations. The experiment results showed that the neural network model provides higher rate of returns compared to the ARIMA models.

According to the above literature, it is clear that the most common model used for value prediction is FNN models. The FNNs show better performance than linear models such as regression and random walk models, in predicting the value of stock market indices.

The most common measures of evaluation for level prediction models are the hit rate and the rate of return (these measures were used by the past studies independent of whether they predicted the direction of the price level or value of the price level).

However, apart from the study done by Pan et al. [65], none of the other studies (mentioned above) paid attention to the 'no change' region when estimating the directional accuracy. As argued by Pan et al. [65], *there is no significant change in the price level (compared to the previous day), if the predicted relative return is zero or close to zero. This matter indicates the necessity of introducing a threshold when estimating the direction accuracy.*

2.2.2 Predicting Trading Signals

Profitability of stock market trading is directly related to the prediction of trading signals. In the last few decades, there has been a growing number of studies attempting to predict the trading signals of financial market indices. Many past studies (for example [21, 88, 95, 100]) considered only two trading signals: buy and sell. Although not very common, some studies (for example [11, 13, 38, 40, 44, 56]) considered a third signal:hold.

Models Applied to Predict Trading Signals and Evaluation Measures

As mentioned in Section 2.2.1, Chen et al. [11] employed probabilistic neural network (PNN), GMM and random walk models to forecast the sign (positive or negative) of the 3-month, 6-month and 12-month excess returns of the Taiwan Stock Exchange Index. They applied different trading strategies for the PNN and other two models. The PNN was used to estimate the probability of a predicted return showing an upward trend and this probability was used to make the decision of trading. Unlike the case of the PNN, predicted value the GMM and random walk models were used to make the trading decision.

The performance of the models considered was evaluated in terms of profitability. The rate of return obtained by performing trading simulations was used for the evaluations. In trading simulations, they assumed that an investor invests a fixed amount of money at the beginning of each month in either risk-free bonds or the stock index fund, depending on the prediction results. This simulation was tested against a buy and holds strategy. This strategy assumes that the investor invests money in the stock index fund and hold till the end of the period. Results showed that the investor can gain profits by responding to the prediction results obtained by all three models. The trading strategies guided by PNNs were more profitable than those related to GMM and random walk models.

To forecast buy and sell signals of the S&P 500 Index, Chenoweth et al. [13] embedded some technical analysis knowledge into neural network. They used a threshold to define up and down trends of the index and combine this information with the average direction index (ADX) [17, 41]. When compared with the benchmark 'buy and hold' strategy, the trading system based on their neural network model was more profitable.

Fernando et al. [21] and Yao et al. [100] followed similar approaches. Instead of predicting the trading signals, they predicted the value of the indices by using FNNs and then used different criteria to classify the corresponding prediction as a buy or a sell signal. The rate of return obtained by performing trading simulations is among the number of measures of evaluation used.

Kohara et al. [38] employed FNN, recurrent neural network and multiple regression to predict the daily change in the Close price of the TOPIX (Tokyo Stock Exchange Price Index). The trading signals (buy and sell) were defined based on this predicted value. The neural network models outperformed the multiple regression model in terms of stock-trading profit, while the recurrent network model outperformed the FNN model.

To predict the trading signals of the Taiwan stock market Kuo [40] applied two separate models for two types of factors: (1) a fuzzy neural network to model quantitative factors; and, (2) a fuzzy Delphi to model qualitative factors. In order obtain the final prediction he integrated the decisions produced by the two models using a FNN model. The other main feature of this study is the consideration of a hold signal addition to the buy and sell signals. The predictions were evaluated by the number of buy and sell signals.

Unlike the other studies, Leung et al. [44] applied both the classification and level estimation methods to predict the trading signals of three stock market indices (Refer Section 2.2.1 and 2.2.1 for more details). They employed discriminant analysis, logit model, probit model and PNN as the classification models, while adoptive exponential smoothing, vector autoregressive model with Kalman filter, multivariate transfer function and multi-layered feedforward neural network were used as the level estimation models.

The performance of both the classification and level estimation models were evaluated by rate of return. For the classification models, if the probability of an upward (positive) movement is greater than 0.5, then the corresponding signal was identified as a buy signal. Otherwise, it was assumed that the corresponding signal was a non-buy signal and the money was invested in treasury bills, instead of buying shares. For the level estimation models, a different trading strategy was applied: if the predicted excess return was greater than zero, the corresponding signal was considered as a buy signal, otherwise the money was invested in treasury bills. They found that classification models are better than the level estimation models in terms of profitability

Mizuno et al. [56] employed a FNN model to predict trading signals, buy, sell and no change, of the Tokyo Stock Exchange Price Index (TOPIX). Their network model produced more accurate predictions in the most dominant class: the no change signal. Prediction results were evaluated by the ratio of accuracy of each type of signal (that is the ratio of correctly classified signals out of the total classified to a particular class).

Vanstone [88] used an artificial neural network model to identify trading signals in the Australian stock market. He predicted the value of the ASX200 Index using artificial neural networks. Then a selection criterion was followed to identify trading signals, buy and sell. In addition to the value of the index, this criterion takes signal strength into account.

The majority of past studies examine the profitability of the predictions (of trading signals) rather than their predictability. The most commonly adopted evaluation measure of profitability is the rate of return obtained by performing trading simulations. Few studies which evaluated the predictions in terms of predictability adopted the hit rate as the measure of evaluation.

Studies aiming at predicting the three trading signals, including the hold signal, are very rare in the literature. The literature does not provide evidence for such predictions related to the Australian stock market.

Criteria Used to Defining Trading Signals

To classify the trading signals, Fernando et al. [21] assumed that a predicted value (of index) greater than zero indicates a buy signal while this value less than zero indicates a sell signal. Yao et al. [100] followed two strategies to define the trading signal corresponding to the predicted value of the index:

Strategy 1

$$\text{if} \quad (\hat{x}_{t+1} - \hat{x}_t) > 0, \text{ then } \textit{buy} \text{ else } \textit{sell};$$

Strategy 2

$$\text{if} \quad (\hat{x}_{t+1} - x_t) > 0, \text{ then } \textit{buy} \text{ else } \textit{sell}$$

where x_t and $\hat{x}_t$ are the actual and the predicted values of the index at time t.

The main disadvantage of the criteria used in theses two studies [21, 100] *is the disregard of the cases which are zero. Neither of two studies mentioned what would be the trading action, if the predicted value* (*in reference to* [21]) *or the difference* (*in reference to* [100]) *is zero.*

Vanstone [88] argued the following criteria is suitable to define the trading signals:

Buy tomorrow if today's predicted value $> x'$, and today's predicted value $>$ yesterday's predicted value;

Sell tomorrow if today's predicted value $\leq x'$, and today's predicted value $<$ yesterday's predicted value;

where x' is the signal strength threshold chosen.

This criterion seems to be more practical than the criteria suggested in [21, 100]*. However, it is not practicable to make adjustments to it to include the hold class.*

Kohara et al. [38] classified the corresponding signal as buy (or sell), if the next day's positive (or negative) change in the stock market was larger than a preset value which represents a large change. The network model designed by Kuo [40] outputs a value which represents the trading signal. The corresponding trading signal was determined by comparing this value with an upper and a lower bound. Different values were tested for these boundaries and [0.2, 0.8] gave the best predictions of trading signals.

Similar to Kuo [40], Mizuno et al. [56] also applied two thresholds to define trading signals. If the predicted value was below 0.4, the corresponding signal was considered as a sell signal while if this value was above 0.6, then the corresponding signal was considered as a buy signal.

The criteria adopted by Chen et al. [11] is based on the probability of the predicted stock return being in an upward trend, P. They used both single threshold and multiple threshold criteria:

Single threshold criterion

$$\begin{array}{lll} \textit{buy} & \text{if} & P > 0.5 \\ \textit{hold} & \text{if} & P = 0.5 \\ \textit{sell} & \text{if} & P < 0.5; \end{array}$$

Multiple threshold criterion

$$\begin{array}{lll} \textit{buy} & \text{if} & P > 0.7 \\ \textit{hold} & \text{if} & 0.5 \leq P \leq 0.7 \\ \textit{sell} & \text{if} & P < 0.5 \end{array}$$

Leung et al. [44] also followed a similar multiple threshold criterion. Results obtained by Chen et al. [11], suggested that the multiple threshold criterion is more profitable than the single threshold criterion.

The multiple threshold criteria adopted in [11, 44, 40, 56] *seems to be more practicable, the only shortcoming is that the probability levels in* [11, 44] *or boundaries in* [40, 56] *may vary from one stock market index to another.*

2.3 Algorithms Used for Stock Market Prediction

Past studies (mentioned in Section 2.2.1, and 2.2.2) evidence that FNN, PNN and SVM are the most successful algorithms for predicting the direction as well as trading signals of the stock market indices. Therefore, in this section, we will review the literature regarding to the training of these three algorithms and also discuss their shortcomings. Furthermore, the attempts to overcome the problems associated with theses algorithms by the past studies, are revealed.

2.3.1 Applications of Feedforward Neural Networks for Predicting Trading Signals

The literature [16, 21, 22, 44, 65, 68, 95, 100] shows that FNN is the most commonly used model to predict the value (price level) of stock market indices. Also it was proved that FNN outperforms linear models such as the regression, ARIMA and random walk models. Some studies [21, 44, 65, 68, 95, 100] went beyond the value prediction by classifying the predicted value into two categories, upward and downward trend, and then assessed the FNN's ability to predict the direction of stock market indices. FNN is seems to be a promising alternative algorithm to classification algorithm such as PNN and SVM.

Fernando et al. [21] applied a three-layered neural network model to predict the relative return of the General Index of the Madrid Stock Market. This model consists of one hidden layer with four neurons. A logarithmic function was used as the transfer function between the input and hidden layers while hyperbolic tangent function was used between the hidden and the output layers. The values assigned for the parameters of the model such as learning rate and momentum as well as sizes of training and test sets are not mentioned in the paper.

The FNN models applied by Leung et al. [44] also consist of three layers with one hidden layer. Different numbers of neurons were allocated for the hidden layer depending on the stock market index. They used 'ThinkPro' computer software to develop these FNNs. The whole data set consists of 348 samples. 17% of the most recent data was used for testing while the remaining data was allocated for training. No further information is mentioned in their paper.

Pan et al. [65] also applied a three-layered neural network model with one hidden layer to predict the direction of the AORD. The hidden layer consists of three neurons. A sigmoid function was used as the transfer functions between the input and the hidden layers, and a linear transfer function was used between the hidden and the output layers. 20% of data randomly chosen to be used for testing while the remaining data was used for training. The best predictions were obtained when the learning rate and momentum were 0.03 and 0.4, respectively.

Vanstone [88] employed a software package, 'NeuroLab' to develop FNN models. This software uses logistical sigmoid functions as the transfer functions. The number of hidden neurons as well as the learning rate and the momentum were varied according to the different models tested.

A three-layered FNN model with one hidden layer employed by Wood and Dasgupta [95] was trained with a learning rate equal to 0.0001 and momentum equal to 0.1. Sigmoid functions were used as the transfer functions. The data set used in this study consists of 142 data points. The most recent 16% of data was used for testing while the rest was allocated for training.

Unlike the studies discussed above, the study done by Yao et al. [100] employed three-layered as well as four-layered FNNs. They also varied the number of neurons in the hidden layers. Sigmoid hyperbolic tangent functions were used as the transfer functions. The data used in this study consists of daily time series data from January, 1984 to October, 1991. The most recent data was used for testing and the remaining was used for training.

Although above studies claim that FNN produced more accurate predictions, there are some shortcomings associated with FNN. The literature [15, 32] reveals the possibility of the FNN finding suboptimal solutions as a result of being trapped in local minima. Several studies (for example [32, 54, 87, 103]) attempted to find global solutions for the parameters of the FNNs by developing new algorithms. Minghu et al. [54] proposed a hybrid algorithm of global optimization of dynamic learning rate for FNNs and this algorithm is shown to have a global convergence for error backpropagation multilayer FNNs. Ye and Lin [103] proposed a new approach to supervised training of weights in MLFNNs. Their algorithm is based on a 'subenergy tunnelling function' to reject searching in unpromising regions and a 'ripple-like' global search to avoid local minima. Jordanov [32] proposed a global algorithm which makes use of a stochastic optimization technique based on so-called low discrepancy sequences to trained FNNs. Toh et al. [87] also proposed an iterative algorithm for global FNN learning.

There is another problem specific to the application of FNN as a classifier. The ordinary least squares (OLS) error function (see 3.3), which is used as the error function in standard FNNs, is inappropriate for the problem of classifying trading signals of a stock market index. This is because, when minimising the OLS error function, FNNs try to minimise the difference between the actual and the predicted value. On the other hand, in classification of trading signals the aim is to minimise the misclassification irrespective of the size of the error (the difference between the actual and the predicted value).

To address this issue, some past studies [7, 69, 101, 102] proposed modification to the OLS error function. These studies incorporated factors which represent the direction of the prediction [7, 101, 102] and recency of the data that was used as inputs [69, 101, 102], when suggesting the modifications. *However, these studies considered only two trading signals: buy and sell, which correspond to up and down movements, and therefore, penalised the wrongly classified direction (positive/negative). Hence the modified error function proposed by these studies may not be suitable for the case of classifying three trading signals: buy, hold and sell.*

2.3.2 Applications of Probabilistic Neural Networks for Predicting Trading Signals

When training PNNs to obtain the probability that the predicted relative return showing an upward trend, Chen et al. [11] assumed that the joint distribution of the input variables is gaussian. Estimation of the parameters of the distribution (the mean and the standard deviation) were based on the training data. This study used moving windows to train the networks. The first 68 samples were used as the training set to predict the value corresponding to the 69th sample; then the second 68 samples (from second to 69th samples) were used to predict 70th sample. This procedure was repeated 60 times to obtain the probabilities corresponding to the last 60 observations of the data set.

The paper authored by Leung et al. [44] does not mention any detailed description of the network training. 288 observations (83% of the whole data set) were used for training PNNs while the remaining 60 observations were used as the test sample.

Some studies (for example [44] proved that the PNN outperformed the FNN in terms of profitability of predictability. However, PNN algorithms also show some shortcomings. For example, PNN algorithms, such as the Matlab PNN algorithm [14] does not allow the consideration of a distribution other than the Gaussian, as the distribution of input variables [77]. There is evidence from the literature [3, 24, 50] that the distribution of the stock price index returns deviate from the Gaussian distribution.

To deal with the problem of imbalanced data, PNN has a solution: that is, varying the loss due to misclassification [11, 74, 77] (Section 3.2.2) according to the size of the class. *However, it is not straightforward to assign a proper value for loss due to misclassification for different trading signals, as the loss depends on seriousness of the misclassification. For instance, the misclassification of a buy signal as a sell is very serious mistake while if the same signal is misclassified as a hold signal, then the mistake is less serious.*

2.3.3 Applications of Support Vector Machines for Predicting Trading Signals

Huang et al. [26] used a support vector machine (SVM) with a radial basis function as the kernel to forecast the direction (upward and downward trend) of the Japanese NIKKEI 225 Index. The parameter of this radial basis kernel was set to $\sqrt{10}$. The training set used included 640 samples (approximately 95% of data) while 36 samples were included in the test set.

Kim [34] also used a radial basis function as the kernel of the SVMs trained to predict the direction of the Korea Composite Stock Price Index. He tested different values for the parameter of the radial basis function and best prediction results were obtained when the parameter was equal to 5. The training set included 2347 samples (80% of data) while test set consisted of 581 samples (20% of data).

Furthermore, Kim [34] showed that the SVM outperformed the FNN in terms of predictability. However, it is widely accepted by researchers that traditional classification algorithms such as SVM, decision trees, neural networks, etc. do not perform well when the data has an imbalanced distribution among the classes of interest [1]. Both studies [26, 34] applied SVM for prediction of two classes, buy and sell signals which correspond to up and down trends. Therefore, the data used in these studies has balanced distribution. This may be the reason that SVM produced accurate prediction results in these studies.

Several attempts have been made in the literature (for example [1, 10, 27, 57, 90, 97]) to modify the SVM algorithm to address the issue of imbalanced data. Chawla et al. [10] tried over-sampling the minority class by creating a synthetic minority class. Veropoulos et al. [90] suggested penalising classes making errors on positive instances at a higher rate than errors on negative instances. Combining the algorithms proposed by Chawla et al. [10] and Veropoulos et al. [90], Akbani et al. [1] introduced a new SVM algorithm to deal with the problem of imbalanced data. Their experiments showed that this new algorithm outperforms the other two algorithms. Morik et al. [57] also introduced a correction factor to deal with the problem of imbalanced data and this is incorporated in 'SVM-Light' (version 6.01) software [29]. Modifying the kernel matrix according to the imbalanced data distribution, Wu et al. [97] proposed a kernel-boundary-alignment algorithm.

2.4 Integrated Markets and Intermarket Influences

Bhattacharyya and Banerjee [5] argued that capital markets are not only influenced by the domestic macro economic factors. The electronic communication and media have increased the availability and timeliness of information across the globe. The movements in the assets prices in a particular country's capital markets are continuously affected by the inflow relevant to 'global' information [5]. Many other past studies (for instance [5, 18, 80, 96, 99]) which are focused on examining the relationships between world's major stock markets suggest that they are interrelated (integrated).

Becker et al. [4] examined the inter-temporal relationship between the USA and

Japanese stock markets. Their research revealed the existence of a high correlation between the open to close returns for USA stocks for the previous trading day and the Japanese equity market performance in the current period. In contrast, the Japanese market has only a small impact on the USA returns in the current period.

Eun and Shim [18] investigated the international transmission mechanism of stock market movement from 1980 to 1985. They analysed daily return data from markets from nine countries; Australia, Japan, France, Germany, Switzerland, the UK, Canada, the USA and Hong Kong (China). Their research provided evidence that a substantial amount of multi-lateral interaction exits among these national stock markets. Furthermore, their research confirmed that the USA stock market was the most influential market among the nine markets considered while none of the eight foreign markets significantly influenced the USA markets.

Compared to the previous studies [18, 4], the study done by Wu and Su [96] has made several advances such as systematic examination of the existence of four possible relationships and the testing of the relationships between stock markets, after removing the effect of the other markets. They analysed the stock returns of the USA, Britain, Japan, and Hong Kong from 1982 to 1991. They used a multiple hypothesis testing procedure to systematically examine the existence of four possible types of relationships; independent, contemporaneous, unidirectional, and feedback, among the markets. Their approach also allowed the examination of the relationships between stock markets conditioned on the effect of the other markets.

They found the existence of significant dynamic relationships among the four markets considered. These relationships were strengthened considerably after the 1987 stock market crash. Correlations among markets have been much higher in more recent years. Their results revealed that there was an asymmetry in the cross correlations between stock market returns. Larger markets appeared to lead smaller markets. The USA stock market continued to exert a strong influence on other markets after 1987. However, other markets also had an impact on the USA market particularly in recent years. They also found that the Japanese market had a fairly strong influence on other markets after the influence of the USA was removed. Their research provided evidence for a structural shift in the international market dynamics. Therefore, they suggested that lead-lag relationships might change over time.

A study done by Taylor and Tonks [80] suggested that since the abolition of the UK exchange control, the UK stock market has become cointegrated with other stock markets, namely the stock markets of Germany, the Netherlands and Japan. Furthermore, their results suggested that in the long run, the returns of these markets are highly correlated.

The research done by Yang et al. [99] suggested that there is not enough evidence of integration among the larger markets (particularly, the US, Japanese, the UK and German stock markets) in the long run. However, they found sufficient evidence for the increasing the integration between the US and the smaller markets of the world.

Furthermore, Mendelsohn [53] suggested that intermarket analysis can be used in conjunction with traditional single-market technical indicators to broaden trading perspective.

The literature [5, 18, 80, 96, 99] *confirms that the world's major stock markets are integrated. Also some studies* [4, 18, 96] *provide evidence that US stock markets have a strong influence on the other major global markets. These studies confirm the existence of intermarket influences (Section 1.1.2) among the global stock markets. Hence, each stock market, which belongs to this set of integrated markets, can be considered as a part of a single global system* [84]*. The influence from one integrated stock market on a dependent market may include the influence from one or more stock markets on the former. This matter indicates that the intermarket influences (from a set on influential markets on a dependent market) needs to be quantified in order to use them effectively in applications such as predictions.*

Surprisingly, the literature does not provide any evidence for an existing method for quantifying intermarket influences. This highlights the necessity of introducing such techniques.

2.5 Input Features Used for Predictions

As noted in Section 1.1.1, regarding stock market predictions, there are two types of analysis: fundamental analysis and technical analysis. Technical analysis looks in depth at financial conditions and operating results of a specific company and underlying behaviour of its common stock; the value of a stock is established by analysing the fundamental information associated with the company, such as accounting, competition, and management. For fundamental analysis, retail sales, gold prices, industrial production indices, foreign currency exchange rates, etc. can be used as the input features [100].

On the other hand, technical analysis is based on the assumption that stock markets move in trends and these trends can be captured and used for forecasting. It attempts to use past stock prices and volume information to predict the future price movements [100].

Intermarket technical analysis was coined by Murphy [58] and is a relatively recent area of technical analysis practised by professional analysts. Based on qualitative analysis, he suggested that all major traded asset markets (commodities, bonds, stocks and currencies) are interlinked in an international framework.

Nowadays, experts argue that stock markets are influenced by many interrelated factors including the affects of economic, political and even psychological factors. These factors interact with each other in a complex fashion, and it is therefore, very difficult to find an exact set of factors which determine the behaviour of stock markets [82].

2.5.1 Input Features Used for Directional Predictions

Some published research (for example [11, 26, 35, 44, 45]) used input variables which consist of combinations of fundamental and technical indicators, to predict the direction (up/down or positive/negative) of different stock market indices. Although, not very common, the input set used by Qi and Maddala [68] consists of only the fundamental variables. The directional prediction based on the technical indicators seems to be a very common feature in the literature (for instance [34, 74, 104, 21, 23, 65, 95]).

Among the technical indicators employed for direction prediction, the most commonly used inputs are the lagged price indices of the stock market index to be predicted or their derivatives, such as relative or log returns, moving averages (for example [11, 21, 35, 44, 65, 104]). *However, the application of intermarket influences, that is the information (such as lagged price or relative return) of other stock market indices, is rare.* Studies done by Huang et al. [26], Pan et al. [65], and Poddig and Rehkugler [66] are among the few such studies. Huang et al. [26] and Pan et al. [65] applied the lagged data of the US S&P 500 Index to predict the direction of the Japanese NIKKEI 225 Index and the AORD, respectively. Poddig and Rehkugler [66] used a system of interdependent equations to predict the direction of the stock markets, bond markets and currency rates of the US, Japan and Germany. To predict the direction of each capital market, the lagged data of the other capital markets considered were taken as the input features.

It is noteworthy that these studies [63, 65, 66] *suggested that incorporating intermarket influences for predictions improves the prediction accuracy. However, these studies did not take the quantified intermarket influences into account for the direction prediction.*

2.5.2 Input Features Used for Predicting Trading Signals

Vanstone [88] claimed that fundamental variables may be suitable as input features, if the intention is to do long term forecasts. On the other hand, if the intention is to do short term predictions, technical variables may be more suitable.

Some studies (for example [11, 38, 40, 44, 88] relied on both types of variables for forecasting. Although not very common, Kohara et al. [38] and Kuo [40] included qualitative variables, such as the effect of news and political effects, in the input sets that they used for predicting trading signals.

As in the case of directional prediction, many studies (for instance [11, 13, 21, 38, 40, 44, 56]) used technical indicators to predict trading signals. Some of these studies [13, 21, 56] relied only on technical indicators. The application of the lagged price or a derivative of the price of the stock market whose trading signals to be predicted seems are to be a common feature in the fast studies [11, 13, 21, 38, 44, 56].

It is noteworthy that the application of lagged prices or the derivatives of the prices of foreign stock markets, to predict the trading signals of a selected market, is very rare. The use of such information to predict trading signals may improve the predictability and profitability of the prediction.

2.6 Summary

Published research suggests that intermarket influences improves the accuracy of predictions related to stock markets. However, using intermarket influences for predicting the direction or trading signals of a selected stock market is not very common in the literature. The need to quantify intermarket influences before applying them for predictions, was understood. Surprisingly, literature does not provide any evidence for the existence of techniques which can be applied to quantify intermarket influences (from a selected set of influential markets on a given market).

The prediction of three trading signals including the hold signal is also not very common among the past studies. The literature provides evidence for the drawbacks of the most commonly used algorithms (FNN, PNN and SVM) for the classification of trading signals. This matter suggests the need to develop new algorithms by addressing these issues.

The next chapter focuses on the methodology used on this study. Also it explains the three algorithms, FNN, PNN and SVM in detail.

Chapter 3

Methodology and the Techniques Applied

This chapter describes the methodology that was used to achieve the objectives of the study. The classification techniques (algorithms) used in the experiments are also discussed.

3.1 Methodology

The research methodology included the following six steps:

1. Designing a technique to quantify intermarket influences by minimising the rank correlation between the relative return of the Close price of day t of a given dependent market and the lagged relative returns of the Close prices of a set of potential influential markets.

2. Quantifying intermarket influences from different combinations of potential influential markets on the AORD by using the technique proposed in step 1;

3. Predicting the direction of the Close price of the AORD using FNNs, in particular using the quantified intermarket influences for the directional prediction. Inves-

tigating whether the quantified intermarket influence can effectively be used for directional prediction;

4. Predicting whether it is best to buy, hold or sell (trading signals) using the FNNs, PNNs and the SVMs, in particular using the quantified intermarket influences for predicting trading signals. Investigating whether quantified intermarket influences can effectively be used for such predictions;

5. Investigating the effectiveness of the methods and algorithms used for predicting trading signals and identifying their shortcomings; Verification of the predictions obtained from these algorithms;

6. Developing new algorithms, which predict the trading signals of the AORD, by modifying the error function associated with the FNN and employing a global optimization algorithm to train the networks; Investigating the effectiveness of using the quantified intermarket influences for such predictions.

Figure 3.1 illustrates the steps of the overall methodology and links to the chapter in which each step is addressed. Sections 3.1.1 to 3.1.5 provide brief descriptions about all steps involved in the methodology.

3.1.1 Designing a Technique for Quantifying Intermarket Influences

Literature provides evidence that the most of the world's major stock markets are integrated (Section 2.4). Therefore, there is a possibility that the influence from an integrated stock market on a dependent stock market will include the influence from one or more stock markets on the former. A direct measure of influence from one market on another is inappropriate. Hence, this study introduces a technique which generates a relative measure. This measure estimates the influence from an influential market to a dependent market, relative to the influence from other influential markets on this dependent market.

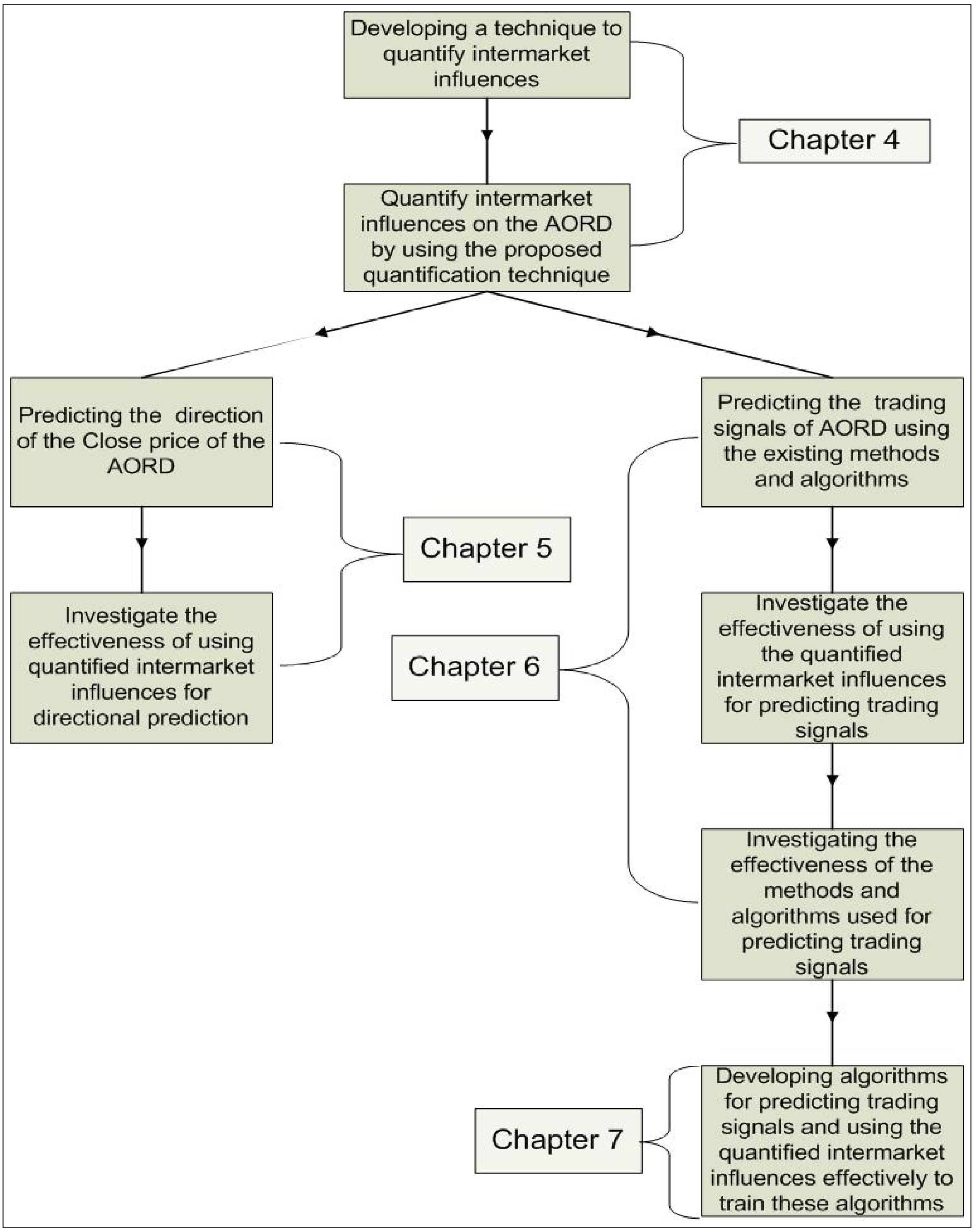

Figure 3.1: Overall methodology

This was done by estimating the combined influence of a set of influential markets and finding the contribution from each influential market to this combined influence. The full description of the proposed technique is given in Chapter 4 (Section 4.2).

3.1.2 Quantifying Intermarket Influences on the AORD

Intermarket influences from different market combinations on the AORD was quantified by applying the quantification technique proposed in Chapter 4 (Section 4.2). These market combinations included major stock market indices of the US, European, Asian markets as well as the AORD index itself (Section 4.3). Quantification was carried out by considering the whole study period as a single window as well as for different moving windows. Influences from the Close price of potential influential markets within a week were studied. Since, the correlation structure may change over the time [83, 96], different moving windows were considered in order to capture the dynamic patterns of the intermarket influences. A detailed description of quantification of intermarket influences on the AORD together with the quantification results are given in Chapter 4.

3.1.3 Directional Prediction Using Quantified Intermarket Influences

To a lesser extent, feedforward neural networks (FNNs) are used as a technique for predicting the direction of a stock market index [64, 104] (Section 2.2.1). On the other hand, instead of predicting the direction of the stock market of interest, some studies [21, 23, 65, 95, 100] (Section 2.2.1) predicted the price level using FNNs, and then prediction accuracy was evaluated by comparing the sign (positive or negative) of the predicted value with that of the actual value.

If the relative return of day $(t+1)$ is zero or approximately zero, then there is no substantial difference between prices corresponding to day t and day $(t+1)$, irrespective of the sign. Sign correctness measure does not take this matter into account. To fix this problem, Pan et al. [65] and Tilakaratne [82] introduced a threshold (Section 2.2.2).

Even though the signs of the actual and the predicted values are not the same, if the absolute value of the difference between the actual and the predicted values is less than the threshold, then it was assumed that the signs of the both values are the same.

We also applied FNN (see Section 3.2.1 for more detail about FNN) to predict the relative return of the Close price of day $(t + 1)$ of the AORD. A similar approach to Pan et al. [65] and Tilakaratne [82], was adopted for evaluating the predictions. We introduced a more appropriate measure, Direction Correctness Percentage (DCP) to asses the performance of the networks. DCP indicates the percentage of predictions with the correct direction (up, down or no change). More details about the DCP is given in Section 5.2.2.

Three types of inputs were considered when training the FNNs:

1. Lagged relative returns of the potential influential markets as separate inputs;

2. Sum of the quantified lagged relative returns (that is, lagged relative returns multiplied by the corresponding quantification coefficients, ξ_i, i=1, 2, ...,) of these markets as a single input;

3. Quantified lagged relative returns of these markets as separate inputs.

The last two types of input were employed in order to examine how the quantified intermarket influences can be incorporated for the directional prediction.

Chapter 5 discusses the prediction of the direction of the Close price of the AORD using quantified relative return in detailed. Description of neural network experiments together with the results obtained are also given in this chapter.

3.1.4 Predicting Trading Signals Using Quantified Intermarket Influences

Some past studies considered the direction, upward or downward trends, of the Close price of a stock market as buy or sell signals, respectively (for example [21, 101, 102]).

In practice, a trader does not buy or sell if there is no significant change in the price level; instead, he/she holds the money or shares in hand. Therefore, we noted that the consideration of hold signals is as important as buy and sell signals. We used our own criterion (Criterion A) to identify the trading signals. This criterion is described in Section 6.2.

This study investigated whether quantified intermarket influences can effectively be used to predict the trading signals; buy, hold and sell, of the AORD. Therefore, the quantified relative returns of the influential markets as well as their un-quantified counterparts were used as the input variables. These input variables belong to the above mentioned (Section 3.1.3) three types of inputs.

As argued in the literature (Section 2.3) FNN, PNN and SVM are better algorithms for predicting the trading signals of a given stock market. The detailed description of the mechanism behind FNN, PNN and SVM are given in Section 3.2.1, 3.2.2 and 3.2.3. This study also adopted these three algorithms to predict the trading signals of the AORD. The three types of inputs mention in Section 3.1.3 above were used to train these algorithms.

The prediction results were evaluated in terms of predictability as well as the profitability. When evaluating the predictability, it is important to consider not only the classification rate but also the misclassification rate. This is because loss incurred due to serious misclassification (such as misclassification of sell signal as buy signal and vice versa) may overrun the gain obtained by responding the correctly classified signals. Therefore, we employed the classification and misclassification rates (refer Section 6.4 for more details) as the measures of predictability while the rate of return obtained by performing trading simulations (Section 6.4) was used as the measure of profitability.

A new trading simulation was proposed (Section 6.4.1). The speciality of this simulation is that it searches the proportion of money and shares involved in trading in order to gain higher the profits. Chapter 6 explains the related experiments and their results in detail.

3.1.5 Developing Algorithms for Predicting Trading Signals

This study is interested in predicting three trading signals, buy, hold and sell. Consideration of these three classes (signals) resulted in an imbalance in data distribution (section 6.6.2). This imbalance caused the classification algorithms, PNN and SVM, to be less effective (see results in Section 6.6.2 and 6.6.3). FNN provided results which could be acceptable (see results in Section 6.6.1. However, FNN uses backpropagation learning for weight modification (Section 3.2.1) and backpropagation learning is heavily dependent on the initial weight randomisation and can often converge to the solutions which are less than optimal [9]. An analysis of the distribution of the error function resulting from the FNN training was carried out and is described in Section 6.7. The results of this analysis suggested that the solutions could be far from the global optimal solutions.

New algorithms for predicting trading signals, were developed based on neural network techniques. When developing these new algorithms, the main concern was to modify the ordinary least squares (OLS) error function (see (3.3) in Section 3.2.1), in a way that suits the problem of interest: classification of trading signals into three classes, buy, hold and sell. Following similar past studies [7, 69, 101, 102]), we introduced two modified error functions. A detailed description about the alternative error function is given in Section 7.2.1.

This study proposed four neural network algorithms and these algorithms are explained in Section 7.3. We used a global optimization algorithm, AGOP (Section 4.2.2) to train these networks. By using a global optimization algorithm for network training, we aim to find 'deep' solutions to respective error minimisation functions.

The same types of inputs that are mentioned in Section 3.1.3 above, were used as inputs to these new algorithms. The performance of these algorithms were evaluated by overall classification rate as well as overall misclassification rate (Section 7.3.2). The prediction results of the best of each type of neural network algorithm were further evaluated in terms of profitability (by performing trading simulations; Section 7.5).

A full description about the development of neural network algorithms for predicting trading signals together with the results obtained are given in Chapter 7.

3.2 Algorithms for Predicting Trading Signals

This section explains the three algorithms that this study adopted for predicting trading signals. The literature (Section 2.2.1 and 2.2.2) gives evidence that feedforward neural networks (FNN), probabilistic neural networks (PNN) and the support vector machines (SVM) are the most commonly used and appropriate algorithms.

3.2.1 Feedforward Neural Networks

Figure 3.2 depicts an example of a multilayer feedforward neural network. A multilayer feedforward neural network can have any number of layers and any number of units (neurons) per layer. The first layer is called the input layer and the last layer is called the output layer. The middle layers are called hidden layers. The network shown below has four neurons (or units) in the input layer, three neurons in the hidden layer, and one neuron in the output layer.

Each neuron-to-neuron connection is modified by a weight (or connection strength). In addition, each neuron has an extra input that is assumed to have a constant value of one, and the weight that modifies this extra input is called the bias. All the information propagates along the connections in the direction of network inputs to network outputs, hence the term feedforward.

The input neurons simply pass on the input vector $a' = \{I_j : j = 1, 2, \ldots, n\}$. The following equation gives the net input to the rth neuron of the hidden layer:

$$net_r = f_1(\Sigma_j w_{jr} I_j + b_r) \tag{3.1}$$

where I_j is the jth input variable to the input layer, w_{jr} is the weight of the link connecting jth neuron of the input layer to rth neuron of the hidden layer, and b_r is the bias associated with the rth neuron of the hidden layer. f_1 is the transformation function between the input layer and the hidden layer.

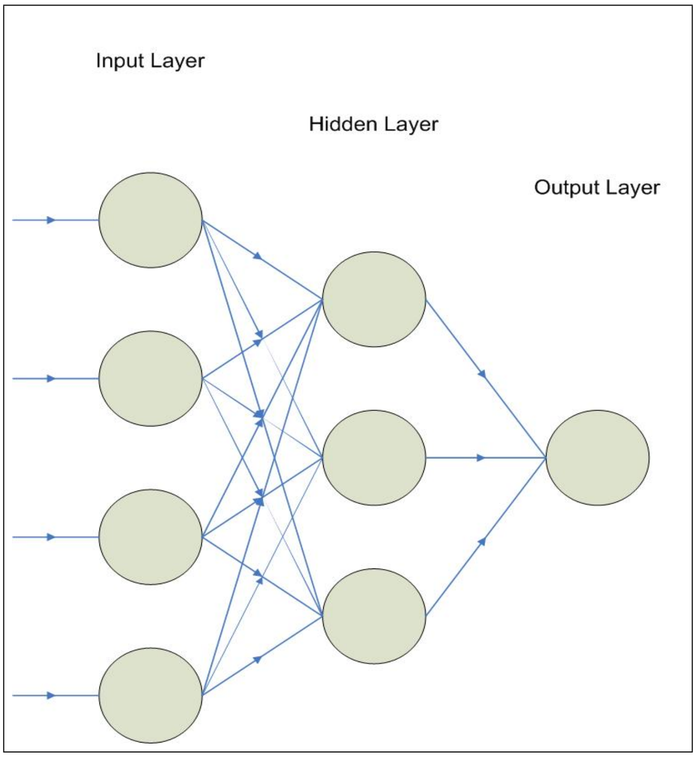

Figure 3.2: A three-layered feedforward neural network

Input to the sth neuron of the output layer is given by:

$$net'_s = f_2(\Sigma_r w_{rs} net_r + b'_s) \tag{3.2}$$

where f_2 is the transformation function between the hidden layer and the output layer. w_{rs} is the weight of the link connecting rth neuron of the hidden layer to sth neuron of the output layer, and b_r is the bias associated with the sth neuron of the output layer. net_r is defined by (3.1).

A network is fed with inputs as well as outputs (training data or sample data). Then the network learns the mapping from inputs to corresponding outputs. This is called *supervised learning*. In order for the network to learn the patterns of the data, a learning algorithm is needed. Backpropagation is the learning algorithm most commonly used for the feedforward neural networks.

The Backpropagation Algorithm

Backpropagation is an algorithm that modifies network weights to minimise the mean squared errors between the predicted and actual outputs of the network. Backpropagation is a supervised learning algorithm. Once the network is trained, the weights are optimized and these optimized weights can then be used to compute outputs for new inputs.

Let the training set be denoted by $\{(\widetilde{a}^i, a^i) | i = 1, 2, \ldots, N\}$. Once the input vector, $\{\widetilde{a}^i : i = 1, 2, \ldots, N\}$ is fed into the network, it computes an output vector $\{o^i : i = 1, 2, \ldots, N\}$. Then $o^i, i = 1, 2, \ldots, N$ is compared against the training target $a^i, i = 1, 2, \ldots, N$. A performance criterion function is defined based on the difference between a^i and o^i. The commonly used criterion function is the ordinary least squared (OLS) error function, which is given by (3.3).

$$E_{OLS} = \frac{1}{N} \sum_{i=1}^{N} (a^i - o^i)^2 \tag{3.3}$$

Backpropagation tries to minimise the sum of squared errors, by forcing the network weights to change in such a way that errors are minimised. Backpropagation training consists of three steps:

1. Output Calculations: present the given input vector to the network inputs and run the network: compute the activation functions sequentially forward from the first hidden layer to the output layer.

2. Error Backpropagation: compute the difference between the predicted output and the actual output. Propagate the error sequentially backward from the output layer to the input layer.

3. Weight Modification: for every connection, change the weight by modifying that connection in proportion to the error.

When these three steps have been performed for every example from the data series, one epoch of training has occurred. Learning usually runs through thousands of epochs, either until a predetermined maximum number of epochs is reached, or the network output error falls below an acceptable threshold. Training (learning) of a network can be time consuming, depending on the network size, size of the training data set, number of epochs, and the desired network output errors.

During the first step mentioned above, an input vector is presented to the input layer, and then the network computes the output for the non-input units. For instance, the network output for the ith example is:

$$o_s^i = f_2(\Sigma_r w_{rs}(f_1(\Sigma_j w_{jr} I_j^i + b_r)) + b'_s) \tag{3.4}$$

where k, s, w_{rs}, w_{jr}, b_r, b'_s, f_1, and f_2 are defined in Section 3.2.1. I_j^i represents the jth input variable (input to the jth neuron of the input layer) of the ith training sample.

During the second step, the error terms for each output neuron are computed (given by 3.5), as well as for each neuron of each hidden layer (given by 3.6).

$$\delta_s^i = (o_s^i - a_s^i)\phi'(net'_s) \tag{3.5}$$

where s is the index of output neuron and ϕ' is the derivative of transformation function between the hidden and the output layers.

$$\delta_r^i = \phi(net_r)\Sigma_s \delta_s^i w_{rs} \tag{3.6}$$

where r is the index of hidden neuron and ϕ is the derivative of the transformation function between the input and hidden layers.

During the third step, the error computed from the output layer is backpropagated through the network, and weights are modified according to their contribution to the error function defined by (3.3) above. The change in the weight is computed according to (3.7) and added to the original weight.

$$\Delta w_{jr}^i = \eta \delta_r^i o_r^i \tag{3.7}$$

where w_{jr} is the weight of the link connecting jth neuron to rth neuron, and o_r^i is the output of rth neuron corresponding to the ith sample. η is called the *learning rate*.

Choice of Learning Rate (η)

The learning rate controls how quickly and how finely a network converges to a particular solution. At the start of a training sample using the backpropagation algorithm, the weights change proportional to the negative gradient of the error; but the magnitude of the desired weight change is not fixed. The magnitude depends on the appropriate choice of the learning rate, η. A large value of η will lead to rapid learning but the weight may then oscillate, while low values imply slow learning. The proper value for η depends on the application. Usually this value changes from 0 to 1 [52].

Momentum

Backpropagation may lead the weights in a neural network to a local minimum of the error function (see (3.3)). This local minimum may be substantially different from the global minimum that corresponds to the best choice of weights. Therefore, it is essential to take some corrective action to prevent the network from getting stuck in a local minimum. This problem can be overcome by making weight changes in an iteration of the backpropagation algorithm dependent on the immediately preceding weight change. Then (3.7) can be modified as below:

$$\Delta w_{jr}(t+1) = \eta \delta_r o_r + \alpha \Delta w_{jr}(t) \tag{3.8}$$

where $w_{jr}(t)$ is called *momentum*, which is the weight change required at time t and α is called the *momentum coefficient*. A correct choice of α will significantly reduce the number of iterations to convergence. A value close to 0 indicates that the past history does not have much effect on the weight change. However, a value close to 1 suggests that current error has little effect on the weight change [52].

3.2.2 Probabilistic Neural Networks

A Probabilistic neural network (PNN), introduced by Specht [77], is a nonlinear, non-parametric classification algorithm that has been described as the neural network implementation of kernel discriminant analysis [72].

PNN Logic

PNN, which is based on the Bayesian method of classification, is capable of classifying a sample with the maximum probability of success, provided that there is enough data to estimate the distribution ([92] cited in [11]). The principle of a Bayesian classifier rests on the selection of class C with the largest product term in the Bayesian Classification Theorem:

$$\mathrm{Max}_C\{P_C L_C f_C(X)\}, \tag{3.9}$$

where P_C is a priori probability for class C, L_C is the loss incurred by misclassifying a sample which actually belongs to class C, $X = (x_1, x_2, \ldots, x_j)$, is the input vector (of j number of elements) to be classified, and $f_C(X)$ is the probability of X given the density function of class C [11].

There is no particular method or technique to decide the value of L_C; prior knowledge or a trial and error method is used to estimate this value. Many studies related to financial predictions assume that the loss of misclassification is equal for each class.

In general, the distribution of the vector X assumed to be Gaussian:

$$f_C(X) = \frac{1}{(2\pi)^{j/2}\sigma_C^j n_C} \sum_{j=1}^{n_C} \exp \frac{-(X - Y_{iC})'(X - Y_{iC})}{2\sigma_C^2} \tag{3.10}$$

where j is the number of elements in X, n_C is the number of training samples belong to class C, Y_{iC} is the ith training sample in class C, and σ_C, which is equal to the standard deviation of samples belong to class C, is called a smoothing parameter. However, the distribution of the vector X may take other possible forms of distributions [77].

A basic PNN topology consists of four layers: an input, an output and two hidden layers) of processing units (for example Figure 3.3) [11]. The input layer has a processing unit to represent each independent variable in the input vector X, while the output layer consists of a set of processing units to indicate the output class. The first hidden layer is called the *pattern layer* and uses a processing unit to 'memorise' each training sample. The second hidden layer is termed the *class layer* and is made up of an array of units with the number equal to the total number of classes.

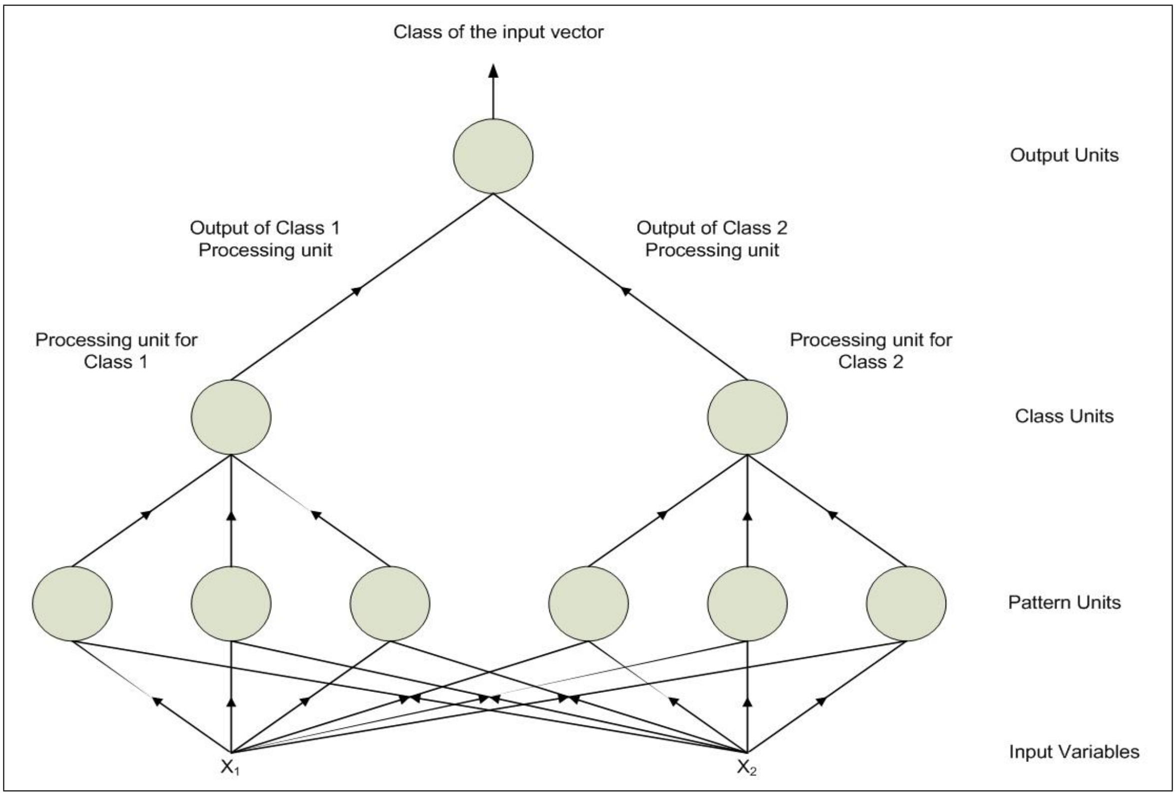

Figure 3.3: An example for probabilistic neural network Source: Chen *et al.* [11]

Figure 3.3 depicts a simple PNN which represents a model with two input variables (X_1 and X_2), one output with two classes, and three training samples for each of the two classes.

3.2.3 Support Vector Machines

A support vector machine (SVM), a novel network algorithm, was developed by Vapnik [89]. Unlike the traditional neural network models which minimise the deviation from the correct solution, SVM minimise an upper bound of generalisation error [34]. Hence, solutions produced by SVM may be global optimal solutions.

A SVM maps a set of input vectors $\mathbf{x}$ into high-dimensional feature space, through some non-linear mapping, chosen a priori [89]. A linear model constructed in the new space can represent a nonlinear decision boundary in the original space [34]. In the new space, an optimal separating hyperplane is constructed and this hyperplane is termed the *maximum margin hyperplane.* The maximum margin hyperplane gives the maximum separation between decision classes. The training samples that lie on this plane are called *support vectors* while all other training samples are considered to be irrelevant for defining the binary class boundaries.

Formalisation of the SVM problem

Consider data points of the form:

$$\{(\mathbf{x}_1, c_1), (\mathbf{x}_2, c_2), \ldots, (\mathbf{x}_n, c_n)\} \tag{3.11}$$

where the c_i is either 1 or -1 which denotes the class to which the point $\mathbf{x}_i$ belongs. Each $\mathbf{x}_i$ is a p-dimensional real vector. (3.11) can be considered as the training data, which denotes the correct classification which we would like the SVM to eventually distinguish, by means of the dividing (or separating) hyperplane [105]:

$$\mathbf{w} \cdot \mathbf{x} - b = 0. \tag{3.12}$$

The vector $\mathbf{w}$ is perpendicular to the separating hyperplane. The offset parameter b allows to increase the margin. In its absence, the hyperplane is forced to pass through the origin, restricting the solution [105].

Since the aim is to maximise the margin, it is necessary to find the support vectors and the parallel hyperplanes (to the optimal hyperplane) closest to these support vectors in either class. These parallel hyperplanes can be described by equations (by scaling w and b) [105]:

$$\begin{aligned} \mathbf{w} \cdot \mathbf{x} - b &= 1 \\ \mathbf{w} \cdot \mathbf{x} - b &= -1. \end{aligned} \tag{3.13}$$

If the training data are linearly separable, these hyperplanes can be selected in a way that there are no points between them and then try to maximise their distance. The perpendicular distance between the hyperplanes is $2/\parallel w \parallel$, so we want to minimize $\parallel w \parallel$. To exclude data points, we need to ensure that for all i either [105]:

$$\begin{aligned} \mathbf{w} \cdot \mathbf{x}_i - b &\geq 1 \quad \text{or} \\ \mathbf{w} \cdot \mathbf{x}_i - b &\leq -1. \end{aligned} \tag{3.14}$$

This can be rewritten as [105]:

$$c_i(\mathbf{w} \cdot \mathbf{x}_i - b) > 1, \quad 1 < i < n. \tag{3.15}$$

Primal Problem

The problem now is to minimize $\parallel w \parallel$ subject to the constraint 3.15. This is a quadratic programming (QP) optimization problem. More clearly [105],

$$\text{Minimize} \quad (1/2) \parallel \mathbf{w} \parallel^2, \tag{3.16}$$

$$\text{s.t.} \quad c_i(\mathbf{w} \cdot \mathbf{x}_i - b) > 1, \quad 1 < i < n. \tag{3.17}$$

The factor of (1/2) is used for mathematical convenience [105].

Dual Problem

Writing the classification rule in its dual form reveals that classification is only a function of the support vectors, i.e., the training data that lie on the margin. The dual of the SVM can be shown to be [105]:

$$\text{Maximise} \quad \sum_{i=1}^{n} \alpha_i - \sum_{i,j} \alpha_i \alpha_j c_i c_j \mathbf{x}_i^T \mathbf{x}_j \tag{3.18}$$

$$\text{s.t.} \quad \alpha_i \geq 0, \tag{3.19}$$

where the α_i; $1 < i < n$ constitute a dual representation for the weight vector in terms of the training set [105]:

$$\mathbf{w} = \sum_i \alpha_i c_i \mathbf{x}_i \tag{3.20}$$

Nonseparable Case

If there exists no hyperplane that can split the training sample, the 'Soft Margin' method will choose a hyperplane that splits the examples as cleanly as possible, while still maximizing the distance to the nearest cleanly split examples. This method introduces slack variables, ϵ_i, which measure the degree of misclassification corresponds to x_i [105]:

$$c_i(\mathbf{w} \cdot \mathbf{x}_i - b) > 1 - \epsilon_i, \quad 1 < i < n. \tag{3.21}$$

The objective function is then increased by a function which penalises non-zero ϵ_i, and the optimization becomes a trade off between a large margin, and a small error penalty. If the penalty function is linear, (3.16), (3.17) transform to [105]:

$$\text{Minimize} \quad \parallel \mathbf{w} \parallel^2 + C \sum_i \epsilon_i, \tag{3.22}$$

$$\text{s.t.} \quad c_i(\mathbf{w} \cdot \mathbf{x}_i - b) > 1 - \epsilon_i, \quad 1 < i < n. \tag{3.23}$$

Problem (3.22), (3.23) can be solved using Lagrange multipliers. The key advantage of a linear penalty function is that the slack variables vanish from the dual problem, with the constant C appearing only as an additional constraint on the Lagrange multipliers. Non-linear penalty functions have been used, particularly to reduce the effect of outliers on the classifier, but unless care is taken, the problem becomes non-convex, and thus it is considerably more difficult to find a global solution [105].

For non-liner classification cases, the dot product in (3.17) and (3.23) is replaced by a kernel function, $K(x, y)$. The most commonly used kernel functions are the polynomial kernel, $K(x, y) = (xy+1)^d$ and the Gaussian radial basis function, $K(x, y)$=exp$(-1/\gamma^2(x-y)^2)$ where d is the degree of the polynomial kernel and γ^2 is the bandwidth of the Gaussian radial basis function kernel. Choosing the proper kernel and proper values for its parameters is essential in order to obtain the best model [34].

3.3 Summary

This chapter presented the overall methodology used in this study. The links to the individual chapters, to which each step of the methodology relates, is also given. Additionally, it discussed the algorithms this study used for predicting trading signals.

The next chapter (Chapter 4) covers the first and the second steps of the methodology those of developing a technique for quantifying intermarket influences from a given set of potential influential markets on a selected dependent market and applying this technique for quantifying intermarket influences on the AORD, respectively.

Chapter 4

Quantification of Intermarket Influences on the Australian All Ordinary Index

4.1 Introduction

As described in Section 1.1.2, a significant impact from the lagged prices (or derivative thereof) of a stock market index on the current price (or derivative of price) of a given stock market can be defined as the influence from the former market on the latter. If the performance of two markets are interrelated, then Ruggiero [73] defined one market as an *intermarket* of other. The influence of an intermarket on another can be defined as an intermarket influence.

Currently intermarket influence is an important consideration among investors and decision makers. However, no techniques for quantification of intermarket influences were introduced in the literature (Section 2.4). Discovering and formalizing intermarket influence patterns is likely to prove extremely useful in many applications such as market prediction, portfolio optimization and management. Recent studies [63, 65, 66] have shown that the consideration of intermarket influences as input variables, improves the forecast accuracy.

Past studies confirmed that most of the world's major stock markets are integrated (Section 2.4). Hence, such an integrated market can be considered as a part of a single global system. The influence from an integrated stock market on a dependent market may also include the influence from one or more stock markets on the former.

If there is a set of influential markets to a given dependent market, it is not straightforward to separate the influence from individual influential markets. Instead of measuring the individual influence from one influential market on the dependent market, the strength of the influence from this influential market on the dependent market can be measured compared to the influence from the other influential markets. This study uses a novel approach to quantify intermarket influences. This approach estimates the combined influence of a set of influential markets and the contribution from each influential market to the combined influence.

This chapter focuses on developing a new technique to quantify intermarket influences. This technique is applied to quantify intermarket influences from a selected set of world's major stock markets on the AORD. The quantification is carried out by different time lags and different time periods.

4.2 Quantification of Intermarket Influences

As mentioned earlier in Section 1.1.2, intermarket influence may impact on price and/or one or more derivative properties of price. To achieve the objectives of this study (to predict the direction of the Close price and to predict whether it is best to buy, hold or sell, on day $(t + 1)$), we have only two options to consider: analysing intermarket influence on either Close price or return (such as relative return) of the Close price. We opted analysing intermarket influence on relative returns of the Close price, since returns for different stock indices are comparable.

Quantification of intermarket influences on the AORD was carried out by finding the coefficients, ξ_j s which maximise the median rank correlation between the relative return of Close price of day t of the AORD and the sum of ξ_j multiplied by the lagged relative returns of Close prices of the potential influential markets over a number of small non-overlapping windows of a fixed size. ξ_j measures the contribution of the jth influential market to the combined influence which equals to the optimal correlation. This coefficient will be termed *quantification coefficient*.

The objective function to be maximised is defined by Spearman's correlation coefficients calculated on these windows. The description of the objective function is given in Section 4.2.1.

In this study, we used the global optimization algorithm developed in [47, 48]. A brief description of this algorithm is given in Section 4.2.2. The performance of this algorithm has been demonstrated in solving different optimization problems including discontinuous objective functions (for example [39]), which is the case in our study as well.

Spearman's rank correlation coefficient (see page 54 for more descriptions) was used instead of the more commonly used Pearson's correlation coefficient, for the following reasons:

- Stock market time series are generally non-linear and non-stationary (variance varies with time). Unlike Pearson's correlation coefficient, rank correlation measurers (such as Spearman's rank correlation coefficient) assess how well an arbitrary monotonic function can describe the relationship between two variables.

- Spearman's rank correlation coefficient is a non-parametric measure of correlation. No assumptions about frequency distributions of variables are required.

Since, influential patterns are likely to vary with time [83], the whole study period was divided into a number of moving windows of a fixed length. The correlation structure between stock markets also changes with time [96]. Therefore, each moving window was further divided into a number of small windows of 22 days in length. 22 days of a stock market time series represent a trading month. The rank correlation coefficients were calculated for these smaller windows within each moving window.

The other important matter with regard to this optimization procedure is that the absolute value of the correlation coefficient was considered when finding the optimal median correlation. This is appropriate as we are interested in the strength rather than the direction of the correlation (that is either positively or negatively correlated).

4.2.1 Optimization Problem

Let $Y(t)$ be the relative return of the Close price of a selected dependent market at time t and $X_j(t)$ be the relative return of the Close price of the jth influential market at time t. Also let:

$$X_\xi(t-i) = \sum_j \xi_j X_j(t-i) \tag{4.1}$$

where $\xi_j \geq 0$, $j = 1, 2, ..., m$, is the quantification coefficient associated with the jth influential market X_j. m is the total number of influential markets and i is the time lag.

The aim is to find the optimal values of the quantification coefficients, $\xi = (\xi_1, ..., \xi_m)$ which maximise the rank correlation between $Y(t)$ and $X_\xi(t-i)$ for a given window and time lag i. In the calculations, $i = 0, 1, 2, 3, 4$, which represent correlation/influence within a week, were considered. $i = 0$ represents the same day correlation between the Close price of the dependent market and a selected combination of the Closes price of influential markets. $i = 1$ gives the correlation between the Close price of day t of the dependent market and the Close prices of day $(t-1)$ of a combination of influential markets and this correlation is referred as the previous day's (day $(t-1)$) combined influence from this combination of influential markets on the dependent market. Other time lags can be defined in a similar manner.

The correlation can be calculated for a window of a given size. This window can be defined as:

$$T(t^0, l) = \{t^0, t^0+1, ..., t^0+(l-1)\} \tag{4.2}$$

where t^0 is the starting date of the window and l is its size (in days).

The correlation between the variables $Y(t), X_\xi(t-i),\ t \in T(t^0, l)$, defined on the window $T(t^0, l)$, will be denoted by $C^i(\xi)$:

$$C^i(\xi) = \text{Corr}(Y(t), X_\xi(t-i) \parallel T(t^0, l)). \tag{4.3}$$

For a particular window $T(t^0, l)$, the following optimization problem can be formulated:

$$\text{Maximise} \quad C^i(\xi);$$

$$\text{s.t.} \quad \sum_j \xi_j = 1, \quad \xi_j \geq 0 \quad j = 1, 2, ..., m \tag{4.4}$$

In this way, the optimal quantification coefficients, ξ_j, are obtained for a given combination of influential markets, on the fixed window $T(t^0, l)$. For a period of several years, the optimal correlation changes according to the starting point of the window.

To define optimal values of ξ for a long time period, the following method is applied: let $[1, T] = 1, 2, ..., T$ be a given period (for instance, a large window). This period is divided into n windows of size l (we assume that T is divisible by l):

$$T(t_k, l), \quad k = 1, 2, 3, ..., n; \tag{4.5}$$

so that,

$$T(t_k, l) \cap T(t_{k'}, l) = \phi \quad \text{for} \quad \forall \;\; k \neq k', \tag{4.6}$$

$$\bigcup_{k=1}^{n} T(t_k, l) = [1, T]. \tag{4.7}$$

For given i, the correlation coefficient on a window $T(t_k, l)$ is defined as:

$$C^i_k(\xi) = \text{Corr}(Y(t), X_\xi(t-i) \parallel T(t_k, l)), \quad k = 1, ..., n. \tag{4.8}$$

To define the objective function over the period $[1, T]$, the median of the vector, $(C^i_1(\xi), ..., C^i_n(\xi))$ is used. Now, the main optimization problem can be redefined as:

Problem (P):

$$\text{Maximise} \quad f^i(\xi) = \text{Median} \;\; (C^i_1(\xi), ..., C^i_n(\xi)); \tag{4.9}$$

$$\text{s.t.} \quad \sum_j \xi_j = 1, \quad \xi_j \geq 0 \quad j = 1, 2, ..., m. \tag{4.10}$$

Rank correlation measure

The Spearman's rank correlation coefficient is used in (4.8) as the measure of correlation. Given two variables X and Y, the Spearman's rank correlation coefficient, r_s, can be defined as;

$$r_s = \frac{n(n^2-1) - 6\sum d_i^2 - (T_x + T_y)/2}{\sqrt{(n(n^2-1) - T_x)(n(n^2-1) - T_y)}} \tag{4.11}$$

where:

$n-$ total number of bivariate observations;

d_i- difference between the rank of x and the rank of y in the ith observation;

T_x- number of tied observations of X; and

T_y- number of tied observations of Y.

The Spearman's Rank Correlation depends on the rank of the given vectors. According to Equation (4.1), it is obvious that the rank order of the elements of $X_\xi(t-i)$ doest not change if ξ is replaced by $\lambda\,\xi$, where $\lambda > 0$. In other words, the corresponding elements of the vectors $X_\xi(t-i)$ and $X_{\lambda\,\xi}(t-i)$ have the same rank order. This means that the objective function $f^i(\xi)$, in (4.9), satisfies the following condition:

$$f^i(\lambda\,\xi) = f^i(\xi), \quad \text{for} \quad \text{all} \quad \lambda > 0. \tag{4.12}$$

4.2.2 Global Optimization Algorithm

The objective function $f^i(\xi)$, in (4.9), is not only discontinous, but also piece-wise constant. This is because, for each window k, the correlation coefficient $C_k^i(\xi)$ is a piece-wise constant function as it depends on the ranking of the vectors $Y(t)$ and $X_\xi(t-i)$. Solving this type of optimization problems is extremely difficult. The majority of available algorithms need smoothness or at least semi-smoothness of the objective functions to be minimized.

In this study, the algorithm AGOP, developed in [47] and [48], was applied to solve the optimization problem of interest. This algorithm was designed for continuous optimization problems with box constraints. It uses a line search mechanism where the descent direction is obtained via a dynamic systems approach. It is applicable to a wide range of optimization problems requiring only function evaluations to work. In particular it does not require gradient information and can be used to find minima of non-smooth functions.

The AGOP algorithm will now be described in terms of the minimizing the function $g(\xi) = -f^i(\xi)$. AGOP must first be given a set of initial points, say $\Omega = \{\xi^1, ..., \xi^q\} \subset R^m$, $q \geq 2$. Generally, a suitable choice for an initial set of points is the set of some vertices of a given box. Let $\xi^\star \in \Omega$ be the point in Ω with the smallest cost, that is, $g(\xi^\star) \leq g(\xi)$ for all $\xi \in \Omega$. The set Ω and the values of g at each of the points in Ω allow us to determine a vector v to be used as a possible descent direction from point $\xi^\star$. An inexact line search along the direction of v provides a new point $\hat{\xi}^{q+1} \neq \xi^\star$. A local search around $\hat{\xi}^{q+1}$ is then carried out using a method called *local variation*. This is an efficient local optimization technique that does not explicitly use derivatives and can be applied to non-smooth functions. A good survey of direct search methods can be found in [37]. Letting ξ^{q+1} denote the optimal solution of this local search, the set Ω is augmented to include ξ^{q+1}. Starting with this updated Ω, the whole process can be repeated. The process terminates when v is approximately 0 (or a prescribed bound on the number of iterations is reached). The solution returned is the current $\xi^\star$, that is, the point in Ω with the smallest cost.

To solve Problem **(P)** that contains equality constraints (4.10), property (4.12) will be used. Consider the following problem with box constraints:

Problem (P1):

$$\text{Maximise} \quad f^i(\xi) = \text{ Median } \quad (C_1^i(\xi), ..., C_n^i(\xi));$$

$$\text{s.t.} \quad \xi_j \in [0, 1], \quad j = 1, 2, ..., m. \tag{4.13}$$

It is easy to verify that the following property is true:

Proposition 1 *Let ξ^* be a global optimal solution to Problem* **(P1)**. *Then*

$$\xi = \xi^*/\lambda^*, \quad where \quad \lambda^* = \xi_1^* + ... + \xi_m^*, \tag{4.14}$$

is a global optimal solution to Problem **(P).**

Therefore, to solve Problem **(P)**, first the algorithm AGOP is applied to Problem **(P1)**, taking $g(\xi) = -f^i(\xi)$, and get a solution ξ^*. Then using transformation (4.14), a solution ξ to the original problem with equality constraints is obtained.

Another way to handle the equality constraints in (4.10) would be to eliminate one variable, say ξ_m, taking $\xi_m = 1 - \xi_1 - ... - \xi_{m-1}$. In this case, we would have the following constraints, instead of (4.10);

$$\xi_1 + ... + \xi_{m-1} \leq 1, \quad \xi_j \geq 0 \quad j = 1, 2, ..., m-1. \tag{4.15}$$

Then, the problem (4.9, 4.15), becomes an optimization problem with inequality constraints, which is much easier than the original Problem **(P)**. However, solving problem (4.9, 4.15) is very difficult because the objective function is discontinuous. That is why, solving Problem **(P1)**, that uses only box constraints is preferable in terms of finding better solutions.

4.3 Data and Data Preprocessing

The data set consists of daily relative returns of the Close prices of ten potential influential stock markets and the AORD, from 2nd July 1997 to 30th December 2005. The selected potential influential markets are:

- US S&P 500 Index (GSPC),
- US Nasdaq Composite Index (IXIC),
- UK FTSE 100 Index (FTSE),

- French CAC 40 Index (FCHI),
- German DAX Index (GDAXI),
- Hong Kong Hang Seng Index (HSI),
- Singapore Straits Times Index(STI),
- Japanese Nikkei 225 Index (N225),
- Chinese SSE Composite Index (SSEC),
- Taiwan TSEC Weighted Index (TWII).

The GSPC and IXIC are widely considered as market leaders. The FTSE, FCHI, and GDAXI are major European stock market indices, while the HSI, STI, N225, SSEC and TWII are major Asian stock market indices.

Since different stock markets are closed on different holidays, the regular time series data sets considered have missing values. If no trading took place on a particular day, the rate of change of price should be zero. Therefore, the missing values of the Close price were replaced by the corresponding Close price of the last trading day.

Relative Returns, RR, of the daily Close price of the stock market indices were used for the analysis:

$$RR(t) = \frac{P(t) - P(t-1)}{P(t-1)} \tag{4.16}$$

where $RR(t)$ and $P(t)$ are the relative return and the Close price of a selected index on day t, respectively. Returns are preferred to price, since returns for different stocks are comparable on equal basis.

It is worth noting that the opening and closing times for many of the various markets do not coincide. For example, the Australian, Asian, French and German markets have all closed by the time the US markets open.

4.4 Description of Quantification Experiments

Firstly, the quantification of intermarket influences was carried out for different time lags by considering the whole study period as a single window. Different market combinations were incorporated for this analysis. In this analysis, we expected to investigate the impact of intermarket influences at different time lags.

As mentioned in Section 4.2, the influential patterns between markets are likely to vary with time [83]. Therefore, the quantification process needs to be carried out for different time periods (windows).

Secondly, the quantification process was repeated for different time periods (windows) by considering only the time lag 1. From the previous quantification process (described above), we identified only the intermarket influences at time lag 1 has a significant impact on the AORD (Table 4.1). The quantification coefficients obtained in this process are to be used for the predictions related to the AORD. This goal requires the quantification coefficients to be calculated using 'known data' (that is training data).

4.4.1 Quantification of Intermarket Influences for the Whole Study Period

The quantification coefficients which maximise the median Spearman's rank correlation between the relative return of the Close price of day t of the AORD and the sum of the quantification coefficient multiplied by the lagged relative returns of the Close prices of the potential influential markets were found by considering the whole study period as a single window. This procedure was carried out for different combinations of influential markets. These combinations (with their notations) are:

- GSPC+European markets $\equiv$ (A),
- GSPC+European markets+AORD $\equiv$ (B),
- GSPC+European markets+Asian markets $\equiv$ (C),

- GSPC+European markets+Asian markets+AORD ≡ (D),
- US markets ≡ (E),
- US markets+European markets ≡ (F),
- US markets+European markets+Asian markets ≡ (G),
- US markets+European markets+Asian markets+AORD ≡ (H).

Past studies [65, 82] evidence that the lagged Close price of the AORD itself shows an impact on the Close price of day t of the AORD. Therefore, in addition to the lagged Close prices of the global markets, those of the AORD were also taken into account when forming the market combinations.

The quantification results obtained by these experiments are presented in Section 4.5.

4.4.2 Quantification of Intermarket Influences for Different Time Periods (Training Windows)

The whole study period was divided into six moving windows of three trading years (for stock market time series, 256 days is considered as a trading year). Each time, a window was shifted forward by one trading year in order to get the starting point of the next window. Each window was divided into two parts; the most recent 10% of data (test set) was separated and this portion was allocated for evaluating the predictions. The remainder (that is 90% of data from the beginning of the window) is called a *training window* and this window was used to estimate the quantification coefficients.

For each training window, the quantification coefficients which maximise the median Spearman's rank correlation between the relative return of the Close price of day t of the AORD and the sum of the quantification coefficients multiplied by the respective relative returns of day $(t-1)$ of the Close prices of the potential influential markets were derived. The second quantification process was also carried out for the same combinations of influential markets that were used for the first quantification process (Section 4.4.1).

Section 4.6 presents the results obtained from this second quantification process.

4.5 Numerical Experiments: Quantification of Intermarket Influences for the Whole Study Period

The quantification (of intermarket influences) results at different time lags, derived considering the whole study period as a single window (Section 4.4.1), are presented in this section. In addition to these results, this section discusses some issues identified relating to the optimization problem of interest.

Table 4.1 presents the optimal median Spearman's correlations relevant to the above mentioned market combinations (Section 4.4) at different time lags.

Table 4.1: Optimal median Spearman's correlations at different time lags for different market combinations

Time lag	Market combination							
	(A)	(B)	(C)	(D)	(E)	(F)	(G)	(H)
0	0.275	NA[2]	0.520[1]	NA[2]	0.160	0.275	0.517[1]	NA[2]
1	0.553[1]	0.553[1]	0.554[1]	0.554[1]	0.532[1]	0.554[1]	0.555[1]	0.555[1]
2	0.169	0.182	0.225	0.221	0.153	0.169	0.227	0.228
3	0.195	0.196	0.222	0.220	0.184	0.205	0.223	0.225
4	0.182	0.181	0.216	0.220	0.156	0.194	0.215	0.216

[1]correlations significant at 5% level (one sided critical value for Spearman's correlation at 5% level is 0.425 [76]).

[2]correlation between the relative return of day t of the AORD with the sum of quantified relative return of the same day of a market combination which includes the AORD is not meaningful.

Note: Quantification coefficient multiplied by the respective relative return is termed *quantified relative return.*

Spearman's correlation for a given market combination measures the strength of the combined influence from the markets included in this combination on the AORD. Correlation at time lag 0 represents the same day correlation between the relative return of the AORD and the sum of the quantified relative returns of a given market combination.

At time lag 0, only the correlations relating to the market combinations (B) and (F) are significant (Table 4.1). The reason may be that these two combinations include the Asian markets. There is a period in the trading day in which the Australian and the Asian markets are open simultaneously, and therefore, these markets share the information at the same time.

Correlations at time lag 1 indicates the combined influence from the Close prices of day $(t-1)$ of a given market combination on the AORD Close price. All the correlations at lag 1 are significant (Table 4.1). Market combination (E), which includes only the US markets yielded the lowest correlation at lag 1 (0.532). The next lowest correlation is corresponding to the market combinations (A) and (B) and this value is 0.553. (A) includes the GSPC and the European markets while the AORD is included in (B) in addition to the four indices included in (A). However, there are no substantial differences among the correlations (at time lag 1) corresponding to all market combinations considered, except (E). Therefore, it can be suggested that adding other markets to the combination which includes the GSPC and the European markets did not substantially increase the combined influence on the AORD.

The correlations at the time lags greater than one are not significant (Table 4.1). This indicates that the Close prices of two or more days in the past of the considered market combinations did not have a significant impact on the current day's Close price of the AORD.

The quantification coefficient, ξ, relating to a particular market indicates the contribution from that market to the combined influence compared to the contribution from the other markets considered. Therefore, ξ corresponding to a market can be used as a measure of the contribution of this market to the combined influence. For this purpose it is sufficient to consider the influential market combination (G), since it covers all potential global influential markets considered.

Table 4.2 presents the quantification coefficients for all influential markets considered at time lags 0 and 1. None of the correlations at time lags greater than one are significant (Table 4.1). Therefore, it is not worth considering the quantification coefficients relating to these time lags.

Table 4.2: The optimal values of the quantification coefficients (ξ) which maximise the median Spearman's correlation coefficient between the relative return of the Close price of day t of the AORD and the sum of the quantified coefficient multiplied by lagged relative returns of the influential market combination (G), for the whole study period

Stock market index	Optimal values of ξ	
	at lag 0	at lag 1
GSPC	0.0626	0.7958
IXIC	0.0	0.0
FTSE	0.0	0.1487
FCHI	0.1589	0.0
GDAXI	0.0	0.0222
HSI	0.1290	0.0
STI	0.2161	0.0
N225	0.2582	0.0111
SSEC	0.1450	0.0
TWII	0.0303	0.0222
Optimal Spearman's correlation coefficient	0.5172	0.5554

At time lag 1, the GSPC had the highest contribution to the combined influence on the AORD followed by the FTSE (Table 4.2). The Close prices of the Asian markets (at lag 0), particularly the STI and the N225 were highly correlated with that of the AORD.

4.5.1 Some Issues Related to the Optimization Problem (P1)

As mentioned in Section 4.2.2, due the discontinuity of the objective function, Problem **(P1)** (or Problem **(P)**) is a difficult global optimization problem. Therefore, there is no guarantee that the results presented in Table 4.1 are the global optimal solutions. Nevertheless, the results are quite reasonable in the sense that including extra market leads to a higher optimal correlation coefficient. Following this idea, one can expect the optimal median Spearman's correlations to agree with the following conditions;

- (A)≤(B), (A)≤(C), (A)≤(D), (A)≤(F), (A)≤(G), (A)≤(H),
- (B)≤(D), (B)≤(H),
- (C)≤(D), (C)≤(G), (C)≤(H),
- (D)≤(H),
- (E)≤(F), (E)≤(G), (E)≤(H),
- (F)≤(G), (F)≤(H), and
- (G)≤(H).

There do appear to some exceptions (for example, (C) vs (G) at lag 0, (C) vs (D) at lag 2, (A) vs (B), (C) vs (G) and (D) vs (H) at lag 4), however, the values of Spearman's correlation coefficient in these cases are so similar as not to represent a significant difference. In all other cases, the optimal Spearman's correlation agrees with the above mentioned conditions. Therefore, applying the optimization algorithm described in Section 4.2.2, one can expect to obtain quite reasonable (close to global optimal) solutions to Problem **(P1)**.

4.6 Numerical Experiments: Quantification of Intermarket Influences for Different Training Windows

The quantification results, obtained for the six windows which are described in Section 4.4.2, are presented in this section.

4.6.1 Quantification of Intermarket Influences from the Market Combinations (A) to (D)

Table 4.3 to 4.6 present the optimal values of the quantification coefficients (ξ) together with the optimal median Spearman's correlations corresponding to the market combinations (A) to (D) (Section 4.4), for different windows, respectively. These market combinations do not include the IXIC Index.

Table 4.3: Optimal values of quantification coefficients (ξ) and the optimal median Spearman's correlations corresponding to market combination (A) for different moving windows

Training Window No.	Optimal values of ξ				Optimal median Spearman's correlation
	GSPC	FTSE	FCHI	GDAXI	
1	0.5720	0.2905	0.1141	0.0233	0.5782[1]
2	0.6124	0.1825	0.0787	0.1264	0.5478[1]
3	0.7656	0.0931	0.1328	0.0085	0.5680[1]
4	0.7946	0.0562	0.1492	0.0000	0.5790[1]
5	0.5572	0.1720	0.0346	0.2362	0.5904[1]
6	0.6658	0.0583	0.0795	0.1964	0.5359[1]

[1]correlations significant at 5% level (one sided critical value for Spearman's correlation at 5% level is 0.425 [76]).

Table 4.3 shows that the Spearman's correlation corresponding to each window is significant (at the 5% level). This implies that the Close prices of day $(t-1)$ of the markets included in combination (A) had a significance combined influence on the Close price of day t of the AORD in each window considered. Irrespective of the window, the GSPC had the highest contribution to the combined influence and its contribution was more than 50%. The FTSE had the second highest contribution in the first and the second window. In the next two windows, the FCHI showed the second highest contribution. In the last two windows, the GDAXI showed the second highest contribution. This confirms that the influence patterns change with time.

Table 4.4: Optimal values of quantification coefficients (ξ) and the optimal median Spearman's correlations corresponding to market combination (B) for different moving windows

Training	Optimal values of ξ					Optimal median
Window No.	GSPC	FTSE	FCHI	GDAXI	AORD1	Spearman's correlation
1	0.5621	0.2929	0.0997	0.0282	0.0170	0.5805[1]
2	0.5780	0.1083	0.1266	0.1688	0.0183	0.5500[1]
3	0.7369	0.0000	0.1730	0.0175	0.0726	0.5697[1]
4	0.7898	0.0701	0.1401	0.0000	0.0000	0.5799[1]
5	0.5569	0.1699	0.0394	0.2338	0.0000	0.5904[1]
6	0.6592	0.0431	0.0900	0.1957	0.0120	0.5368[1]

[1]correlations significant at 5% level (one sided critical value for Spearman's correlation at 5% level is 0.425 [76]).

The Spearman's correlation relating to market combination (B) is significant in each window (Table 4.4). When the AORD was added to the GSPC and the three European market indices, the correlation increased in each window, except the fifth window (Table 4.3 and Table 4.4). In the fifth window optimal median correlation remained unchanged. It is noteworthy that the quantification coefficient relevant to the AORD is zero, but those of the other markets are different from the respective quantification coefficients when the input set consists of only the GSPC, FTSE, FCHI and GDAXI. This indicates that Problem **(P1)** has multiple optimal solutions. However, the quantification coefficients relevant to the two optimal solutions are approximately the same.

As in the previous case (Table 4.3), the GSPC contributes more than 50% to the combined influence. Surprisingly, the previous day's Close price of the AORD had the least contribution in all windows, except the third window.

Table 4.5: Optimal values of quantification coefficients (ξ) and the optimal median Spearman's correlations corresponding to market combination (C) for different moving windows

Training Window No.	Optimal values of ξ						
	GSPC	FTSE	FCHI	GDAXI	HSI	STI	N225
1	0.5681	0.2894	0.1195	0.0187	0.0042	0.0000	0.0000
2	0.5975	0.1198	0.1122	0.1420	0.0040	0.0000	0.0000
3	0.5818	0.0168	0.2684	0.0168	0.0000	0.0000	0.0000
4	0.7994	0.0000	0.1072	0.0000	0.0000	0.0311	0.0000
5	0.7000	0.1133	0.1497	0.0117	0.0000	0.0008	0.0129
6	0.5500	0.1916	0.1344	0.0499	0.0166	0.0000	0.0000

Training Window No.	Optimal values of ξ		Optimal median Spearman's correlation
	SSEC	TWII	
1	0.0000	0.0000	0.5782[1]
2	0.0246	0.0000	0.5663[1]
3	0.1162	0.0000	0.5782[1]
4	0.0623	0.0000	0.5974[1]
5	0.0000	0.0117	0.5906[1]
6	0.0000	0.0575	0.5404[1]

[1]correlations significant at 5% level (one sided critical value for Spearman's correlation at 5% level is 0.425 [76]).

Irrespective of the window, the correlation is significant which indicates that there was a significance combined influence from the Close prices of day $(t-1)$ of the markets included in combination (C), on the Close price of day t of the AORD. Still the GSPC shows the highest contribution to the combined influence. In the first window, no Asian markets other than the HSI contributed to the combined influence. Except for the fifth window the N225 did not contribute and its contribution in this window also very small.

Spearman's correlation relating to all windows, except the first, increased when the quantified relative return of day $(t-1)$ of the Asian markets were added to those of the GSPC and the European markets (Table 4.3 and Table 4.5). The correlation corresponding to the first window remained unchanged. It is noteworthy that the quantification coefficients relevant to all the Asian markets except the HSI, are zero, and those of the US and the European markets are different from their respective quantification coefficients shown in Table 4.3. This implies that the maximum value of the objective function (optimal median Spearman's correlation) may be achieved at different points. However, the values of the quantification coefficient relevant to the two optimal solutions are approximately the same.

Table 4.6 evidences that the Close prices of day $(t-1)$ of market combination (D) had a significance combined influence on the Close price of day t of the AORD during the study period. Also it indicates that the GSPC had the highest contribution among the indices included in market combination (D) while the N225 did not have any contribution in any window. Surprisingly, the Close price of day $(t-1)$ of at least one European market showed stronger influence on the Close price of day t of the AORD than its Close price of day $(t-1)$, during the study period.

When comparing the markets combinations (B) and (D), the optimal median correlation corresponding to (D) is greater than that of (B) in each window (Table 4.4 and 4.6). The correlation remained unchanged in the last window, when the quantified relative return of day $(t-1)$ of the AORD was added to those of the market indices included in (C). The quantification coefficient relevant to the AORD is zero while these coefficients relevant to most of the other indices are different from their respective values in Table 4.5. Another notable issue is that the correlation reduced in the second and the third windows, when an extra index was added to market combination (C) (Table 4.5 and 4.6). However, there is no substantial reduction.

Table 4.6: Optimal values of quantification coefficients (ξ) and the optimal median Spearman's correlations corresponding to market combination (D) for different moving windows

Training Window No.	Optimal values of ξ						
	GSPC	FTSE	FCHI	GDAXI	HSI	STI	N225
1	0.5635	0.2828	0.1039	0.0263	0.0059	0.0000	0.0000
2	0.5902	0.1051	0.1234	0.1559	0.0000	0.0000	0.0000
3	0.6409	0.0168	0.1575	0.0000	0.0000	0.0413	0.0000
4	0.7822	0.0000	0.1069	0.0000	0.0050	0.0292	0.0000
5	0.5648	0.2482	0.0535	0.0000	0.0000	0.0000	0.0000
6	0.5432	0.1700	0.1325	0.0685	0.0335	0.0000	0.0000

Training Window No.	Optimal values of ξ			Optimal median Spearman's correlation
	SSEC	TWII	AORD1[3]	
1	0.0025	0.0000	0.0151	0.5822[1]
2	0.0254	0.0000	0.0000	0.5635[1]
3	0.0398	0.0000	0.1036	0.5754[1]
4	0.0643	0.0000	0.0124	0.5985[1]
5	0.1292	0.0000	0.0044	0.5951[1]
6	0.0000	0.0523	0.0000	0.5404[1]

[1]correlations significant at 5% level (one sided critical value for Spearman's correlation at 5% level is 0.425 [76]).

[3]AORD1 $\equiv$ AORD(t-1)

4.6.2 Quantification of Intermarket Influences from the Market Combinations (E) to (H)

The optimal values of quantification coefficients (ξ) together with the optimal median Spearman's correlations corresponding to the market combinations (E) to (H) (Section 4.4), for different moving windows are shown in Table 4.7 to 4.10, respectively. Unlike the market combinations (A) to (D), these market combinations include the IXIC Index.

Table 4.7: Optimal values of quantification coefficients (ξ) and the optimal median Spearman's correlations corresponding to market combination (D) for different moving windows

Training Window No.	Optimal values of ξ		Optimal median Spearman's correlation
	GSPC	IXIC	
1	0.7228	0.2772	0.5289[1]
2	0.8974	0.1026	0.4972[1]
3	0.9299	0.0701	0.5210[1]
4	1.0000	0.0000	0.5505[1]
5	0.7270	0.2730	0.5566[1]
6	0.7834	0.2166	0.4946[1]

[1]correlations significant at 5% level (one sided critical value for Spearman's correlation at 5% level is 0.425 [76]).

The combined influence from the Close price of day $(t-1)$ of the stock market indices that belong to combination (E) is significant in each window (Table 4.7). The contribution of the GSPC is much greater than that of the IXIC. In the forth window, the GSPC contributes 100% to the combined influence.

Irrespective of the window, the Close price of day $(t-1)$ of the indices included in combination (F) had a significance combined influence on the Close price of day t of the AORD (Table 4.8). The GSPC showed the highest contribution in each window.

The addition of the three European stock market indices to the market combination of the GSPC and the IXIC (combination (F)) improved the correlation by substantial amounts (10.49%, 10.06%, 9.14%, 4.61%, 7.13% and 8.29% in windows 1 to 6, respectively; Table 4.7 and Table 4.8).

Table 4.9 evidences that there was a significance combined influence from the Close prices of day $(t-1)$ of the indices belong to combination (F). The GSPC had the highest contribution to the combined influence from this market combination. Irrespective of the window, the N225 did not show any contribution.

Table 4.8: Optimal weights and the optimal median Spearman's correlations corresponding to market combination (F) for different moving windows

Training Window No.	Optimal values of ξ					Optimal median Spearman's correlation
	GSPC	IXIC	FTSE	FCHI	GDAXI	
1	0.5310	0.0520	0.2879	0.1041	0.0249	0.5844[1]
2	0.5886	0.0000	0.1886	0.0639	0.1590	0.5472[1]
3	0.4864	0.0353	0.3308	0.1267	0.0207	0.5686[1]
4	0.8707	0.0000	0.0000	0.1293	0.0000	0.5759[1]
5	0.4426	0.1870	0.1997	0.1510	0.0197	0.5963[1]
6	0.5473	0.1200	0.3163	0.1633	0.0000	0.5356[1]

[1]correlations significant at 5% level (one sided critical value for Spearman's correlation at 5% level is 0.425 [76]).

Except for the first window, the addition of the Asian markets to combination (G) improved the correlations by substantial amounts (Table 4.8 and Table 4.9). The correlation corresponding to the first window was reduced when the Asian markets were added to (H), however, this drop is not a substantial drop.

The GSPC had the highest contribution to the combined influence from the market combination (H) (Table 4.10). Both the N225 and the AORD did not show any contribution.

When the AORD was added to the market combination (G), the correlation decreased in all windows except the first and the fifth windows (Table 4.9 and Table 4.10).

4.7 Possible Variations of the Quantification Coefficients

We investigated the possible variations in the optimal values of quantification coefficients relevant to a given market combination, when the respective optimal median correlation varies by a small value.

Let $\xi^*=(\xi_1^*,\ldots,\xi_i^*,\ldots,\xi_m^*))$ be the vector of quantification coefficients corresponding

Table 4.9: Optimal values of quantification coefficients (ξ) and the optimal median Spearman's correlations corresponding to market combination (G) for different moving windows

Training Window No.	Optimal values of ξ						
	GSPC	IXIC	FTSE	FCHI	GDAXI	HSI	STI
1	0.5537	0.0332	0.2712	0.0957	0.0319	0.0142	0.0000
2	0.5923	0.0000	0.1110	0.1195	0.1426	0.0069	0.0000
3	0.6635	0.0758	0.0000	0.1986	0.0000	0.0000	0.0000
4	0.7899	0.0000	0.0000	0.1192	0.0026	0.0000	0.0208
5	0.3439	0.1961	0.2171	0.0895	0.0701	0.0000	0.0740
6	0.4461	0.2104	0.2279	0.0199	0.0000	0.0890	0.0000

Training Window No.	Optimal values of ξ			Optimal median Spearman's correlation
	N225	SSEC	TWII	
1	0.0000	0.0000	0.0000	0.5839[1]
2	0.0000	0.0277	0.0000	0.5659[1]
3	0.0000	0.0517	0.0103	0.5785[1]
4	0.0000	0.0675	0.0000	0.5991[1]
5	0.0000	0.0093	0.0000	0.6000[1]
6	0.0000	0.0066	0.0000	0.5409[1]

[1]correlations significant at 5% level (one sided critical value for Spearman's correlation at 5% level is 0.425 [76]).

Table 4.10: Optimal values of quantification coefficients (ξ) and the optimal median Spearman's correlations corresponding to market combination (H) for different moving windows

Training Window No.	Optimal values of ξ						
	GSPC	IXIC	FTSE	FCHI	GDAXI	HSI	STI
1	0.5375	0.0433	0.2817	0.1097	0.0271	0.0006	0.0000
2	0.5825	0.0000	0.0960	0.1348	0.1534	0.0000	0.0000
3	0.5758	0.0000	0.0194	0.2689	0.0000	0.0000	0.0194
4	0.8434	0.0000	0.0550	0.0339	0.0000	0.0000	0.0677
5	0.3389	0.1996	0.2183	0.0767	0.0767	0.0000	0.0792
6	0.5269	0.1535	0.2831	0.0000	0.0000	0.0243	0.0000

Training Window No.	Optimal values of ξ				Optimal median Spearman's correlation
	N225	SSEC	TWII	AORD1[3]	
1	0.0000	0.0000	0.0000	0.0000	0.5850[1]
2	0.0000	0.0333	0.0000	0.0000	0.5635[1]
3	0.0000	0.1165	0.0000	0.0000	0.5743[1]
4	0.0000	0.0000	0.0000	0.0000	0.5867[1]
5	0.0000	0.0106	0.0000	0.0000	0.6011[1]
6	0.0000	0.0066	0.0122	0.0000	0.5398[1]

[1]correlations significant at 5% level (one sided critical value for Spearman's correlation at 5% level is 0.425 [76]).

[3]AORD1 ≡ AORD(t-1)

to a selected combination of m (m is an positive integer) markets. Also let ξ^* is an optimal solution to the problem (P) ((4.9) and (4.10) in Section 4.2.1) with time lag i=1.

Given a vector ξ=$(\xi_1, \ldots, \xi_m)$, the value of the objective function in (4.9) will be denoted by $C(\xi)$=$f^1(\xi)$=Median$(C_1^1(\xi), \ldots, C_n^1(\xi))$ (refer Section 4.2.1 for the meaning of the objective function). We also denote C^*=$C(\xi^*)$.

We aimed at finding the possible range for quantification coefficient associated with a given market, which provides a close to optimal value for median correlation. For this purpose, we consider the maximum variation for the quantification coefficient associated with this market (assuming that the quantification coefficients relevant to other markets are fixed), given that the median correlation does not lie below $C^*(1-\varepsilon)$. In the experiments, we choose ε=0.05 which corresponding to the 95% of the optimal value of the correlation.

For a given market $i \in \{1, \ldots, m\}$, we define:

$$\xi_i^{\max} = \text{Max}\{\xi \in [\xi_i^*, 1], C(\xi_1^*, \ldots, \xi_{i-1}^*, \xi, \xi_{i+1}^*, \ldots, \xi_m^*) \geq C^*(1-\varepsilon)\},$$
$$\text{for all } \tilde{\xi}_i \in [\xi_i^*, \xi_i]; \qquad (4.17)$$

and,

$$\xi_i^{\min} = \text{Min}\{\xi \in [0, \xi_i^*], C(\xi_1^*, \ldots, \xi_{i-1}^*, \xi, \xi_{i+1}^*, \ldots, \xi_m^*) \geq C^*(1-\varepsilon)\},$$
$$\text{for all } \tilde{\xi}_i \in [\xi_i, \xi_i^*]. \qquad (4.18)$$

Then the interval $[\xi_i^{\min}, \xi_i^{\max}]$, is the interval that the quantification coefficient associated with ith ($i = 1, \ldots, m$) market can vary, while the corresponding correlation lies within the interval $[C^*(1-\varepsilon), C^*]$.

To study the possible variations in the optimal values of quantification coefficients, we considered the market combination (A) (Section 4.4). The reason for choosing this combination is that the quantified relative returns of the markets included in this combination (GSPC, FTSE, FCHI and GDAXI) yielded the best accuracy for predicting the trading signals of the AORD (Section 7.6 in Chapter 7) and the second best accuracy for predicting the direction of the Close price of the AORD (Section 5.4 in Chapter 5).

Table 4.11 gives the quantification coefficients relating to the indices included in the market combination (A) with their lower and upper bounds for different windows. These lower and upper bounds are denoted by $\xi_i^{\min}$ and $\xi_i^{\max}$, respectively.

Table 4.11: The lower ($\xi_i^{\min}$) and the upper ($\xi_i^{\max}$) bounds of the quantification coefficients corresponding to the market combination (A)

Training Window No.	Market Index	Quantification coefficient	Lower bound ($\xi_i^{\min}$)	Upper bound ($\xi_i^{\max}$)	Range
Window No.	Market Index	Quantification coefficient	Lower bound	Upper bound	Range
1	GSPC	0.57	0.50	0.62	0.12
	FTSE	0.29	0.26	0.35	0.09
	FCHI	0.11	0.00	0.12	0.12
	GDAXI	0.02	0.00	0.05	0.05
2	GSPC	0.61	0.54	0.66	0.12
	FTSE	0.18	0.16	0.22	0.06
	FCHI	0.08	0.07	0.13	0.06
	GDAXI	0.13	0.12	0.23	0.11
3	GSPC	0.77	0.53	0.91	0.38
	FTSE	0.09	0.04	0.18	0.14
	FCHI	0.13	0.03	0.27	0.24
	GDAXI	0.01	0.01	0.06	0.05
4	GSPC	0.79	0.70	0.99	0.29
	FTSE	0.06	0.01	0.08	0.07
	FCHI	0.15	0.01	0.17	0.16
	GDAXI	0.00	0.00	0.02	0.02
5	GSPC	0.56	0.46	0.68	0.22
	FTSE	0.17	0.03	0.22	0.19
	FCHI	0.03	0.00	0.11	0.11

Continued on next page

Table4.11 – continued from the previous page					
Training Window No.	Market Index	Quantification coefficient	Lower bound ($\xi_i^{\min}$)	Upper bound ($\xi_i^{\max}$)	Range
	GDAXI	0.24	0.18	0.33	0.15
6	GSPC	0.67	0.63	0.73	0.10
	FTSE	0.06	0.05	0.08	0.03
	FCHI	0.08	0.05	0.14	0.09
	GDAXI	0.20	0.18	0.23	0.05

In each window the lower boundary of the quantification coefficient relevant to the GSPC is greater than the upper boundary of the quantification coefficient relevant to any other market in the combination (A) (Table 4.11). The lowest lower boundary of the quantification coefficient associated with the GSPC (given that the corresponding median correlation does not lie below 95% of its optimal value) is around 0.5 which shows a high contribution.

The quantification coefficients relating to the other markets, which provides the median correlation not less than 95% of its optimal value, vary within narrow intervals being mainly close to zero (Table 4.11). This indicates that during the study period, the other markets showed a lesser impact on the Close price of day t of the AORD, than the GSPC. Finally, it can be suggested that the optimal coefficients are not very flexible in terms of providing the highest correlation.

4.8 Conclusions Derived from the Experiments

The optimal median Spearman's correlations obtained at different time lags suggested that only the combined influence from the Close prices of day $(t-1)$ of the different market combinations considered, on the Close price of day t of AORD, was significant during the study period.

Quantification results for different time periods confirm that influential patterns vary with time. The Close price of day $(t-1)$ of the GSPC showed the highest contribution to the combined influences on the Close price of day t of the AORD, followed by those of the three European markets considered. The Close prices of day $(t-1)$ of the considered Asian markets did not show any substantial influence. Surprisingly, the Close price of day $(t-1)$ of the AORD itself did not contribute as much as those of the European markets.

4.9 Summary

This chapter proposed a technique to quantify intermarket influences from a given combination of potential influential markets on a dependent market. This was done by assigning a coefficient (ξ) to each influential market included in the market combination. The optimal value of the each coefficient was derived in such a way that they maximise the median Spearman's correlation between the relative return of the Close price of day t of the dependent market and the sum of ξ_i multiplied by the lagged relative return of the ith influential market in the combination. This coefficient is termed quantification coefficient and measures the contribution from the respective influential market to the combined influence from the set of markets included in the combination.

This proposed technique was applied to find the possible influence from a selected set of global stock indices on the AORD. The results obtained suggest that the proposed quantification technique is successful. However, more experiments need to be carried out in order to justify this claim. One such attempt is to investigate if the quantified intermarket can effectively be used for prediction.

The next three chapters investigate whether the quantified intermarket influences on the AORD can effectively be used to predict the direction (up or down; Chapter 5) and the trading signals (Chapter 6 and Chapter 7) of the AORD.

Chapter 5

Predicting the Direction of the Australian All Ordinary Index

5.1 Introduction

A number of previous studies have attempted to predict the price levels of stock market indices (for example [16, 23, 67, 75]). However, in the last few decades, there has been a growing number of studies attempting to predict the direction or the trend movements of financial market indices (Section 2.2.1). Directional prediction is useful for traders as well as policy makers. Some studies have suggested that trading strategies guided by forecasts on the direction of price change may be more effective and may lead to higher profits [98].

Many previous studies (Section 2.5) have used technical indicators of the local markets or economical variables to predict the stock market time series. Only a few studies (for example [26, 65, 66]) incorporated the lagged data of foreign stock markets to predict the direction of a selected stock market, but there was no formal quantification of influence from those foreign markets.

The focus of this chapter is to investigate whether the quantified intermarket influences can be effectively used to predict the direction of the Close price of day (t+1) of the AORD. Measures of the strength of intermarket influences (that is, quantification coefficients) were incorporated for this prediction.

This chapter includes the identification of the techniques used to predict the direction of stock market indices, description about neural network training followed by the evaluation measures. It also explains how the prediction experiments were carried out and then presents the results of the numerical experiments with their interpretations. Finally it suggests the conclusions derived from the results obtained.

5.2 Techniques Used for Predicting the Direction of the Stock Market Indices

Chenowethet et al. [13] suggested that linear approaches such as linear time series models, are not capable of identifying dynamic or non-linear relationships in financial data. Neural networks adopt non-linear and non-parametric approach for modelling data. During the past few decades there has been growing interest in applications of artificial neural networks for predicting stock returns. Many studies have reported promising results [81]. It was found that the FNN outperforms the conventional prediction tools, such as multiple linear regression models and autoregressive integrated moving average models, in terms of directional prediction or prediction of percentage change in price level (Section 2.2.1).

To a lesser extent, backpropagation (feedforward) neural networks have also been used to predict the direction of different stock market indices [64, 104]. Alternatively, instead of predicting the direction, some researchers [23, 65, 68, 95, 100] predicted the price level using backpropagation neural networks, but the prediction accuracy was evaluated by the sign correctness (negative or positive) of the prediction (or hit rate) (Section 2.2.1).

Following past research, this study also adopted the FNN for predicting the direction of the Close price of the AORD. These networks predicted the price level of the Close price of the AORD and an approach similar to the approache used by Pan et al. [65] is used for evaluation.

5.2.1 Neural Network Training

In our past studies [65, 82], we obtained satisfactory results by applying a three-layered FNN for predicting the direction of the AORD. Following those studies, three-layered FNNs with one hidden layer were trained to predict the relative return of the Close price of the AORD. These FNNs were designed with the help of Matlab neural network toolbox [14]. The layers' weights and biases were initialised using the Nguyen-Widrow function [61].

In FNN applications for stock market predictions (Section 2.3.1), sigmoid functions are commonly used as transfer functions (for instance [65, 95, 100]). Theses functions are continuous, monotonically increasing, invertible, continuously differentiable, and bounded. According to Mehrotra et al. [52], these are the main reasons for selecting sigmoid functions. Furthermore, Kaastra and Boyd [33] argued if a network is to learn average behaviour, a sigmoid transfer function is suitable. According to the objectives of this research, it is expected that the networks will learn average behaviour. We also expected to transfer original relative returns to values between [-1, +1] and interested in non-linear modelling of data. Therefore, we employed a tan-sigmoid function, as the transfer function between the input layer and the hidden layer.

A linear transformation function was used as the transfer function between the hidden and the output layers. We assumed that a linear transformation function is sufficient as non-linear patterns were already identified (by the hidden layer) and also such an output tallies with our evaluation measures (Section 5.2.2).

The slope of a sigmoid function approaches zero as the input gets large and therefore the gradient can have a very small magnitude. If the steepest descent algorithm is used, this causes small changes in the weights and biases, even though the weights and biases are far from their optimal values [14]. The Resilient backpropagation training algorithm (Rprop) [71] eliminates the harmful effects of the magnitudes of the partial derivatives. It uses the sign of the derivative to determine the direction of the weight update; the magnitude of the derivative has no effect on the weight update. Therefore, the networks were trained with the resilient backpropagation training algorithm.

Stock market indices exhibit evolutionary characteristics which change over time [70]. This means that any model fitted will also have to be evolutionary or its usefulness will be short lived [9]. To address this issue, it was decided to use a number of moving windows for network training, rather than considering the whole study period as a single window. The same six moving windows that used to quantify intermarket influences on the AORD (Section 4.4) were considered for this purpose.

The most recent 10% of data of each moving window (76 samples) was used for testing while the remaining data (692 samples) was used for training. 692 samples are sufficient for learning the patterns within data while 76 samples are sufficient for evaluate the learning.

The most recent 22.2% of the training data (that is 20% of the window) was used for the validation. The majority of past studies with similar aim (directional prediction) did not use a validation set (Section 2.3.1). However, in this study, validation sets were used to monitor training progress so as to prevent the network from over-fitting.

The Nguyen-Widrow function uses different initial values for network parameters (weights and biases) [61]. This results in different solutions for network parameters. Since the network parameters vary according to their initial values, the network output also varies [9]. The general practice to overcome this problem is to train neural networks for a number of times and calculate the average output.

Input Sets

The networks were trained with three types of inputs:

1. **Type 1** - Multiple un-quantified inputs: $X_j(t-i)$, $j = 1, 2, ..., m$, where $X_j(t-i)$ is the relative return at time lag i of the jth influential market and m is the number of influential markets;

2. **Type 2** - Sum of the quantified inputs: $\sum_{j=1}^{m} \xi_j X_j(t-i)$ where $X_j(t-i)$ is defined as in Type 1 above while ξ_j is the quantification coefficient (Section 4.2) associated with the j influential market;

3. **Type 3** - Multiple quantified inputs: $\xi_j X_j(t-i)$, $j = 1, 2, ..., m$, where $X_j(t-i)$ and ξ_j are defined as 1 and 2 above.

Since the aim is to predict the direction (up or down) of day $(t+1)$ of the AORD, the value of i can be varied from 0 to any positive integer. However, it is not meaningful to consider i >5. This is because it is very unlikely that the Close prices of either the influential markets or the AORD, of more than one week in the past, have a significant impact on the Close price of day $(t+1)$ of the AORD. Therefore, in the experiments $i = 0, 1, ..., 5$ were considered. i=0 indicates the relative returns of day t, i=1 indicates the relative return of day $(t-1)$ and so on.

Each set of influential markets included one or more of the following stock markets which are assumed to be potential influential markets to the AORD:

1. US S&P 500 Index (GSPC),
2. US Nasdaq Composite Index (IXIC),
3. UK FTSE 100 Index (FTSE),
4. French CAC 40 Index (FCHI),
5. German DAX Index (GDAXI),
6. Hong Kong Hang Seng Index (HSI),
7. Singapore Straits Times Index(STI),
8. Japanese Nikkei 225 Index (N225),
9. Chinese SSE Composite Index (SSEC),

10. Taiwan TSEC Weighted Index (TWII).

The number of neurons in the input layer was varied according to the number of input variables. The output layer consisted of a single neuron which generates the predicted value of the relative return of the Close price of day t of the AORD.

Different numbers of neurons for the hidden layer were tested along with various values for learning rate and the momentum (Section 3.2.1).

5.2.2 Evaluation Measures

To evaluate the prediction accuracy, the rate of return and hit rate are usually employed in the literature (Section 2.2.1). The rate of return assesses the profitability of predictions while hit rate assesses their predictability. When applying the rate of return as a measure of evaluation of directional prediction, past studies assumed that an upward trend (or positive prediction) as a buy signal and downward trend (or negative prediction) as a sell signal. Furthermore, these studies did not consider the hold signal (Section 2.2.1).

In practice, a trader does not buy or sell if there is no significant change in the price level; instead he/she holds the money or shares in hand. Therefore, it is not practicable to use the rate of return as the evaluation measure.

Hit rate indicates the percentage of correct predictions (the percentage of prediction with correct direction, in the case of predicting the direction). However, if the relative return of day $(t+1)$ is zero or approximately zero, then there is no substantial difference between Close price of day t and that of day $(t+1)$, irrespective of the sign. Hit rate does not take this matter into account.

To fix this problem, Pan et al. [65] used a different measure which considers a threshold which helps to represent a ‘no change’ region. When the sign of the actual and the predicted values are different, they checked whether the absolute value of the difference between the actual and the predicted values is less than this threshold. If so they considered that the signs of the both values are the same (Section 2.2.1).

This study adopted a similar approach to Pan et al. [65]. It uses *Direction Correctness Percentage* (DCP) to asses the performance of the networks. DCP indicates the percentage of predictions with the correct direction (up, down or no change).

DCP can be described as follows:

> If the relative return of a particular day is positive that means there is an increase in this day's Close price compared to the previous day and vice-versa. If both the actual and the predicted relative return are positive or both negative, it is obvious that the direction of both values are the same. Even though the sign is different, if the absolute value of both actual and predicted relative returns lie close to zero, then it is reasonable to assume that there is no significant change in the next day's Close price. This study adopts 0.001 as the threshold. In other words, if the observed and the predicted relative returns of the AORD lie in the range (-0.001, 0.001), then it was considered that the direction of both are the same regardless of whether they had the same sign.

5.2.3 Description of Directional Prediction Experiments

The same six moving windows mentioned in Section 4.4 of the previous chapter (Chapter 4) was considered for neural network experiments. The most recent 10% of data of each window were allocated for testing while the remaining was used for training. The training sets used for these experiments are the same as the training windows used for quantification of intermarket influences (Section 4.4). Therefore, we were able to use the quantification coefficients derived in the previous experiments (Chapter 4) for the neural network experiments.

Initially, the neural networks were trained with the relative returns of different individual markets. Since, it is not meaningful to consider quantified relative return of individual stock markets, only the first type of inputs (Section 5.2.1) were considered. Depending on the results obtained from the inputs sets, which included relative returns of the individual markets, relative returns of the different market combinations were selected as inputs. All three types of inputs described in Section 5.2.1 were considered in this stage. Network training with each input set was repeated for all six training windows.

The DCP (Section 5.2.2) was calculated for the test set of a window for each trial of neural network training. Since the output of the networks varies according to the initial values of the network parameters (Section 5.2.1), each FNN was trained 500 times (trials). Then the average DCP over 500 trails was calculated. This procedure was repeated for all six windows. Finally, the overall average DCP over six windows was calculated.

5.3 Numerical Experiments and Interpretations

FNNs were trained with single inputs as well as sets of several inputs. Three neurons in the hidden layer with a learning rate equal to 0.003 and a momentum coefficient equal to 0.01 always gave better performance for validation and testing.

Note: In this section Type 1 to Type 3 inputs are also referred as *multiple un-quantified inputs*, *multiple quantified inputs*, and *sum of the quantified inputs*, respectively (Section 5.2.1).

5.3.1 Prediction Based on Single Markets

Firstly, an investigation was carried out to understand the contribution from the relative returns of the potential influential markets as well as those of the AORD, at different time lags, for predicting the direction of the Close price of day $(t + 1)$ of the AORD. Table 5.1 shows the average DCP for testing of the neural network performance. The relative returns at different time lags of AORD and the considered potential influential markets were used as the input features.

Except for the US markets, the average DCP decreased when the relative return of day $(t-1)$ (that is, $X(t\text{-}1)$) was added to the input set consists of relative return of day t, $X(t)$ (Table 5.1). This indicates that it is not worth adding the relative returns of one or more days in the past of the European as well as those of the Asian markets to the input set. Adding these extra inputs seems to have created noise that reduces the predictive ability of the networks.

At time lag one, the GSPC had the highest DCP (67.33%). The next five highest DCPs were produced by the GDAXI, FTSE, FCHI, IXIC and HSI, respectively. These results indicate that the relative returns of day t of the US and the European markets are more informative for predicting the relative return of day $(t+1)$ of the AORD, than those of the other markets. Among the Asian markets considered, the HSI seemed to be the most useful market for the prediction of the AORD.

Table 5.1: Average Direction Correctness Percentage (DCP) for testing for the $(t+1)$th day's prediction based on single market

Stock market	Average DCP relating to	
	the relative return of day t $(X(t))$	the relative returns of day t and day $(t-1)$ $(X(t)$ & $X(t\text{-}1))$
GSPC	67.3254	67.4138
IXIC	63.3281	63.7049
FTSE	65.4715	64.6191
FCHI	64.1482	63.5342
GDAXI	65.6118	64.8196
SSEC	59.8412	57.412
HSI	61.0118	59.152
TWII	59.9579	58.3218
STI	55.2553	53.5253
N225	56.6947	53.5764
AORD	59.0978	57.6956

The average DCP corresponding to the US market increased when the relative returns at time lag two were added. Therefore, the relative returns of higher lags (>2), of these markets were also considered. Table 5.2 presents the average DCP relevant to the US markets when the relative returns at time lags greater than two were added to the input set.

Table 5.2: Average Direction Correctness Percentage (DCP) for testing for the $(t+1)$th day's prediction based on the US markets ($X(t)$, $X(t$-1), ... refer to the relative returns of day t and day $(t-1)$ and so on.)

Inputs	Average DCP for	
	GSPC	IXIC
$X(t), X(t\text{-}1), X(t\text{-}2)$	67.9644	64.3432
$X(t), X(t\text{-}1), X(t\text{-}2), X(t\text{-}3)$	68.0114	64.1237
$X(t), X(t\text{-}1), X(t\text{-}2), X(t\text{-}3), X(t\text{-}4)$	68.1521	64.7155
$X(t), X(t\text{-}1), X(t\text{-}2), X(t\text{-}3), X(t\text{-}4), X(t\text{-}5)$	68.4000	64.7530

Table 5.2 shows that the average DCP value increased as the relative returns at the time lags greater than two of the GSPC were added to the input set. When the relative returns at time lag three was added, the average DCP relevant to the IXIC decreased. However, it increased when those at time lags greater than three were added to the input set.

5.3.2 Prediction Based on Different Influential Market Combinations (Using Quantification Coefficients)

When the relative returns of day t of single markets were considered as inputs, the GSPC gave the highest DCP (Table 5.1). Therefore, the neural networks were trained with the relative return of day t of the GSPC together with those of different sets of markets as the inputs. These sets included the markets (GDAXI, FTSE, FCHI, IXIC and HSI) which gave the next highest DCPs after the GSPC. The aim was to find the input combination(s) which increases the DCP. The three types of inputs described in Section 5.2.1, were considered.

The quantification coefficients (at time lag 1) corresponding to the market combinations that were considered in this section are presented in Appendix A.

Network Training with Type 1 and Type 2 Inputs

Table 5.3 compares the average DCP for the un-quantified input combinations of influential markets (Type 1) with their single quantified counterparts (Type 2).

Table 5.3: Average Direction Correctness Percentage (DCP) for the test set when networks were trained with Type 1 and Type 2 inputs (Bold values represent the higher DCPs for the multiple un-quantified inputs and the sum of the quantified inputs while $X(t)$ refers to the relative return of day t of market X).

Input variables	Average DCP for		% increase in average DCP when using sum of the quantified inputs for prediction
	multiple un-quantified inputs (Type 1)	sum of the quantified inputs (Type 2)	
$\text{GSPC}(t), \text{GDAXI}(t)$	**67.6158**	**68.6798**	1.57
$\text{GSPC}(t), \text{FTSE}(t)$	67.6136	68.5070	1.32
$\text{GSPC}(t), \text{FCHI}(t)$	67.5667	68.3162	1.11
$\text{GSPC}(t), \text{IXIC}(t)$	65.1456	67.3706	3.42
$\text{GSPC}(t), \text{HSI}(t)$	67.0202	67.2627	0.36
$\text{GSPC}(t), \text{FTSE}(t), \text{GDAXI}(t)$	67.5066	68.8443	1.98

Continued on next page

Table5.3 – continued from the previous page			
Table5.3 – continued from the previous page			
Input variables	Average DCP for multiple un-quantified inputs (Type 1)	Average DCP for sum of the quantified inputs (Type 2)	% increase in average DCP when using sum of the quantified inputs for prediction
GSPC(t), FCHI(t), GDAXI(t)	67.1561	68.7474	2.37
GSPC(t), FTSE(t), FCHI(t)	67.3088	**68.8689**	2.32
GSPC(t), FTSE(t), IXIC(t)	66.3711	68.4237	3.09
GSPC(t), GDAXI(t), HSI(t)	67.3829	68.4118	1.53
GSPC(t), FTSE(t), HSI(t)	67.5311	68.1224	0.88
GSPC(t), GDAXI(t), IXIC(t)	66.837	68.4395	2.40
GSPC(t), FTSE(t), FCHI(t), IXIC(t)	66.0388	66.8469	1.22
GSPC(t), FTSE(t), FCHI(t), GDAXI(t)	67.3338	**69.1136**	2.64
GSPC(t), FTSE(t), FCHI(t), GDAXI(t), HSI(t)	66.8570	66.8508	0.42
GSPC(t), FTSE(t), FCHI(t), GDAXI(t), IXIC(t)	65.6560	66.2974	0.98
GSPC(t), FTSE(t), FCHI(t), GDAXI(t), HSI(t)	66.8570	68.6434	2.67

Initially all possible two market combinations were used to form the inputs sets. Among these input sets, the multiple un-quantified input set (Type 1) with the GSPC and the GDAXI gave the highest average DCP (Table 5.3). This is also true for the single inputs which represent the sum of quantified inputs (Type 2).

In the next step, all possible three market combinations which include the GSPC and the GDAXI were considered when forming the input sets. Additionally, all possible three market combinations with the GSPC and the FTSE were taken into account. The reason is that the GSPC and the FTSE showed the first and second highest contribution the to the combined influence on the AORD, respectively (Section 4.6). Among the single inputs with three market combinations, the one corresponding to the GSPC, FTSE and FCHI yielded the highest average DCP and this value was grater than the best average DCP of two market combinations.

Then all possible four market combinations with the GSPC, FTSE, and FCHI were considered. The average DCP was further improved when the GDAXI was added to the market combination with the GSPC, FTSE and FCHI.

Adding a fifth market to the market combination with the GSPC, FTSE, FCHI and the GDAXI, did not improve the average DCP. Therefore, it is not reasonable to train the network with input sets which include all six markets.

It is noteworthy that the DCP corresponding to the input set of Type 2 was always higher than that of the input set of Type 1. This may be due to the separate inputs containing more noise than their quantified counterparts and consequently, the quantified inputs show high correlation with the output (relative return of day $(t+1)$ of the AORD).

Network Training with Type 1 and Type 3 Inputs

Table 5.4 compares the average DCP for the un-quantified input combinations of influential markets (Type 1) with the respective multiple quantified inputs (Type 3).

Table 5.4: Average Direction Correctness Percentage (DCP) for the test set when networks were trained with Type 1 and Type 3 inputs (Bold values represent the higher DCPs for multiple un-quantified and quantified inputs while $X(t)$ refers to the relative return of day t of market X).

Input variables	Average DCP for multiple un-quantified inputs (Type 1)	Average DCP for multiple quantified inputs (Type 3)	% increase in average DCP when using quantified inputs for prediction
GSPC(t), GDAXI(t)	**67.6158**	67.4899	-0.19
GSPC(t), FTSE(t)	67.6136	**68.1377**	0.76
GSPC(t), FCHI(t)	67.5667	67.4939	-0.11
GSPC(t), IXIC(t)	65.1456	64.2531	-1.37
GSPC(t), HSI(t)	67.0202	67.3566	0.50
GSPC(t), FTSE(t), GDAXI(t)	67.5066	67.5886	0.12
GSPC(t), FCHI(t), GDAXI(t)	67.1561	67.1022	-0.08
GSPC(t), FTSE(t), FCHI(t)	67.3088	67.6373	0.49
GSPC(t), FTSE(t), IXIC(t)	66.3711	66.8399	0.71
GSPC(t), GDAXI(t), HSI(t)	67.3829	67.7360	0.52
GSPC(t), FTSE(t), HSI(t)	67.5311	**68.4443**	1.35
GSPC(t), GDAXI(t), IXIC(t)	66.837	65.2746	-2.34
GSPC(t), FTSE(t), FCHI(t), HSI(t)	66.5684	67.7298	1.74
GSPC(t), FTSE(t), GDAXI(t), HSI(t)	66.8802	**68.0316**	1.72
GSPC(t), FTSE(t), HSI(t), IXIC(t)	66.6933	66.6031	-0.14
GSPC(t), FTSE(t), FCHI(t), GDAXI(t), HSI(t)	66.8570	67.6978	1.26

Firstly, the all possible two market combinations were considered for network training. The market combination of the GSPC and the FTSE gave the highest average DCP among the multiple quantified input combinations (Table 5.4).

In the next step, the all possible three market combinations with the GSPC and the GDAXI as well as those combinations with the GSPC and FTSE were considered when forming the input sets. When a third market was added to any multiple un-quantified input set, which is a combination of two markets, the average DCP dropped. The average DCP increased when the HSI was added to the multiple quantified input set which consists of the GSPC and the FTSE.

Then all possible four market combinations which include the GSPC, FTSE and the HSI were used to form the input sets. The average DCP corresponding to the multiple un-quantified input sets was further reduced. The addition of the forth market to the multiple quantified input combinations with the GSPC, FTSE and HSI also did not contribute to any improvement in average DCP. Among the multiple quantified inputs sets with four market combinations, the one consisting of the GSPC, FTSE, GDAXI and HSI showed the highest average DCP. The addition of the FCHI to the multiple quantified input combination of the GSPC, FTSE, GDAXI and HSI reduced the average DCP. It is unlikely that the considering input sets which includes all six markets will produce better results.

The highest average DCP for the multiple un-quantified input sets was obtained when the relative returns of day t of the GSPC and the GDAXI were included in the market combination (Table 5.4). In general DCP decreased when more markets were added.

For the majority of the market combinations considered, the multiple quantified inputs gave higher average DCP than their un-quantified counterparts. However, there are some combinations for which the multiple un-quantified inputs generated higher average DCP than their quantified counterparts. The reason may be the inability of the networks (used in this chapter) to provide deep (global) solutions.

When considering the multiple quantified inputs, the one corresponding to the market combination the GSPC, FTSE and HSI produced the highest average DCP. The second highest average DCP was obtained when the networks were trained with the multiple quantified inputs of the market combination of the GSPC and the FTSE, followed by that of the market combination of the GSPC, FTSE, GDAXI and HSI.

Comparison of the Results Corresponding to Three Types of Inputs

The single inputs, which represent the sum of quantified inputs, always gave higher DCP values for the relative return of day $(t+1)$ of AORD than their un-quantified counterparts (Table 5.3). Except for a few cases, the average DCP produced by these single inputs (Type 2), is higher than that produced by the respective multiple quantified inputs (Table 5.3 and 5.4). As mentioned above, the neural networks may not be capable enough to find deep solutions when trained with multiple inputs. (To deal with this problem, this study developed new neural network algorithms described in Chapter 7.)

When considering the single input combinations, the one corresponding to the relative returns of day t of the GSPC together with those of the European markets, resulted in the highest DCP (69.1136). This value is even higher than the highest average DCP produced by any of the multiple quantified inputs (68.4443; Table 5.3 and 5.4). The next highest DCP was generated by the single input of the sum of the quantified relative return of day t of the GSPC, FTSE and the FCHI followed by that corresponding to the relative return of day t of the GSPC and the GDAXI. However, when a fifth market was added to the single input, which gave the highest prediction accuracy, the DCP decreased. This indicates that either the IXIC or the HSI did not contribute significantly the directional prediction of the Close price of day $(t+1)$ of the AORD and indeed they reduced the predictive power of the model. Therefore, it can be suggested that the sum of the quantified relative return of day t of the US GSPC market and the European markets was the most appropriate input to predict the direction of the Close price of the AORD.

Among the multiple un-quantified input combinations, the relative returns of day t of the GSPC and the GDAXI gave the best DCP (67.6158; Table 5.3). As with the earlier case (Table 5.4), DCP decreased when more markets were added.

According to the input combinations already tested, the sum of the quantified relative returns of day t of the GSPC and the European markets yielded the best prediction for the direction of the relative return of the Close price of day $(t+1)$ of the AORD. The possibilities of improving the predictive power of this model are worth investigating. One possibility is adding extra markets to this market combination.

5.3.3 Prediction Based on Different Influential Market Combinations and the AORD

Past studies [65, 82] revealed that the lagged time series data of the AORD itself is useful for predicting the direction of the Close price of the AORD. Furthermore, Table 5.1 shows that the relative return of day t of the AORD alone has an ability to predict its direction corresponding to day $(t+1)$, with an accuracy of 59%. This indicates that the Close price of day (t) of the AORD had an slight impact on the following day's Close price of the AORD itself. Therefore, in order to investigate the possibilities of improving the DCP, the relative return of day t of the AORD was added to the best input set with multiple un-quantified relative returns as well as the best three single inputs (Table 5.3) and the best three multiple quantified input sets (Table 5.4).

The results obtained by the neural network training when the relative return of day t of the AORD was added to the best input sets shown in Table 5.3 are presented in Table 5.5. The quantification coefficients relevant to the market combinations shown in these two tables are presented in Appendix A.

When the relative return of day t of the AORD was added to the best three input sets, the DCP values relating to all multiple un-quantified sets decreased (Table 5.5). However, the DCPs corresponding to their counterparts with single inputs increased. The highest average DCP was produced by the single input set which includes the sum of the quantified

Table 5.5: Average Direction Correctness Percentage (DCP) for test set when the relative return of day t of the AORD was added to the best input sets in Table 5.3 (Bold value represents the highest average DCP).

Input variables	Average DCP for		% increase in average DCP when using sum of the quantified inputs for prediction
	multiple un-quantified inputs (Type 1)	sum of the quantified inputs (Type 2)	
$\mathrm{GSPC}(t), \mathrm{GDAXI}(t), \mathrm{AORD}(t)$	66.7136	68.5096	2.69
$\mathrm{GSPC}(t), \mathrm{FTSE}(t), \mathrm{FCHI}(t), \mathrm{AORD}(t)$	66.1996	69.0689	4.33
$\mathrm{GSPC}(t), \mathrm{FTSE}(t), \mathrm{FCHI}(t), \mathrm{GDAXI}(t), \mathrm{AORD}(t)$	66.2175	**69.1605**	4.44

relative returns of day t of the GSPC, FTSE, FCHI, GDAXI and AORD. This value is the highest average DCP among the all input sets considered in this study. However, addition of the AORD to the input set that included the GSPC and the three European markets increased the prediction accuracy only by 0.07%.

Table 5.6 presents the results of the neural network training when the relative return of day t of the AORD was added to the best input sets shown in Table 5.4.

In the case of quantified inputs, except for the market combination which includes the GSPC, FTSE, GDAXI, and HSI, adding the AORD to all other market combinations, helped to improve the average DCP (Table 5.6). The highest average DCP was obtained when the multiple quantified inputs include the GSPC, FTSE, HSI and AORD (68.5382%). However, when considering the multiple un-quantified inputs, the addition of the relative return of day t of the AORD did not improve the average DCP.

Table 5.6: Average Direction Correctness Percentage (DCP) for test set when the relative return of day t of the AORD was added to the best input sets shown in Table 5.4 (Bold value represents the highest average DCP).

Input variables	Average DCP for		% increase in
	multiple un-quantified inputs (Type 1)	multiple quantified inputs (Type 3)	average DCP when using quantified inputs for prediction for prediction
$\mathrm{GSPC}(t), \mathrm{FTSE}(t), \mathrm{AORD}(t)$	66.2658	68.3101	3.08
$\mathrm{GSPC}(t), \mathrm{FTSE}(t), \mathrm{HSI}(t), \mathrm{AORD}(t)$	66.3806	**68.5382**	3.25
$\mathrm{GSPC}(t), \mathrm{FTSE}(t), \mathrm{GDAXI}(t), \mathrm{HSI}(t), \mathrm{AORD}(t)$	66.5847	67.6702	1.63

5.3.4 Investigating the Possibilities of Including Other Markets to the Best Input Combinations

Table 5.1 and Table 5.2 demonstrate that the relative returns of both the GSPC and the IXIC at time lags greater than 1 help improve the DCP. Therefore, the sum of the quantified relative returns of day t of the GSPC, European markets and AORD, and the relative returns of day $(t-1)$ of the GSPC was considered as a single input. This resulted in a decrease of the DCP value (DCP = 68.4898), suggesting that it is not worth adding the relative returns of the GSPC at higher lags.

Adding the relative return of day t of the IXIC to the sum of the quantified relative returns of the GSPC and the European markets also did not improve the DCP value (Table 5.3). Hence, it is not helpful to add the relative returns of the IXIC at the time lags greater than 1 to the input combination which gave the highest DCP. Finally, it can be proposed that the direction (up or down) of the Close price of day $(t+1)$ of the AORD can best be predicted using the sum of the quantified relative returns of the Close price of day t of the GSPC and the European markets (FTSE, GDAXI and FCHI) together with that of the AORD.

5.4 Conclusions Derived from the Experiments

Usage of quantified intermarket influences for predicting the direction (up or down) of the Close price of day $(t+1)$ of the AORD seems to be better than their un-quantified counterparts. We used two different ways to use the quantified intermarket influences: (1) considering the sum of the quantified intermarket influences of the different stock market indices as one 'combined input' (Type 2); and, (2) considering the quantified intermarket influences from different stock market indices as separate inputs (Type 3). Out of theses two, the first option proved to be more productive for the directional prediction of the AORD Close price. The best results were obtained when the combined influence from the Close price (relative return) of day t of the US S&P 500 Index (GSPC) and those of the European stock market indices (the FTSE, FCHI and GDAXI) together with that of the AORD itself, was used as the input to for the prediction of interest. However, the addition of the relative return of the Close price of day t of the AORD to the input set, with corresponding relative returns of the GSPC and the European markets, improved the prediction accuracy only by 0.07%.

5.5 Summary

This chapter investigated whether the quantified intermarket influences on the AORD can be effectively used to predict the direction (up or down) of its Close price. Also, it examined how the quantified intermarket influences can be used to obtain better prediction accuracy. Results suggested that the quantified intermarket influences can effectively be used to predict the direction of the Close price of day $(t+1)$ of the AORD. This indicates the success of the proposed technique for quantifying intermarket influences.

Some studies consider the direction, that is upward trend and downward trend, as corresponding to buy and sell signals. However, in practice, a trader will not buy or sell if there is no significant increase or decrease in the price level of a stock market index and instead, he/she will hold the money or the shares in hand. Therefore, it is more useful to predict whether it is best to buy, hold or sell shares of a stock market index (in other words predict the trading signals), rather than the directional prediction. Therefore, the next chapter (Chapter 6) focuses on predicting the trading signals of the AORD.

Chapter 6

Predicting Trading Signals of the Australian All Ordinary Index

6.1 Introduction

Majority of the past studies focused on classification of future values into two categories (up or down) which are considered to be buy and sell signals (Section 2.2.2). Timely decisions must be made which result in buy signals when the market is low and sell signals when the market is high [9]. However, it is worth holding shares if there is no significant rise or drop in the price index. Therefore, from the practical point of view, it is important to consider the 'hold' category.

Many studies found evidence for the existence of intermarket influences among global stock markets (Section 2.4). However, the use of intermarket influences from foreign stock market indices to predict trading signals of a given market is very rare in the literature (Section 2.5). The few studies which used intermarket influences did not attempt to quantify the intermarket influences.

This chapter focuses on predicting whether it is best to buy, hold or sell; in other words predicting the trading signals of day $(t+1)$ of the AORD. Several types of input sets were used in forecasting in order to identify the best way of using the available information of the foreign stock market indices to forecast the trading signals. These input sets include the intermarket influences from the major influential markets to the AORD in quantified form as well as un-quantified form.

Prediction of trading signals was done by applying three different algorithms. The prediction results were evaluated in terms of predictability as well as profitability.

6.2 Defining Trading Signals

As mentioned earlier most of the past studies classified the future values into buy or sell signals based on the direction of the trend (down or up) of the future values (section 2.2.2). Since, this study considers three classes, the following criterion was introduced to identify the trading signals.

Criterion A

$$\begin{aligned} &\textit{buy} \text{ if } && Y(t+1) \geq l_u \\ &\textit{hold} \text{ if } && l_l < Y(t+1) < l_u \\ &\textit{sell} \text{ if } && Y(t+1) \leq l_l \end{aligned}$$

where $Y(t+1)$ is the relative return the Close price of day $(t+1)$ of the AORD while l_u and l_l are two thresholds.

The values of of l_u and l_l depend on traders' choice. There is no standard criterion found in the literature on how to decide the values of l_u and l_l and theses values may vary from one stock index to another. A trader may decide the values for these thresholds according to his/her knowledge and experience.

The proper selection of the values for l_l and l_u could be done by performing a numerical experiments. We experimented different pairs of values for l_l and l_u. For different windows, different pairs gave better predictions. These values also varied according to the prediction algorithm used. However, for the definition of trading signals, these values needed to be set.

For this study we chose $l_u = -l_l = 0.005$, assuming that 0.5% increase (or decrease) in Close price of day $(t+1)$ compared to that of day t, is reasonable enough to consider the corresponding movement as a buy (or sell) signal.

6.3 Algorithms Used for Predicting Trading Signals

The three algorithms (FNN, PNN, and SVM) which were claimed by past studies (Section 2.3) as the most successful algorithms for predicting trading signals were adopted by this study. Both the PNN and the SVM output the predicted class while the FNN gives the value of the prediction instead of the class. The theory behind these three algorithms are discussed in detail in Chapter 3 (Section 3.2.1 to 3.2.3).

6.4 Evaluation Measures

The majority of past studies evaluated their predictions in terms of profitability while a few studies concerned on the predictability (Section 2.2.2). In these studies, the profitability of predictions was determined according to the rate of returns obtained by performing different trading strategies. The rate of return is a measure that provides the net gain in assets as a percentage of the initial investment.

The most common measure of evaluation of the predictability is the hit rate (Section 2.2.2). The hit rate indicates the percentage of cases correctly classified.

This study aimed at classifying trading signals into three classes: buy, hold and sell. From a trader's point of view, the misclassification of a hold signal as a buy or sell signal is a more serious mistake than misclassifying a buy signal or a sell signal as a hold signal. The reason is in the former case a trader will loses the money by taking part in an unwise investment while in the later case he/she only loses the opportunity of making a profit, but makes no monetary loss. The most serious monetary loss occurs when a buy signal is misclassified as a sell signal and vice-versa. The hit rate does not take the seriousness of the misclassification into account, and therefore, it is not an adequate measure for the evaluation of the forecasting relating to this study.

This study used the classification and misclassification rates to evaluate the forecasting accuracy. These rates indicate the patterns of classification/misclassification of data belonging to a class. The classification rate indicates the proportion of correctly classified signals to a particular class out of the total number of actual signals in that class whereas, misclassification rate indicates the proportion of incorrectly classified signals from a particular class to another class, out of the total number of actual signals in the former class.

The other measure this study used to asses validity the forecasting is the rate of return obtained by performing trading simulations. Predictability does not necessarily imply profitability. The results from trading are also useful in identifying better models when the predictive performances are not significantly different [81].

Different past studies employed different trading strategies to asses the profitability of the forecasts [81]. This study adopted a buy and sell strategy to form the trading simulation. Following Yao et al. [101] this study also assumed the major blue chips in the stock basket are bought or sold, and the aggregate price of the major blue chips is the same as the index.

For this study we proposed a new trading simulation. The speciality of the trading simulation is that it searches for the proportion of money that a trader needs to invest and the proportion of shares that he/she needs to sell, in order to earn higher profit. In this sense, the proposed simulation is very close to the reality.

6.4.1 Trading Simulations (Paper Trading)

This study assumes that at the beginning of each period, the trader has some amount of money as well as a number of shares. Furthermore, it is assumed that the value of money in hand and the value of shares in hand are equal. Two types of trading simulations were used: (1) response to the predicted trading signals which might be a buy, hold or a sell signal; (2) do not participate in trading, and hold the initial shares and the money in hand until the end of the period. The second simulation was used as a benchmark.

First Trading Simulation (The Proposed Trading Simulation)

Let the value of the initial money in hand be M^0 and the number of shares at the beginning of the period be S^0. Assume that $S^0 = M^0/P_0$, where P_0 is the Close price of the AORD on the day before the starting day of the trading period.

Also let M_t, S_t, P_t, VS_t be the money in hand, number of shares, Close price of the AORD, value of shares holding on the day t (t=1, 2, ..., T), respectively. This simulation assumes that a fixed amount of money is always used in trading regardless of whether the trading signal is buy or sell. Let this fixed amount be denoted as F^0 and be equal to M^0/L, $L > 0$. In the calculations $L = 1, 2, ..., 10$ is considered. When $L = 1$, F^0 equals M^0, when $L = 2$, F^0 equals 50% of M^0 and so on. Let Δ_t^b and Δ_t^s be the number of shares bought and the number of shares sold at day t, respectively.

Suppose the trading signal at the beginning of the day t is a buy signal. Then the trader spends $F = \min\{F^0, M_{t-1}\}$ amount of money to buy a number of shares at a rate of the Close price of day $(t-1)$.

$$M_t = M_{t-1} - F, \qquad F = \min\{F^0, M_{t-1}\} \tag{6.1}$$

$$\Delta_t^b = \frac{F}{P_{t-1}} \tag{6.2}$$

$$S_t = S_{t-1} + \Delta_t^b \tag{6.3}$$

$$VS_t = S_t \times P_t \tag{6.4}$$

Suppose the trading signal is a hold signal, then:

$$M_t = M_{t-1} \tag{6.5}$$

$$S_t = S_{t-1} \tag{6.6}$$

$$VS_t = S_t \times P_t \tag{6.7}$$

Let the trading signal at the beginning of the day t be a sell signal. Then the trader sells S'=min$\{(F^0/P_{t-1}), S_{t-1}\}$ amount of shares.

$$\Delta_t^s = S', \qquad S' = \min\{(F^0/P_{t-1}), S_{t-1}\} \tag{6.8}$$

$$M_t = M_{t-1} + S' \times P_{t-1} \tag{6.9}$$

$$S_t = S_{t-1} - \Delta_t^s \tag{6.10}$$

$$VS_t = S_t \times P_t \tag{6.11}$$

According to the above definitions, it is clear that if there is no money in hand (F=0), a buy signal will treated as a hold signal. Similarly, if there are no shares in hand, a sell signal will be treated as a hold signal.

Second Trading Simulation (The Benchmark Trading Simulation)

In this case, the trader does not participate in trading. Therefore, $M_t = M^0$ and $S_t = S^0$ for all t=1, 2, ..., T. However, the value of the shares changes with the time and therefore, the value of shares at day t, $VS_t = S^0 \times P_t$.

Rate of Return

Let the total value of money and shares in hand at the end of the period (day T) be TC. This value can be calculated as:

- for the first trading simulation

$$TC = M_T + S_T \times P_T \tag{6.12}$$

- for the second trading simulation

$$TC = M^0 + S^0 \times P_T \tag{6.13}$$

The rate of return ($R\%$) at the end of a trading period is calculated as below:

$$R\% = \frac{TC - 2M^0}{2M^0} \times 100 \tag{6.14}$$

6.5 Data

The results obtained from Chapter 5 (Section 5.4) showed that the relative returns of the Close prices of day t of the GSPC (US S&P 500 Index), the three European market indices; the FTSE (UK FTSE 100 Index), FCHI (French CAC 40 Index), GDAXI (German DAX Index) as well as the AORD are suitable to predict the direction of the Close price of day $(t + 1)$ of the AORD. Therefore, to forecast the trading signals, the data from two combinations of stock market indices were selected: (1) the GSPC, FTSE, FCHI and GDAXI; and, (2) the GSPC, FTSE, FCHI, GDAXI and AORD. The following six input sets were used for forecasting trading signals:

1. Four input features of the relative returns of the Close prices of day t of the market combination (1) (denoted by **GFFG**)
   - (GSPC(t), FTSE(t), FCHI(t), GDAXI(t));

2. Four input features of the quantified relative returns of the Close prices of day t of the market combination (1) (denoted by **GFFG-q**)
   - (ξ_1GSPC(t), ξ_2FTSE(t), ξ_3FCHI(t), ξ_4GDAXI(t));

3. Single input feature of the sum of the quantified relative returns of the Close prices of day t of the market combination (1) (denoted by **GFFG-sq**)
   - (ξ_1GSPC(t)+ξ_2FTSE(t)+ξ_3FCHI(t)+ξ_4GDAXI(t));

4. Five input features of the relative returns of the Close prices of day t of the market combination (2) (denoted by **GFFGA**)
   - (GSPC(t), FTSE(t), FCHI(t), GDAXI(t), AORD(t));

5. Five input features of the quantified relative returns of the Close prices of day t of the market combination (2) (denoted by **GFFGA-q**)
   - (ξ_1^AGSPC(t), ξ_2^AFTSE(t), ξ_3^AFCHI(t), ξ_4^AGDAXI(t), ξ_5AORD(t));

6. Single input feature of the sum of the quantified returns of the Close prices of day t of the market combination (2) (denoted by **GFFGA-sq**)
   - (ξ_1^AGSPC(t)+ξ_2^AFTSE(t)+ξ_3^AFCHI(t)+ξ_4^AGDAXI(t)+ξ_5^AAORD(t)).

(ξ_1, ξ_2, ξ_3, ξ_4) and ($\xi_1^A, \xi_2^A, \xi_3^A, \xi_4^A, \xi_5^A$) are the solutions to (4.9), (4.10) (in Section 4.2.1) corresponding to the market combination (1) and (2), respectively. We note that it may be $\xi_i \neq \xi_i^A$, for i=1, 2, 3, 4.

6.6 Numerical Experiments and Evaluations of Prediction Results

The same six moving windows which were used for quantifying intermarket influences on the AORD (refer Section 4.4 and 4.6) were considered for theses experiments. For all experiments, each window was divided into two sets: a test set and a training set. The most recent 10% of data was used for testing while the remaining 90% was used for training. Each window consisted of 768 samples and therefore, each training set consisted of 692 samples while each test set consisted of 76 samples. 692 samples are sufficient for learning the patterns within data while 76 samples are sufficient for evaluate the learning.

The training sets used for the experiments are the same as the training windows used for quantification of intermarket influences (Section 4.4) in Chapter 4. Therefore, it is appropriate to use the same sets of values for quantification coefficients which were derived previously in Chapter 4. Table 4.3 and 4.4 (in Section 4.6) present the quantification coefficients of these six training windows, corresponding to the market combinations (1) and (2), respectively. The quantification coefficients shown in Table 4.3 are the values of ξ_i, $i = 1, 2, 3, 4$ and those given in Table 4.4 are the values of ξ_i^A, $i = 1, 2, 3, 4, 5$.

The classification and misclassification rates (Section 6.4) as well as the trading simulations (Section 6.4.1) were used to evaluate the prediction results obtained from training FNN, PNN and SVM.

The trading simulations described in Section 6.4.1 were performed on the trading signals generated by FNNs, PNNs and SVMs. In each case, the rate of return corresponding to each window was the highest when the whole amount of money in hand and the total amount of shares in hand was involved in training. In other words, the rate of return reached its highest value when L=1 (Section 6.4.1).

Subsection 6.6.1 to 6.6.3 describe the experiments relating to each algorithm used for predicting trading signals in detail. These sections also include the evaluation of prediction results.

6.6.1 Predicting Trading Signals using FNN

FNN Experiments

Three-layered FNNs with one hidden layer were trained 1000 times using the Resilient backpropagation training algorithm (Rprop). The reasons for using this algorithm were explained in Section 5.2.1. A tan-sigmoid function was used as the transfer function between the input layer and the hidden layer while the linear transformation function was employed between the hidden and the output layers. The reasons for selecting these functions are also described in Section 5.2.1. Different numbers of neurons for the hidden layer and different values for learning rate as well as the momentum coefficient were tested.

FNNs were trained for each one of the six windows considered. For FNN experiments, each training set was further divided into two sets; the most recent 22.2% of data of each training set (20% of the full data set) was allocated for validation while the remaining 77.8% (70% of the full data set) was used for training.

These FNNs were designed with the help of Matlab Neural Network Toolbox [14]. This toolbox uses the Nguyen-Widrow function [61] to initialise layers' weights and biases. Initial weight randomisations can greatly affect the solution to which the network converges [9]. To balance the effect of weight randomisation, this study trained FNNs 1000 times in each window.

The above mentioned six sets of inputs (Section 6.5) were considered when training the networks. These networks output the relative return of day $(t+1)$ of the AORD. The average value of the prediction (over 1000) for each day was calculated and this average value subsequently classified into the three classes of interest according to Criterion A (Section 6.2).

FNNs gave the best results when there were three neurons in the hidden layer and the learning rate and the momentum coefficient were 0.003 and 0.01, respectively.

Evaluation of the Prediction Results Using Classification and Misclassification Rates

Table 6.1 compares the average (over the six windows) classification and misclassification rates relating to the forecasting results obtained from the FNNs trained with input sets GFFG and GFFGA. Both input sets consist of the un-quantified relative returns of the Close prices of day t of the GSPC, FTSE, FCHI and GDAXI. In addition to these input features, the un-quantified relative return of day t of the AORD also included in the input set GFFGA.

Table 6.1: Average classification rate /misclassification rate corresponding to the results obtained from the FNNs trained with the input sets GFFG and GFFGA

Actual class	Average classification/misclassification rates for input set GFFG			Average classification/misclassification rates for input set GFFGA		
	Predicted class			Predicted class		
	Buy	Hold	Sell	Buy	Hold	Sell
Buy	26.40%	72.77%	0.83%	26.40%	72.77%	0.83%
Hold	5.79%	86.69%	7.52%	5.38%	86.28%	8.34%
Sell	0.00%	79.79%	20.21%	0.00%	77.71%	22.29%

The classification rate for the hold class is very high irrespective of the input features used (Table 6.1). When the un-quantified relative return of day t of the AORD was added to the input set, the classification rate relevant to the sell class increased. However, adding this extra input feature also resulted in an increase in misclassification rate from hold class to sell class, which is a negative impact. Although small, both cases show a very serious mistake which arose due the misclassification of 0.83% of buy signals as sell signals. It is not obvious whether adding the information of the AORD helped to improve the forecasting accuracy.

The average (over the six windows) classification and misclassification rates relating to the forecasting results obtained from the FNNs trained with input sets GFFG-q and GFFGA-q are shown in Table 6.2. The input set GFFG-q includes the quantified relative returns of the Close prices of day t of the GSPC, the three European market indices. In addition to these input features, the quantified relative return of the Close price of day t of the AORD is also included in input set GFFGA-q.

Table 6.2: Average classification rate /misclassification rate corresponding to the results obtained from the FNNs trained with the input sets GFFG-q and GFFGA-q

Actual class	Average classification/misclassification rates for input set GFFG-q			Average classification/misclassification rates for input set GFFGA-q		
	Predicted class			Predicted class		
	Buy	Hold	Sell	Buy	Hold	Sell
Buy	25.57%	74.43%	0.00%	26.40%	73.60%	0.00%
Hold	5.00%	88.30%	6.70%	5.00%	87.86%	7.14%
Sell	0.00%	79.79%	20.21%	0.00%	79.80%	20.21%

Although, the addition of the extra input feature associated with the AORD increased the prediction accuracy relating to buy signal, the rate of misclassification of hold signals to sell signal also increased (Table 6.2). Therefore, as with the previous case, it is not very clear that adding the extra input feature made any effect in improving the forecasting accuracy. However, when compared to the output relevant to un-quantified inputs, using quantified intermarket influences improved the forecasting accuracy in the sense that the later input features eliminated the serious mistake of misclassification of buy signals to sell signals (Table 6.1 and Table 6.2).

Table 6.3 presents the average (over the six windows) classification and misclassification rates relating to the forecasting results obtained from the FNNs trained with input sets GFFG-sq and GFFGA-sq. The input set GFFG-sq consists of the sum of the quantified relative returns of the Close prices of day t of the GSPC, the three European market indices, while the input set GFFGA-sq consists of the sum of the quantified relative returns of the Close prices of day t of the GSPC, the three European market indices and the AORD. The speciality of these two input sets is that each of them consists of only a single input feature.

Table 6.3: Average classification rate /misclassification rate corresponding to the results obtained from the FNNs trained with the input sets GFFG-sq and GFFGA-sq

Actual class	Average classification/misclassification rates for input set GFFG-sq			Average classification/misclassification rates for input set GFFGA-sq		
	Predicted class			Predicted class		
	Buy	Hold	Sell	Buy	Hold	Sell
Buy	21.55%	78.45%	0.00%	23.21%	76.79%	0.00%
Hold	4.18%	88.68%	7.14%	4.18%	88.68%	7.14%
Sell	0.00%	79.72%	20.28%	0.00%	80.83%	19.17%

In this case also there is no clear evidence that adding the information relevant to the AORD make an significant impact on improving the forecasting accuracy. Although there is an improvement in the forecasting accuracy of the buy signal, that relating to sell signal declined (Table 6.3). When compared with the previous two cases, the quantified single inputs reduced the forecasting accuracy corresponding to both buy and sell signals (Table 6.1 to 6.3). However, in this case also there is no serious mistakes such as misclassification of buy signals to sell signals.

Comparing all three tables (Table 6.1 to 6.3), it can be suggested that better forecasts can be achieved by training FNNs with multiple inputs of the quantified relative returns of the Close prices of day t of the GSPC, the European market indices and the AORD. In other words, it can be suggested that employing the quantified intermarket influences from the GSPC, the three European markets (FTSE, FCHI and GDAXI) and the AORD as multiple inputs improves the prediction accuracy, when compared to using the corresponding un-quantified intermarket influences.

Evaluation of the Prediction Results Using Trading Simulations

Table 6.4 shows the average rates of return obtained by performing the proposed trading simulation (described in Section 6.4.1) on the prediction results generated by the FNNs trained with the six sets of inputs (mentioned in Section 6.5).

Table 6.4: Average (over the six windows) rates of return relating to the FNNs trained with different input sets (*The annual average rate of return relating to the benchmark simulation = 9.57%*)

Input set	Rate of return for test period	Annual rate of return
GFFG	7.49%	25.23%
GFFGA	7.20%	24.25%
GFFG-q	7.69%	25.90%
GFFGA-q	6.71%	22.60%
GFFG-sq	7.33%	24.69%
GFFGA-sq	7.30%	24.59%

Table 6.4 demonstrates that, irrespective of the input set, a trader can gain higher profits by responding to the trading signals produced by the FNNs. The highest average rate of return was obtained when responding to the trading signals generated by the FNNs trained with the input set GFFG-q. This input set consists of the quantified relative returns of the Close prices of day t of the GSPC and the three European markets. Therefore, it can be suggested that the FNNs produced the trading signals which are more profitable, when they were trained with the multiple inputs of quantified intermarket influences of the GSPC and the three European markets.

Concluding Remarks

FNN produce better results (by means predictability and profitability) when the multiple inputs of the quantified relative returns of the influential markets were used for training. Therefore, it can be assumed that quantified intermarket influences can be effectively used to predict the trading signals of day $(t+1)$ of the AORD.

6.6.2 Predicting Trading Signals using PNN

PNN Experiments

PNNs were also trained for the same six moving windows, which were used for training FNNs (Section 6.6.1). The same six input sets (Section 6.5) were considered for network training. Networks output the class (buy, hold, or sell) relevant to the day t of the AORD.

The lost incurred by misclassification, L_C (Section 3.2.2), for each class was assumed to be equal. The joint distribution of the input variables was assumed to be Gaussian. The parameters of the distribution were estimated by using the training data. When there were multiple inputs, the average standard deviation of the individual input variables was considered as the standard deviation of the joint distribution.

Evaluation of the Prediction Results Using Classification Rate /Misclassification Rate

Table 6.5 compares the average classification rate /misclassification rate (over the six windows) relating to the prediction results produced by the PNNs trained with the input sets GFFG and GFFGA.

Table 6.5: Average classification rate /misclassification rate corresponding to the results obtained from the PNNs trained with the input sets GFFG and GFFGA

Actual class	Average classification/misclassification rates for input set GFFG			Average classification/misclassification rates for input set GFFGA		
	Predicted class			Predicted class		
	Buy	Hold	Sell	Buy	Hold	Sell
Buy	13.85%	86.15%	0.00%	15.52%	84.48%	0.00%
Hold	2.10%	93.00%	4.90%	2.47%	93.05%	4.48%
Sell	0.00%	84.10%	15.90%	0.00%	84.10%	15.90%

PNNs did not generate serious misclassifications when they were trained with unquantified relative returns of the Close prices of day t of the influential markets (Table 6.5). The average classification rates relevant to buy signals increased when the relative return of the Close price of day t of the AORD was added to the input features. Also adding this extra feature resulted in an increase in the rate of misclassification from hold signals to buy signals, while that corresponding to misclassification of hold signals to sell signals decreased. However, theses changes were not substantial.

The average classification rate /misclassification rate (over the six windows) corresponding to the prediction results obtained by the PNNs trained with the input sets GFFG-q and GFFGA-q are shown in Table 6.6.

The PNNs trained with the quantified relative returns of the Close prices of day t of the influential market also did not produce serious misclassifications (Table 6.6). When

Table 6.6: Average classification rate /misclassification rate corresponding to the results obtained from the PNNs trained with the input sets GFFG-q and GFFGA-q

Actual class	Average classification/misclassification rates for input set GFFG-q			Average classification/misclassification rates for input set GFFGA-q		
	Predicted class			Predicted class		
	Buy	Hold	Sell	Buy	Hold	Sell
Buy	12.29%	87.71%	0.00%	14.76%	85.24%	0.00%
Hold	2.47%	91.70%	5.83%	2.47%	91.70%	5.83%
Sell	0.00%	81.74%	18.26%	0.00%	82.01%	17.99%

the quantified relative return of the Close price of day t of the AORD was added to the input features, the average classification rate of buy signals increased by 20% (from 12.29% to 14.76%), which is a substantial increase. In contrast, this rate relevant to sell signals decreased by 1% (from 18.26% to 17.99%) which is not a substantial drop. The misclassification rates of hold signals to buy/sell remained unchanged.

Table 6.7 presents the average classification rate /misclassification rate (over the six windows) relating to the trading signals produced by the PNNs when they were trained with the input sets GFFG-sq and GFFGA-sq.

Table 6.7 shows that there were no serious misclassifications when the PNNs trained with the single inputs which consist of the sum of the quantified relative returns of the Close prices of day t of the influential markets. The sum of the quantified relative returns of the Close prices of day t of the GSPC, the three European markets and the AORD, showed lower classification rates of buy and sell signals, than the respective rates corresponding to its counterpart without the AORD.

Comparing Table 6.5 to 6.7, it can be suggested that a better prediction accuracy can be obtained when the quantified intermarket influences from the GSPC, the three European markets as well as the AORD itself are used as the input features to predict

Table 6.7: Average classification rate /misclassification rate corresponding to the results obtained from the PNNs trained with the input sets GFFG-sq and GFFGA-sq

Actual class	Average classification/misclassification rates for input set GFFG-sq			Average classification/misclassification rates for input set GFFGA-sq		
	Predicted class			Predicted class		
	Buy	Hold	Sell	Buy	Hold	Sell
Buy	9.30%	90.70%	0.00%	7.79%	92.21%	0.00%
Hold	1.20%	95.50%	3.30%	1.23%	95.49%	3.28%
Sell	0.00%	88.00%	12.00%	0.00%	89.10%	10.90%

the trading signals.

When the prediction results of PNNs (Table 6.5 to 6.7) trained with different inputs are compared with their respective results produced by the FNNs (Table 6.1 to 6.3), it is obvious that FNNs produce predictions with higher accuracy.

Although, the past studies (Section 2.2.1 and 2.2.2) showed that the PNN provides good results as a classifier, it did not provide the expected results in this study (we noted that no buy or sell signals were predicted in some windows). The reason may be that the previous studies aimed at predicting only two classes and the data used is approximately evenly distributed among these two classes. In contrast, this study considered three classes and the data is not evenly distributed; usually the hold class dominates. Table 6.8, which presents the distribution of the test data among three classes confirms this matter.

Irrespective of the window, the hold class dominates (Table 6.8). In each window, at least 50% of the data falls into this class.

To address the problem of imbalanced distribution of data, one possibility is to allocate different values as the lost incurred by misclassification (L_C in 3.9 in Chapter 3) for different classes (trading signals). However, there is no particular way to estimate these values; the only possibility is to follow a trial and error method. The PNN experiments

Table 6.8: Distribution of test data among buy, hold and sell classes (Percentage of data in each class is also shown in brackets)

Window Number	Buy	Hold	Sell
1	20 (26.32%)	40 (52.63%)	16 (21.05%)
2	20 (26.32%)	44 (57.89%)	12 (15.79%)
3	23 (30.26%)	38 (50.00%)	15 (19.74%)
4	12 (15.79%)	56 (73.68%)	08 (10.53%)
5	11 (14.47%)	59 (77.63%)	06 (7.90%)
6	21 (27.63%)	40 (52.63%)	15 (19.74%)

were repeated by randomly allocating higher values for the L_C corresponding to the buy and sell classes. This attempt was also unsuccessful as it resulted in higher rate of misclassification of buy signals as sell signals and vice-versa.

The other reason behind the less satisfactory performance by PNN (compared to that of FNN) may be that the deviation of distributions of the input variables from the Gaussian. Some past studies [3, 24, 50] (Section 2.3.2) provide evidence that the distribution of the relative returns shows deviations from the Gaussian distribution.

Evaluation of the Prediction Results Using Trading Simulations

The average (over the six windows) rates of return obtained by performing the proposed trading simulations (Section 6.4.1) on the trading signals produced by the PNNs trained with the six types of inputs (Section 6.5) are shown in Table 6.9.

Table 6.9 also evidences that a trader can make higher profits by responding to the trading signals produced by the PNNs trained with any type of input. The highest average rate of return was obtained when the trading simulations were performed on the trading signals produced by the PNNs trained with the input set GFFGA-q. This input set includes multiple inputs of the quantified relative returns of the Close prices of day t of the GSPC, the three European markets and the AORD. This suggests that the PNNs

Table 6.9: Average (over the six windows) rates of return relating to the PNNs trained with different input sets (*The annual average rate of return relating to the benchmark simulation = 9.57%*)

Input set	Rate of return for test period	Annual rate of return
GFFG	5.09%	17.15%
GFFGA	5.13%	17.28%
GFFG-q	5.06%	17.04%
GFFGA-q	5.97%	20.11%
GFFG-sq	4.04%	13.61%
GFFGA-sq	4.17%	14.05%

produced the more profitable trading signals when they were trained with quantified intermarket influences of these markets.

When compared with the average rates of return corresponding to FNNs (Table 6.4), the respective rates of returns relevant to PNNs are smaller. Therefore, for the purpose of gaining profits, FNNs produce more accurate predictions, than the PNNs. As described earlier, reason may that PNNs does not perform well, when the data is imbalanced.

Concluding Remarks

The results obtained from the PNNs also suggest that the quantified intermarket influences of the influential markets can be effectively used to obtain the trading signals of day $(t+1)$ of the AORD. In other words, the predictability and the profitability of the trading signals are higher when the PNNs were trained with quantified intermarket influences.

6.6.3 Predicting Trading Signals using SVM

SVM Experiments

This study used SVM-Light (version 6.01) software developed by Joachims [29] to train SVMs. The past studies [8, 34, 78], which focused on predictions relating to financial time series, used the radial basis function (Gaussian kernel) as the kernel function (Section 3.2.3). Following these studies, radial basis function was adopted as the kernel function.

Kim [34] found that the parameter of the radial basis function, γ gave the best results (for predicting the direction of the Korean composite stock price index) when it equals 5. Therefore, different values around 5 were tested for the value of γ. SVMs used in this study also predicted the trading signals more accurately when γ=5.

The same six windows which were used for the experiments with FNN and PNN, were considered for the SVM experiments. Training SVM with one input feature is not meaningful. Therefore, the SVMs were trained with the input sets GFFG, GFFG-q, GFFGA and GFFGA-q only (Section 6.5).

Evaluation of the Prediction Results

Table 6.10 compares the average (over the six windows) classification rate /misclassification rate relating to the prediction results produced by the SVMs trained with the input sets GFFG and GFFGA while Table 6.11 compares those rates corresponding to the input sets GFFG-q and GFFGA-q.

Irrespective of the input set, SVM did not produce serious misclassifications (Table 6.10 and 6.11). SVMs trained with input features which include the un-quantified relative returns (Table 6.10) yielded better prediction accuracies than those trained with their quantified counterparts (Table 6.11). Both tables evidence that there is no impact on the prediction accuracy by including the information relevant to the AORD to the input sets.

Table 6.10: Average classification rate /misclassification rate corresponding to the results obtained from the SVM trained with the input sets GFFG and GFFGA

Actual class	Average classification/misclassification rates for input set GFFG			Average classification/misclassification rates for input set GFFGA		
	Predicted class			Predicted class		
	Buy	Hold	Sell	Buy	Hold	Sell
Buy	7.07%	92.93%	0.00%	7.90%	92.10%	0.00%
Hold	1.23%	95.49%	3.28%	1.23%	95.49%	3.28%
Sell	0.00%	89.31%	10.69%	0.00%	89.31%	10.69%

Table 6.11: Average classification rate /misclassification rate corresponding to the results obtained from the SVM trained with the input sets GFFG-q and GFFGA-q

Actual class	Average classification/misclassification rates for input set GFFG-q			Average classification/misclassification rates for input set GFFGA-q		
	Predicted class			Predicted class		
	Buy	Hold	Sell	Buy	Hold	Sell
Buy	1.56%	98.44%	0.00%	1.56%	98.44%	0.00%
Hold	0.44%	96.68%	2.88%	0.44%	96.68%	2.88%
Sell	0.00%	91.67%	8.33%	0.00%	91.67%	8.33%

Although, the SVM is proved to be a promising method for different classification problems (for example [26, 28, 34]), it did not produce the expected classification accuracy for the classification problem of interest in this study. In majority of cases it classified all signals as hold signals. The literature (Section 2.3.3) provides evidence for poor performance of SVM when data is not equally distributed among the classes of interest (imbalanced distribution). The data used in this study has an imbalanced distribution

(Table 6.8). This may be the reason for the poor performance showed by the SVM in predicting trading signals of the AORD.

Joachims [29] introduced a correction factor to SVM-Light (version 6.01) to deal with the problem of imbalanced data. We attempted to improve the classification accuracy by assigning different values for this correction factor when training SVM. However, our attempt was not successful.

As explained in Section 2.3.3, there are some modified SVM algorithms (for example [1, 27, 97]) to deal with imbalanced data. However, due to the unavailability of the source codes of these modified SVM algorithms, this study was not able to apply them to classify the trading signals.

Evaluation of the Prediction Results Using Trading Simulations

Table 6.12 presents the average (over the six windows) rates of return obtained by performing the proposed trading simulation (described in Section 6.4.1) on the prediction results generated by the SVM trained with the input sets GFFG, GFFGA, GFFG-q and GFFGA-q (Section 6.5).

Table 6.12: Average (over the six windows) rates of return relating to the SVMs trained with different input sets (*The annual average rate of return relating to the benchmark simulation = 9.57%*)

Input set	Rate of return for test period	Annual rate of return
GFFG	4.16%	14.01%
GFFGA	4.64%	15.63%
GFFG-q	3.25%	10.95%
GFFGA-q	2.80%	9.43%

In contrast to the results of the trading simulations relating to FNNs (Table 6.4) and PNNs (Table 6.9), the higher average rates of return was obtained when the SVMs were trained with un-quantified relative returns (Table 6.12).

Concluding Remarks

The predictions obtained from the SVM algorithm used (SVM-Light), are poorer than those obtained by the FNNs and the PNNs. The reason may be that the ability of the SVM algorithm to produce precise predictions is weaken by the imbalanced distribution of the data used.

The results produced by the SVM algorithm are not precise enough to suggest that the inputs with the un-quantified relative returns are better than their quantified counterparts, to predict trading signals.

6.7 Shortcomings of the Algorithms

The experiments results suggest that the SVM algorithm used did not perform well as a classification algorithm to forecast the trading signals. The predictions produced by the PNN are also not as satisfactory as those produced by the FNNs. According to Criterion A (Section 6.2), the hold class dominates creating an imbalance in the data. This seems to be the main reason for the low performance of the two algorithms, the SVM and PNN. The FNN proved to be a better algorithm to classify trading signals in that sense.

As mentioned earlier (Section 6.6.1) Matlab Neural Network Toolbox [14] was employed to develop the FNNs used for forecasting. Layers' weights and biases were initialised using the Nguyen-Widrow function [61] and gradient descent algorithm with momentum was employed for weight modification. The FNNs were trained by the Resilient backpropagation training algorithm [71].

The Nguyen-Widrow function uses different initial values for network parameters (weights and biases) [61]. This results in different solutions for network parameters. Since the network parameters vary according to their initial values, the network output also varies. The general practice to overcome this problem is to train neural networks for a number of times and calculate the average output.

In the experiments, we trained the networks 1000 times (Section 6.6.1). Each time the network gave one local solution to the error function. Figure 6.1 and 6.2 present the bar charts corresponding to windows 1 and 5. These windows were arbitrarily selected out of the six windows considered. These figures depict the distribution of the 1000 local solutions of the error function relevant to each window. The input set GFFG-q was selected as the input features, as this set gave the best forecasting performance for FNNs.

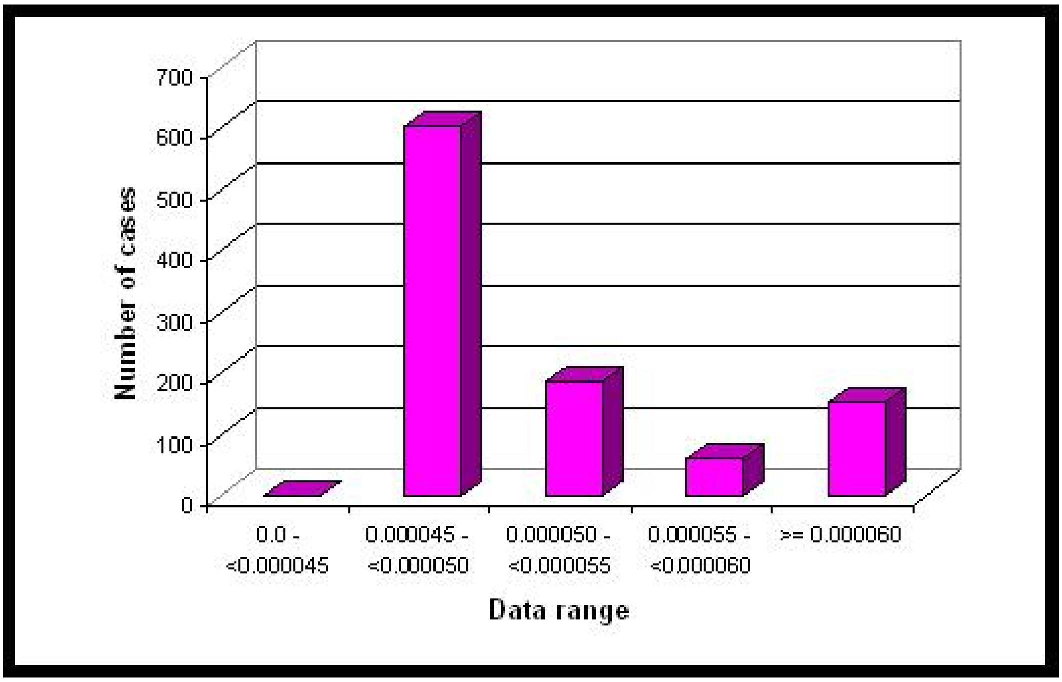

Figure 6.1: Bar chart for the distribution of the local optimal values of the error function corresponding to window 1 (The local optimal values vary in the interval [4.458×10^{-5}, 0.0079].)

Both charts confirm that the solution to the error function varies (Figure 6.1 and 6.2). For example, in window 1, almost all the local solutions are far from the global solution. In window 5, the solutions relating to the first bar might be close to the global solution; however, a significant number of local solutions are located far from the global

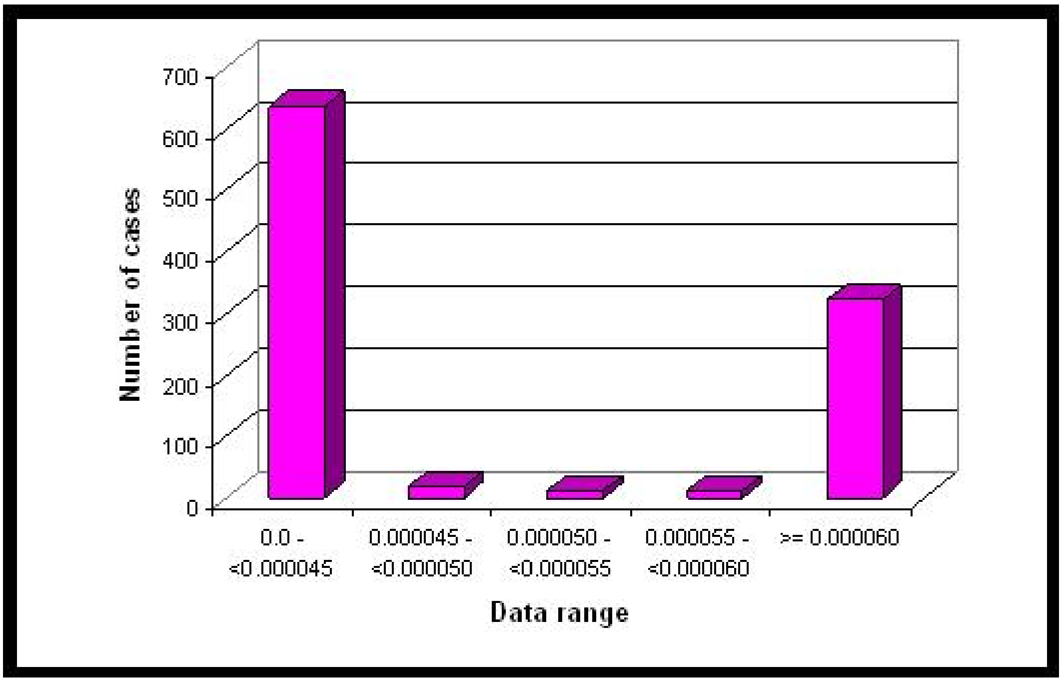

Figure 6.2: Bar chart for the distribution of the local optimal values of the error function corresponding to window 5 (The local optimal values vary in the interval [2.3973×10^{-5}, 0.01742].)

solution. Therefore, it can be suggested that the majority of the predictions produced by the feedforward neural networks used by this study, are based on local solutions which are far from the global solutions.

The other disadvantage of the standard FNNs, is the usage of the OLS function (see (3.3)) as the error function to be minimised. A standard FNN tries to minimise the deviation between the actual value and the predicted value, and it does not agree with the objective of the classification problem of interest of this study (Section 2.3.1).

Although, this study assumed the distribution of the input features, which are relative returns of stock market indices, as Gaussian, the literature argues that the relative returns have heavy tailed distributions [3, 24, 50]. Apart from the imbalanced distribution of data among three classes, the deviation of the distribution of data within each class from the Gaussian, may resulted in bias predictions by PNNs.

As discussed earlier in Section 2.3.3, the SVM algorithm does not perform well when data shows imbalanced distribution among the classes of interest.

6.8 Summary

This chapter aimed to forecast whether it is best to buy, hold or sell shares (trading signals) of the AORD. The algorithms used for forecasting were FNN, PNN, and SVM. The forecasting results were evaluated by the classification rate /misclassification rate (predictability) and rate of return (profitability) obtained from performing trading simulations. A new trading simulation method was proposed. The special feature in this proposed simulation is that it searches for the proportion of money to be invested as well as the proportion of shares to be sold in order to obtain higher profits.

In this study, both the PNN and SVM did not produce results as accurate as those of the FNN. The reason for poor performance of the the PNN and SVM may be the imbalanced distribution of the data.

The results obtained from the FNN suggested that better results can be achieved by employing multiple quantified relative returns of the Close prices of day t of the GSPC, the European market indices and the AORD, as input features. This fact indicates the effectiveness of the proposed technique (Section 4.2) for quantifying intermarket influences.

However, there are some drawbacks associated with the FNNs used for forecasting. The next chapter (Chapter 7) focuses on overcoming these drawbacks in order to improve the forecasting accuracy.

Chapter 7

Development of New Algorithms for Predicting Trading Signals

7.1 Introduction

One of the algorithms that was employed to predict the trading signals in the previous chapter (Chapter 6) was the FNN. The FNN produced predictions results with higher accuracy than other two algorithms used.

The FNN uses backpropagation learning procedure to minimise the OLS error function (see (3.3) in Chapter 3). The distribution of the minimal values of this error function demonstrated that the networks used in the previous chapter provided solutions that could be far away from the global solution (Section 6.7). Literature [15, 32] also reveals the possibily of FNN finding suboptimal solutions as a result of being trapped in local minima (Section 2.3.1).

To overcome this problem, finding a global solution to the error minimisation function is required. Several attempts to find global solutions for the parameters of the FNNs, by developing new algorithms, are found in the literature (Section 2.3.1).

The main aim of this chapter is to develop new prediction algorithms which improve the prediction accuracy of trading signals. Obtaining a global solution as the network output may lead to an improvement of prediction accuracy to some extend. Another possible way to improve the prediction accuracy is to consider new or modified error functions (Section 2.3.1). Therefore, this chapter focuses on developing new neural network algorithms using:

1. a global optimization algorithm for neural network training;

2. modifications to the OLS error function.

This chapter describes the development of new neural network algorithms. The prediction results obtained with these new algorithms is presented and compared with those of FNNs. The improvements in the prediction accuracy made by these new networks is also highlighted.

7.2 Development of New Neural Network Algorithms

The new neural network algorithms were developed by : (1) using the OLS error function (see 3.3) as well as the modified least squares error functions; and, (2) employing a global optimization algorithm to training the networks. The structure of these algorithms are based on FNN. The structure of FNN is already described in Section 3.2.1.

The reason for using a global optimization technique for network training is to obtain a better single solution for the weights and biases of the network. The global optimization algorithm (described in Section 4.2.2), which was used in the process of quantifying intermarket influences, was used as the network training algorithm.

In addition to the OLS error function, the alternative least squares error functions found in the literature were considered (Section 2.3.1). Some of these alternative error function were modified to suit the prediction problem of interest; predicting whether it is best to buy, hold or sell (predicting trading signals). By considering alternative error functions, this study aimed at improving the predicting accuracy.

7.2.1 Alternative Error Functions

In financial applications, it is more important to predict the direction of a time series rather than its value. Therefore, the minimisation of the absolute errors between the target and the output may not produce the desired results [101, 102]. Having this idea in mind, some past studies aimed to modify the error function associated with the FNNs (for instance [7, 69, 101, 102]). These studies incorporated factors which represent the direction of the prediction (for example [7, 101, 102]) and contribution from the historical data that used as inputs (for example [69, 101, 102]).

This study considered the suitable alternatives to the OLS error function found in the literature as well as modified error functions.

Alternative Error Functions

The functions proposed in [7, 101, 102] penalised the incorrectly predicted directions more heavily, than the correct predictions. In other words, higher penalty was applied if the predicted value (o_i) is negative when the target (a_i) is positive or vice-versa.

Caldwell [7] proposed The Weighted Directional Symmetry (WDS) function which is given below:

$$f_{WDS}(i) = \frac{100}{N} \sum_{i=1}^{N} w_{ds}(i) |a_i - o_i| \tag{7.1}$$

where

$$w_{ds}(i) = \begin{cases} 1.5 & \text{if } (a_i - a_{i-1})(o_i - o_{i-1}) \leq 0, \\ 0.5 & \text{otherwise,} \end{cases} \tag{7.2}$$

and N is the total number of observations.

Yao and Tan [101, 102] argued that weight associated with f_{WDS} (that is $w_{ds}(i)$) should be adjusted more if a wrong direction is predicted for a larger change, while, it should be adjusted less if a wrong direction is predicted for a smaller change and so on. Based on this argument, they proposed the Directional Profit adjustment factor:

$$f_{DP}(i) = \begin{cases} c_1 & \text{if } (\Delta a_i \times \Delta o_i) > 0 \quad \text{and } \Delta a_i \leq \sigma, \\ c_2 & \text{if } (\Delta a_i \times \Delta o_i) > 0 \quad \text{and } \Delta a_i > \sigma, \\ c_3 & \text{if } (\Delta a_i \times \Delta o_i) < 0 \quad \text{and } \Delta a_i \leq \sigma, \\ c_4 & \text{if } (\Delta a_i \times \Delta o_i) < 0 \quad \text{and } \Delta a_i > \sigma. \end{cases} \tag{7.3}$$

where $\Delta a_i = a_i - a_{i-1}$, $\Delta o_i = o_i - a_{i-1}$ and σ is the standard deviation of the training data (including validation set). For their experiments, Yao and Tan [101, 102] used $c_1 = 0.5$, $c_2 = 0.8$, $c_3 = 1.2$ and $c_4 = 1.5$.

Based on this Directional Profit adjustment factor (see (7.3)), they proposed Directional Profit (DP) model [101, 102]:

$$E_{DP} = \frac{1}{N} \sum_{i=1}^{N} f_{DP}(i)(a_i - o_i)^2. \tag{7.4}$$

Refenes at el. [69] proposed Discounted Least Squares (LDS) function by taking the recency of the observations (that is, representation of the contribution of past data of a time series) into account.

$$E_{DLS} = \frac{1}{N} \sum_{i=1}^{N} w_b(i)(a_i - o_i)^2 \tag{7.5}$$

where $w_b(i)$ is an adjustment relating to the contribution of the ith observation and is described by the following equation:

$$w_b(i) = \frac{1}{1 + exp(b - \frac{2bi}{N})}. \tag{7.6}$$

Discount rate b, decides the recency of the observation. Refenes at el. [69] suggested $b = 6$.

Yao and Tan [101, 102] proposed another error function, Time Dependent Directional Profit (TDP) model, by incorporating the approach suggested by Refenes at el. [69] to their Directional Profit model (see (7.4)):

$$E_{TDP} = \frac{1}{N} \sum_{i=1}^{N} f_{TDP}(i)(a_i - o_i)^2 \tag{7.7}$$

where $f_{TDP}(i)$=$f_{DP}(i) \times w_b(i)$. $f_{DP}(i)$ and $w_b(i)$ are described by (7.3) and (7.6), respectively.

Note: In [69, 101, 102], $\frac{1}{2N}$ was used instead of $\frac{1}{N}$ in the formulas given by (7.4), (7.5) and (7.7).

Modified Error Functions

This study considers three classes: buy, hold and sell. The hold class includes both positive and negative values (refer Criterion A in Section 6.2). Therefore, the error functions in which the cases with incorrectly predicted directions (positive or negative) are penalised (for example (7.4) and (7.7)), will not give the desired prediction accuracy for this study. Instead of the weighing schemes suggested by previous studies, this study proposes a novel scheme of weighing.

This novel scheme is based on the correctness of the classification of trading signals. If the predicted trading signal is correct, we assign a very small (close to zero) weight, otherwise, assign weight equal to 1.

Therefore, the proposed weighing scheme is:

$$w_d(i) = \begin{cases} \delta & \text{if the predicted trading signal is correct,} \\ 1 & \text{otherwise.} \end{cases} \tag{7.8}$$

where δ is a very small value. The value of δ needs to be decided according to the distribution of data.

Proposed Error Function 1

The weighing scheme, $f_{DP}(i)$, incorporated in the DP error function (7.4) considers only two classes, upward and downward trend (direction) which are corresponding to buy and sell signals. In order to deal with three classes, buy, hold and sell, we modified this error function by replacing $f_{DP}(i)$ with the new weighing scheme, $w_d(i)$ (see (7.8)). Hence, the new error function (E_{CC}) is defined as:

$$E_{CC} = \frac{1}{N}\sum_{i=1}^{N} w_d(i)(a_i - o_i)^2 \tag{7.9}$$

When training backpropagation neural networks using (7.9) as the error minimisation function, the error is forced to take a smaller value, if the predicted trading signal is correct. On the other hand, the actual size of the error is considered in the cases of misclassifications.

Proposed Error Function 2

Recency of the data (contribution of the past data of the corresponding time series) also plays an important role in the prediction accuracy of financial time series. Therefore, [101, 102] went further, by combining DP error function (7.4) with LDS error function (7.5) and proposed TDP error function (7.7).

Following Yao and Tan[101, 102], this study also proposed a second new error function, E_{TCC}, by combining first new error function (E_{CC}) described by 7.9 with the DLS function (E_{DLS}). Hence the second proposed error minimisation function is:

$$E_{TCC} = \frac{1}{N}\sum_{i=1}^{N} w_b(i) \times w_d(i)(t_i - o_i)^2 \tag{7.10}$$

where $w_b(i)$ is defined by Equation 7.6 while Equation 7.8 defines $w_d(i)$.

7.3 New Neural Network Algorithms

In these experiments, four types of neural network algorithms were employed. These four networks were based on four error functions described by (3.3), (7.5), (7.9) and (7.10). Following notations were used to denote the algorithms:

NN_{OLS} - Neural network algorithm based on OLS error function, E_{OLS} (see (3.3))

NN_{DLS} - Neural network algorithm based on DLS error function, E_{DLS} (see (7.5))

NN_{CC} - Neural network algorithm based on the newly proposed error function 1, E_{CC} (see (7.9))

NN_{TCC} Neural network algorithm based on the newly proposed error function 2, E_{TCC} (see (7.10))

The global optimization algorithm (Section 4.2.2), which was used in the process of quantifying intermarket influences, was used to train these networks. Unlike the FNNs described in the previous chapter, each network produces a single but deep solution which minimises the respective error function. The main advantage of having one (global) solution is that proper estimates of the network parameters (weights and biases) can be obtained.

The structure of the new networks is the same as that of the FNNs employed in the previous chapter. These new networks also consist of three layers: an input, hidden and output layer. The layers are connected using the same structure as the FNN (Section 3.2.1). A tan-sigmoid function was used as the transfer function between the input layer and the hidden layer while the linear transformation function was employed between the hidden and the output layers.

7.3.1 Network Training

The new networks introduced (Section 7.3) were trained with the same six sets of inputs (Section 6.5) which were used as the input features of the FNNs in the previous experiments:

1. Four input features consisting of the relative returns of day t of the Close prices of the market combination (1) (denoted by **GFFG**)
   - (GSPC(t), FTSE(t), FCHI(t), GDAXI(t));

2. Four input features consisting of the quantified relative returns of day t of the Close prices of the market combination (1) (denoted by **GFFG-q**)
   - (ξ_1GSPC(t), ξ_2FTSE(t), ξ_3FCHI(t), ξ_4GDAXI(t));

3. Single input feature consisting of the sum of the quantified relative returns of day t of the Close prices of the market combination (1) (denoted by **GFFG-sq**)
   - (ξ_1GSPC(t)+ξ_2FTSE(t)+ξ_3FCHI(t)+ξ_4GDAXI(t));

4. Five input features consisting of the relative returns of day t of the Close prices of the market combination (2) (denoted by **GFFGA**)
   - (GSPC(t), FTSE(t), FCHI(t), GDAXI(t), AORD(t));

5. Five input features consisting of the quantified relative returns of the Close prices of day t of the market combination (2) (denoted by **GFFGA-q**)
   - (ξ_1^AGSPC(t), ξ_2^AFTSE(t), ξ_3^AFCHI(t), ξ_4^AGDAXI(t), ξ_5AORD(t));

6. Single input feature consisting of the sum of the quantified relative returns of the Close prices of day t of the market combination (2) (denoted by **GFFGA-sq**)
   - (ξ_1^AGSPC(t)+ξ_2^AFTSE(t)+ξ_3^AFCHI(t)+ξ_4^AGDAXI(t)+ξ_5^AAORD(t)).

The values of ξ_i, $i = 1, 2, 3, 4$ and ξ_i^A, $i = 1, 2, 3, 4, 5$ are shown in the Table 4.3 and 4.4 (in Section 4.6), respectively.

Different numbers of neurons for the hidden layer were tested when training the networks with each input set.

The same six moving windows used in the previous chapter (Section 6.6) were considered for these experiments. Each of these windows consists of a test set (the most recent 76 samples) and a training set (the remaining 692 samples).

The minimum and the maximum values of the data (relative returns) used for network training are -0.137 and 0.057, respectively. Therefore we selected the value of δ (see (7.8)) as 0.01. This value is small enough to set the value of the proposed error functions (7.9 and 7.10) approximately zero, if the trading signals are correctly predicted.

As described in Section 7.2.1, the error function, E_{DLS} (see (7.5)), consists of a parameter b (discount rate) which decides the recency of the observations in the time series. Refenes at el. [69] fixed b=6 for their experiments. However, the discount rate may vary from one stock market index to another. Therefore, this study tested different values for

b when training network NN_{DLS}. Observing the results, the best value for b (that is, b=5; see Section 7.4) was selected and this best value was used as b when training network NN_{TCC}.

7.3.2 Evaluation Measures

As with the FNNs used in the previous chapter, these networks (Section 7.3) also output the relative returns of the Close price of day $(t+1)$ of the AORD. Subsequently, the output was classified into trading signals according to Criterion A with $l_u = -l_l = 0.005$ (Section 6.2).

The performance of the networks was evaluated by the overall classification rate (r_{CA}) as well as by the overall misclassification rates (r_{E1} and r_{E2}) which are defined as follows:

$$r_{CA} = \frac{N_0}{N_T} \times 100 \tag{7.11}$$

where N_0 and N_T are the number of test cases with correctly predicted trading signals and the total number of cases in the test sample, respectively.

$$r_{E1} = \frac{N_1}{N_T} \times 100 \tag{7.12}$$

$$r_{E2} = \frac{N_2}{N_T} \times 100 \tag{7.13}$$

where N_1 is the number of test cases where a buy/sell signal is misclassified as a hold signals or vice versa. N_2 is the test cases where a sell signal is classified as a buy signal and vice versa. Because of the seriousness of the mistake, r_{E2} plays a more important role in performance evaluation than r_{E1} (more detailed explanation is given in Section 6.4).

7.4 Results Obtained from Network Training

As mentioned in Section 7.3.1, different values for the discount rate, b (see (7.5) and (7.6)), were tested. b=1, 2, ..., 12 was considered when training NN_{DLS}. The prediction results improved with the value of b up to b=5. For $b \geq 5$ the prediction results remained unchanged. Therefore, value of b was fixed at 5. This value was used as the discount rate also in NN_{TCC} algorithm (see (7.10)).

The best four prediction results corresponding to the four networks were obtained when the number of hidden neurons was equal to two. Therefore, only the results relevant to networks with two hidden neurons are presented. Table 7.1 to 7.4 present the results relating to neural networks, NN_{OLS}, NN_{DLS}, NN_{CC} and NN_{TCC}, respectively.

Table 7.1: Results obtained from training neural network, NN_{OLS} (The best prediction result produced by the network is shown in bold colour)

Input set	Average r_{CA}	Average r_{E2}	Average r_{E1}
GFFG	64.25%	0.00%	35.75%
GFFGA	64.25%	0.00%	35.75%
GFFG-q	**64.69%**	**0.00%**	**35.31%**
GFFGA-q	64.04%	0.00%	35.96%
GFFG-sq	63.82%	0.00%	36.18%
GFFGA-sq	63.60%	0.00%	36.40%

Table 7.2: Results obtained from training neural network, NN_{DLS} (The best prediction result produced by the network is shown in bold colour)

Input set	Average r_{CA}	Average r_{E2}	Average r_{E1}
GFFG	64.25%	0.44%	35.31%
GFFGA	64.04%	0.44%	35.53%
GFFG-q	64.47%	0.22%	35.31%
GFFGA-q	64.25%	0.22%	35.53%
GFFG-sq	63.82%	0.00%	36.18%
GFFGA-sq	**64.04%**	**0.00%**	**35.96%**

Table 7.3: Results obtained from training neural network, NN_{CC} (The best prediction result produced by the network is shown in bold colour)

Input set	Average r_{CA}	Average r_{E2}	Average r_{E1}
GFFG	**65.35%**	**0.00%**	**34.65%**
GFFGA	64.04%	0.22%	35.75%
GFFG-q	63.82%	0.00%	36.18%
GFFGA-q	64.04%	0.00%	35.96%
GFFG-sq	64.25%	0.00%	35.75%
GFFGA-sq	63.82%	0.00%	36.18%

Table 7.4: Results obtained from training neural network, NN_{TCC} (The best prediction result produced by the network is shown in bold colour)

Input set	Average r_{CA}	Average r_{E2}	Average r_{E1}
GFFG	66.67%	0.44%	32.89%
GFFGA	64.91%	0.22%	34.87%
GFFG-q	**66.23%**	**0.00%**	**33.77%**
GFFGA-q	63.82%	0.22%	35.96%
GFFG-sq	64.25%	0.44%	35.31%
GFFGA-sq	64.69%	0.22%	35.09%

The best prediction from NN_{OLS} was obtained when the input set GFFG-q (Section 7.3.1) was used as the input features (Table 7.1). This input set consists of four inputs of the quantified relative returns of the Close price of day t of the GSPC and the European stock indices.

NN_{DLS} yielded non-zero values for the more serious classification error, r_{E2}, when multiple inputs (either quantified or not) were used as the input features (Table 7.2). The best results were obtained when the networks were trained with a single input representing the sum of the quantified relative returns of the Close prices of day t of the GSPC, the European market indices and the AORD (input set GFFGA-sq; Section 7.3.1). When the networks were trained with single inputs (input sets GFFG-sq and GFFGA-sq; Section 7.3.1) the serious misclassifications were prevented.

The overall prediction results obtained from the NN_{OLS} seem to be better than those relating to NN_{DLS} error function (Table 7.1 and 7.2).

Compared to the predictions obtained from NN_{DLS}, those relating to NN_{CC} are better (Table 7.2 and 7.3). In this case the best prediction results were obtained when the relative returns of the Close price of day t of the GSPC and the three European stock market indices (input set GFFG) were used as the input features (Table 7.3). The classification rate was increased by 1.02% compared to that of the best prediction results produced by NN_{OLS} (Table 7.1 and 7.3).

Table 7.4 shows that NN_{TCC} also produced serious misclassifications. However, these networks produced high overall classification accuracy and prevented serious misclassifications when the quantified relative returns of the Close prices of day t of the GSPC and the European stock market indices (input set GFFG-q) were used as the input features. The accuracy was the best among all four types of new neural network algorithms considered in this study.

NN_{TCC} yielded 1.34% increase in the overall classification rate compared to NN_{CC}. When compared with NN_{OLS}, NN_{TCC} showed a 2.37% increase in the overall classification rate and this can be considered as a good improvement in predicting trading signals.

7.4.1 Comparison of NN_{OLS} and the FNNs

Both the FNNs used in the previous chapter and NN_{OLS} use a common error function: the OLS error function. The usage of this common error function made it possible to compare the performance of the two types networks. Comparison is twofold: (1) Comparison of prediction results and (2) Comparison of the optimal values of the error function.

Comparison of Prediction Results Obtained from NN_{OLS} and the FNNs

Table 7.5 to Table 7.7 show the average (over the six windows) classification/ misclassification rates obtained from NN_{OLS} which included two neurons in the hidden layer. Table 7.5 compares the results of the two un-quantified input sets GFFG and GFFGA (Section 7.3.1). The results obtained from the networks trained with the sets of multiple quantified input features, GFFG-q and GFFGA-q (Section 7.3.1) are presented in Table 7.6 while those corresponding to the input sets consist single inputs, GFFG-sq and GFFGA-sq are given in Table 7.7.

Table 7.5: Average (over the six windows) classification rate /misclassification rate corresponding to the results obtained from the NN_{OLS} trained with input sets GFFG and GFFGA

Actual class	Average classification/misclassification rates for input set GFFG			Average classification/misclassification rates for input set GFFGA		
	Predicted class			Predicted class		
	Buy	Hold	Sell	Buy	Hold	Sell
Buy	23.46%	76.54%	0.00%	23.46%	76.54%	0.00%
Hold	5.02%	87.46%	7.52%	5.02%	87.46%	7.52%
Sell	0.00%	79.79%	20.21%	0.00%	79.79%	20.21%

Compared to the FNNs trained with input sets with input features which are not quantified (GFFG and GFFGA), NN_{OLS} trained with the respective inputs showed a lower classification accuracy when predicting buy signals (Table 6.1 and Table 7.5). However, unlike the FNNs, NN_{OLS} eliminated the serious misclassifications such as misclassification of buy signals to sell signals.

NN_{OLS} trained with the quantified multiple input sets, GFFG-q and GFFGA-q, also predicted a lower number of correct buy signals with compared to their respective FNN counterparts (Table 6.2 and Table 7.6). As with the FNNs, these networks also did not produced serious misclassifications.

When trained with the single input sets (that is the sum of the quantified relative returns), GFFG-sq and GFFGA-sq, NN_{OLS} produced a lower number of correct buy as well as correct sell signals, compared to their respective FNN counterparts (Table 6.3

Table 7.6: Average (over the six windows) classification rate /misclassification rate corresponding to the results obtained from the NN_{OLS} trained with input sets GFFG-q and GFFGA-q (The best prediction results obtained from NN_{OLS} are shown in bold colour)

Actual class	Average classification/misclassification rates for input set GFFG-q			Average classification/misclassification rates for input set GFFGA-q		
	Predicted class			Predicted class		
	Buy	Hold	Sell	Buy	Hold	Sell
Buy	**23.46%**	**76.54%**	**0.00%**	22.11%	77.89%	0.00%
Hold	**5.00%**	**88.74%**	**6.27%**	5.00%	87.48%	7.52%
Sell	**0.00%**	**79.79%**	**20.21%**	0.00%	79.79%	20.21%

Table 7.7: Average (over the six windows) classification rate /misclassification rate corresponding to the results obtained from the NN_{OLS} trained with input sets GFFG-sq and GFFGA-sq

Actual class	Average classification/misclassification rates for input set GFFG-sq			Average classification/misclassification rates for input set GFFGA-sq		
	Predicted class			Predicted class		
	Buy	Hold	Sell	Buy	Hold	Sell
Buy	21.16%	78.84%	0.00%	21.16%	78.84%	0.00%
Hold	5.00%	88.73%	6.27%	4.93%	88.42%	6.65%
Sell	0.00%	81.94%	18.06%	0.00%	81.94%	18.06%

and Table 7.7). Both types of networks did not produced serious misclassifications when trained with the above mentioned input sets.

The overall prediction performance of NN_{OLS} was poorer than that of the FNNs. However, NN_{OLS} always generated a single solution to the network parameters, while FNN produced 1000 solutions for 1000 trials which also required a longer computer time.

Comparison of Optimal Values of the Error Function of NN_{OLS} and FNNs

Table 7.8 compares the optimal values of the error function OLS generated by NN_{OLS} and the FNN. The optimal value corresponding to the FNN is the minimum of 1000 such values resulted from 1000 trials.

Table 7.8: Comparison of the Optimal Value of the Error Function Obtained by NN_{OLS} and FNN (The optimal value corresponding to FNN is the minimum value of 1000 trials)

Input set	NN_{OLS} /FNN	Window Number					
		1	2	3	4	5	6
GFFG	NN_{OLS}	2.347E-5	2.218E-5	1.971E-5	1.738E-5	1.108E-5	9.053E-6
	FNN	4.577E-5	3.669E-5	3.534E-5	3.417E-5	2.407E-5	1.557E-5
GFFGA	NN_{OLS}	2.347E-5	2.218E-5	1.971E-5	1.738E-5	1.108E-5	9.053E-6
	FNN	4.510E-5	3.608E-5	3.510E-5	3.322E-5	2.293E-5	1.523E-5
GFFG-q	NN_{OLS}	2.348E-5	2.228E-5	1.981E-5	1.746E-5	1.108E-5	9.095E-6
	FNN	4.522E-5	3.676E-5	3.480E-5	3.390E-5	2.385E-5	1.519E-5
GFFGA-q	NN_{OLS}	2.348E-5	2.220E-5	1.971E-5	1.746E-5	1.108E-5	9.111E-6
	FNN	4.458E-5	3.613E-5	3.458E-5	3.435E-5	2.397E-5	1.478E-5
GFFG-sq	NN_{OLS}	2.381E-5	2.235E-5	1.992E-5	1.772E-5	1.158E-5	9.398E-6
	FNN	4.755E-5	3.803E-5	3.927E-5	3.854E-5	2.590E-5	1.446E-5
GFFGA-sq	NN_{OLS}	2.387E-5	2.235E-5	2.003E-5	1.773E-5	1.116E-5	9.416E-6
	FNN	4.776E-5	3.823E-5	3.943E-5	3.871E-5	2.601E-5	1.654E-5

Irrespective of the window or the input set, the optimal value of the error function produced by the NN_{OLS} is lower than the minimum of the respective values corresponding FNN (Table 7.8). This indicates for that NN_{OLS} is capable in finding better solutions for the OLS error function.

7.4.2 Results Obtained from NN_{DLS}

Table 7.9 and Table 7.10 present the prediction accuracies of the results obtained from NN_DLS trained with input sets contain multiple inputs which are not quantified (GFFG and GFFGA) and multiple quantified input features (GFFG-q and GFFGA-q) , respectively. Those results related to the same network trained with input sets consist of single inputs with sum of the quantified input features (GFFG-sq and GFFGA-sq) are shown in Table 7.11.

Table 7.9: Average (over the six windows) classification rate /misclassification rate corresponding to the results obtained from the NN_{DLS} trained with input sets GFFG and GFFGA

Actual class	Average classification/misclassification rates for input set GFFG			Average classification/misclassification rates for input set GFFGA		
	Predicted class			Predicted class		
	Buy	Hold	Sell	Buy	Hold	Sell
Buy	27.32%	71.85%	0.83%	26.53%	72.64%	0.83%
Hold	5.81%	86.34%	7.84%	5.81%	86.34%	7.84%
Sell	1.39%	78.40%	20.21%	1.39%	78.40%	20.21%

NN_{DLS} made serious errors such as misclassification of buy signals to sell signals and vice-versa, when they were trained with input sets which consists of multiple (either quantified or un-quantified) input features (Table 7.9 and Table 7.10). Compared to the respective FNN counterparts, NN_{DLS} generated higher number of correct buy signals, but

Table 7.10: Average (over the six windows) classification rate /misclassification rate corresponding to the results obtained from the NN_{DLS} trained with input sets GFFG-q and GFFGA-q

Actual class	Average classification/misclassification rates for input set GFFG-q			Average classification/misclassification rates for input set GFFGA-q		
	Predicted class			Predicted class		
	Buy	Hold	Sell	Buy	Hold	Sell
Buy	23.54%	76.46%	0.00%	25.93%	74.07%	0.00%
Hold	4.97%	89.26%	5.77%	5.79%	87.24%	6.97%
Sell	1.39%	80.62%	17.99%	1.39%	79.51%	19.10%

Table 7.11: Average (over the six windows) classification rate /misclassification rate corresponding to the results obtained from the NN_{DLS} trained with input sets GFFG-sq and GFFGA-sq (The best prediction results obtained from NN_{DLS} are shown in bold colour)

Actual class	Average classification/misclassification rates for input set GFFG-sq			Average classification/misclassification rates for input set GFFGA-sq		
	Predicted class			Predicted class		
	Buy	Hold	Sell	Buy	Hold	Sell
Buy	21.23%	78.77%	0.00%	**22.10%**	**77.90%**	**0.00%**
Hold	5.39%	89.22%	5.39%	**4.97%**	**89.20%**	**5.83%**
Sell	0.00%	83.06%	16.94%	**0.00%**	**83.06%**	**16.94%**

a lower number of correct sell signals (Table 7.5 to Table 7.7 and Table 7.9 to Table 7.11). In overall, the performance of NN_{DLS} is poorer than that of NN_{OLS}.

7.4.3 Results Obtained from NN_{CC}

Table 7.12 and Table 7.13 show the average (over the six windows) values of the overall classification and misclassification rates associated with NN_{CC} when trained with input sets which contain multiple inputs which are not quantified (GFFG and GFFGA) and multiple quantified input features (GFFG-q and GFFGA-q), respectively. The corresponding results of the NN_{CC} trained with single inputs which represent the sum of the quantified input features (GFFG-sq and GFFGA-sq) are presented in Table 7.14.

Table 7.12: Average (over the six windows) classification rate /misclassification rate corresponding to the results obtained from the NN_{CC} trained with input sets GFFG and GFFGA (The best prediction results obtained from NN_{CC} are shown in bold colour)

Actual class	Average classification/misclassification rates for input set GFFG			Average classification/misclassification rates for input set GFFGA		
	Predicted class			Predicted class		
	Buy	Hold	Sell	Buy	Hold	Sell
Buy	**23.94%**	**76.06%**	**0.00%**	20.44%	78.73%	0.83%
Hold	**5.00%**	**89.59%**	**6.66%**	4.86%	88.45%	6.69%
Sell	**0.00%**	**77.71%**	**22.29%**	0.00%	79.79%	20.21%

Except in the case where the network was trained with the five input features which are not quantified (input set 4), NN_{CC} did not produce serious misclassifications (Table 7.12 to Table 7.14). In all other cases the performance of this algorithm was better than or equal to the respective cases of NN_{OLS}. Particularly, NN_{CC} showed an improvement in prediction accuracy compared to the respective NN_{OLS} algorithm, when trained with the four input features which are not quantified (input set GFFG; Table 7.5 and Table 7.12). Overall, this algorithm showed an improvement compared to NN_{DLS} (Table 7.9 to Table 7.11).

Table 7.13: Average (over the six windows) classification rate /misclassification rate corresponding to the results obtained from the NN_{CC} trained with input sets GFFG-q and GFFGA-q

Actual class	Average classification/misclassification rates for input set GFFG-q			Average classification/misclassification rates for input set GFFGA-q		
	Predicted class			Predicted class		
	Buy	Hold	Sell	Buy	Hold	Sell
Buy	21.68%	78.32%	0.00%	22.49%	77.51%	0.00%
Hold	4.58%	87.90%	7.52%	4.20%	88.22%	7.58%
Sell	0.00%	79.72%	20.28%	0.00%	79.72%	20.28%

Table 7.14: Average (over the six windows) classification rate /misclassification rate corresponding to the results obtained from the NN_{CC} trained with input sets GFFG-sq and GFFGA-sq

Actual class	Average classification/misclassification rates for input set GFFG-sq			Average classification/misclassification rates for input set GFFGA-sq		
	Predicted class			Predicted class		
	Buy	Hold	Sell	Buy	Hold	Sell
Buy	21.05%	78.95%	0.00%	22.44%	77.56%	0.00%
Hold	4.58%	89.57%	5.85%	5.40%	87.02%	7.58%
Sell	0.00%	81.94%	18.06%	0.00%	78.61%	21.39%

7.4.4 Results Obtained from NN_{TCC}

The average (over the six windows) Classification rate /Misclassification rate corresponding to the results obtained from the NN_{TCC} are shown in Table 7.15 to Table 7.17.

Table 7.15 corresponds to the input sets with multiple input features which are not quantified (input sets GFFG and GFFGA). The results corresponding to the input sets consist of multiple quantified input features and single inputs which represent the sum of the quantified input features are presented in Table 7.16 and Table 7.17, respectively.

Table 7.15: Average (over the six windows) classification rate /misclassification rate corresponding to the results obtained from the NN_{TCC} trained with input sets GFFG and GFFGA

Actual class	Average classification/misclassification rates for input set GFFG			Average classification/misclassification rates for input set GFFGA		
	Predicted class			Predicted class		
	Buy	Hold	Sell	Buy	Hold	Sell
Buy	30.89%	68.28%	0.83%	28.72%	71.28%	0.00%
Hold	5.03%	89.98%	4.99%	4.48%	88.99%	6.53%
Sell	1.39%	79.39%	19.24%	1.39%	82.57%	16.04%

Overall, the prediction performance of NN_{TCC} is poorer than that of NN_{OLS} and NN_{CC}, because, except for the case where four quantified input features used as inputs (input set GFFGA-q), it produced serious misclassifications (Table 7.15 to Table 7.17). However, when trained with the input set GFFGA-q, not only it produced the highest prediction accuracies for buy and sell signals among all algorithms considered, but also did not produce series misclassifications (Table 7.16).

Comparison of the Results Obtained from NN_{TCC} and the FNNs

Table 7.18 compares the best prediction results obtained from NN_{TCC} with those obtained from the FNNs. NN_{TCC} produced the best results when it was trained with the four inputs of the quantified relative returns of the Close prices of day t of the US and the European stock markets indices (input set GFFG-q) while the FNN gave the best results when it

Table 7.16: Average (over the six windows) classification rate /misclassification rate corresponding to the results obtained from the NN_{TCC} trained with input sets GFFG-q and GFFGA-q (The best prediction results obtained from NN_{TCC} are shown in bold colour)

Actual class	Average classification/misclassification rates for input set GFFG-q			Average classification/misclassification rates for input set GFFGA-q		
	Predicted class			Predicted class		
	Buy	Hold	Sell	Buy	Hold	Sell
Buy	**27.00%**	**73.00%**	**0.00%**	23.15%	76.85%	0.00%
Hold	**4.56%**	**89.22%**	**6.22%**	4.16%	88.28%	7.56%
Sell	**0.00%**	**75.49%**	**24.51%**	1.39%	80.42%	18.19%

Table 7.17: Average (over the six windows) classification rate /misclassification rate corresponding to the results obtained from the NN_{TCC} trained with input sets GFFG-sq and GFFGA-sq

Actual class	Average classification/misclassification rates for input set GFFG-sq			Average classification/misclassification rates for input set GFFGA-sq		
	Predicted class			Predicted class		
	Buy	Hold	Sell	Buy	Hold	Sell
Buy	27.04%	72.24%	0.72%	25.48%	74.52%	0.00%
Hold	6.27%	85.71%	8.02%	5.85%	86.51%	7.64%
Sell	1.11%	77.08%	21.81%	1.11%	75.97%	22.92%

was trained with the five inputs of quantified relative returns of the Close prices of day t of the US, the European and the AORD (input set GFFGA-q).

There is a slight improvement of prediction accuracy of buy signals when NN_{TCC} was employed. However, the prediction accuracy of sell signals corresponding to NN_{TCC} shows

Table 7.18: Average (over the six windows) classification rate /misclassification rate corresponding to the results obtained from the NN_{TCC} trained with input sets GFFG-q and the FNNs trained with input set GFFGA-q

Actual class	Average classification/misclassification rates for NN_{TCC} trained with input set GFFG-q			Average classification/misclassification rates for the FNN trained with input set GFFGA-q		
	Predicted class			Predicted class		
	Buy	Hold	Sell	Buy	Hold	Sell
Buy	27.00%	73.00%	0.00%	26.40%	73.60%	0.00%
Hold	4.56%	89.22%	6.22%	5.00%	87.86%	7.14%
Sell	0.00%	75.49%	24.51%	0.00%	79.80%	20.21%

a substantial increment when compared to respective value of the FNN. Misclassification of hold signals to buy/sell signals decreased when NN_{TCC} was used. Therefore, it can be suggested that better predictions can be obtained by using the algorithm NN_{TCC} than the FNN. Finally, it can be suggested that the attempt to improve the prediction accuracy was successful.

7.5 Results from the Trading Simulations

The trading simulations proposed in Section 6.4.1 were applied to the trading signals obtained by the four newly proposed neural network algorithms (Section 7.3). Table 7.19 shows the average (over the six windows) rates of returns obtained by performing the proposed trading simulation on the predictions obtained by the best network of the each algorithm considered.

According to Table 7.19, the trading signals produced by best network corresponding to NN_{TCC} gave the highest rate of return. The second highest rate of return relates

Table 7.19: Average (over the six windows) rate of return for the best predictions relating to each neural network algorithm (*The annual average rate of return relating to the benchmark simulation = 9.57%*)

Algorithm	Rate of return for test period	Annual rate of return
NN_{OLS}	7.16%	24.12%
NN_{DLS}	7.71%	25.97%
NN_{CC}	7.92%	26.68%
NN_{TCC}	8.27%	27.86%

to the best network corresponding to NN_{CC} followed by those of NN_{DLS} and NN_{OLS}. 28% of annual profit can be made by responding the trading signals predicted by the best network corresponding to NN_{TCC}. This value is also higher than the highest rate of return (25.9%; Table 6.4) related to the standard FNNs used in the previous chapter (Chapter 6).

Comparing the annual rate of returns from the two trading simulations, it can be suggested that traders can make more profits by responding the trading signals generated by the four algorithms considered in this study. The profits relating to NN_{TCC} are the highest.

7.6 Conclusions Derived from the Results Obtained by Network Training

The results obtained from the experiments show that the neural network algorithms based on the modified OLS error functions introduced by this study (7.9 and 7.10) produced better predictions for trading signals corresponding to day $(t+1)$ of the AORD. Among these two algorithms, the one based on (7.10) is the best.

This algorithm produced the best prediction results when the network consisted of one hidden layer with two neurons. The quantified relative returns of the Close prices of day t of the GSPC and the three European stock market indices were used as the input features. This network prevented serious misclassifications such as misclassification of buy signals to sell signals and vice-versa and also predicted trading signals with a higher degree of accuracy.

Proposed trading simulations suggest that a trader can gain substantially high (28%) annual return by responding to the trading signals produced by this best neural network algorithm.

These results also indicates the following:

- The quantified influences from the above mentioned four market indices can be used effectively to produced more accurate trading signals.
- The application of the global optimization algorithm, which is described in Section 4.2.2, to minimise the error functions was quite successful. Although, there is no guarantee that the solutions generated by the new algorithms are global solutions, they are much better than the best solutions (out of 1000 trials) obtained by the FNNs trained with the Resilient backpropagation training algorithm.

7.7 Summary

This chapter focused on developing new algorithms in order to produce better predictions for trading signals of the AORD. An attempt was made to develop new neural network algorithms by employing a global optimization technique to train the networks as well as introducing modified error functions. These error functions include the traditional ordinary least squares error function, the discounted least squares error function proposed by Refenes at el. [69] and two modified functions introduced by this study. Four new algorithms: (1) NN_{OLS}, (2) NN_{DLS}, (3) NN_{CC} and (4) NN_{TCC} were tested.

These algorithms were successful in finding an optimal value of the respective error functions. Results suggest that the algorithms based on the modified error functions introduced by this study (NN_{CC} and NN_{TCC}) showed better performance compared to the FNNs (used in Chapter 6), by producing prediction results with better accuracy, provided that they were trained with proper input features and the proper number of neurons were included in the hidden layer. NN_{TCC} which was based on error function described by Equation 7.10, produced the best results in predicting trading signals of the AORD. The other important matter that suggested by the results produced by NN_{TCC} is that the quantified intermarket influences on the AORD can be used effectively to predict trading signals.

The next and the final chapter provides the conclusions of the study together with suggestions for future research.

Chapter 8

Conclusions and Recommendations

This chapter includes the research findings and the key contributions this research has made to knowledge. The limitations of the study together with suggestions for further research are also presented.

8.1 Conclusions

The conclusions of the study can be summarised as below:

1. The Close price of day $(t-1)$ of the US S&P 500 index (GSPC) had the strongest influence on the Close price of day t of the Australian All Ordinary Index (AORD), during the whole study period (from 2nd July 1997 to 30th December 2005). The Close prices of day $(t-1)$ of the UK FTSE 100 Index (FTSE), French CAC 40 Index (FCHI), German DAX Index (GDAXI) and the AORD itself, showed a significant impact on the Close price of day t of the AORD, at different time periods. The Close prices of two or more days in the past of any markets considered did not show a substantial influence on the Close price of day t of the AORD. These results suggest the successfulness of the proposed technique for quantification of intermarket influences.

2. The quantified intermarket influences on the AORD can be effectively used to predict

the direction of the Close price of of day $(t+1)$ the AORD. The sum of the quantified relative return of the Close prices of day t of the influential markets are useful for this directional prediction. This matter supports the effectiveness of applying quantified intermarket influences for the directional prediction.

3. The quantified relative returns of the Close prices of the influential markets can effectively be used to predict the trading signals, buy, hold and sell, of the AORD. This is an indication for the usefulness of the quantified intermarket influences on AORD for predicting the trading signals of day $(t + 1)$ of the AORD.

4. The neural network algorithms, designed by incorporating the modified Ordinary Least Squares error functions, improved the prediction accuracy of trading signals. Among these algorithms, NN_{TCC} (Section 7.3) was the best. NN_{TCC} uses the modified error function for which an adjustment relating to the contribution from the historical data used for training the networks, and the penalisation of incorrectly classified trading signals were incorporated. This algorithm gave better prediction accuracy when trained with the quantified relative returns of the Close prices of the influential markets. Trading simulations demonstrated that this algorithm produced trading signals which are more profitable. These matters confirm that the quantified intermarket influences on the AORD can be effectively used to predict the trading signals of day $(t + 1)$ of the AORD.

The approach developed in this study, which involves quantification of intermarket influences and its applications for prediction of the direction of price level, and the trading signals, can be applied to perform similar predictions related to any given stock market index or stock index.

8.2 Comparison with Previous Studies

Incorporating intermarket influences for predictions relating to stock market indices is a very interesting aspect in finance. There are only a few studies (Section 2.5) which incorporated the possible influence from foreign stock markets when predicting a selected stock market index. However, no techniques were introduced to quantify intermarket influences (Section 2.4).

We developed a technique for quantifying intermarket influences from a selected set of potential influential (global) stock market indices on a given dependent market. We also investigated how the quantified intermarket influences can be used for prediction (directional prediction and prediction of trading signals).

This study employed a new measure (Section 5.2.2) which is more appropriate for evaluating the accuracy of the directional prediction, than the measure used in previous research [65, 82].

A few studies done in the past considered the prediction of three trading signals: buy, hold and sell (Section 2.2.2). Furthermore, literature does not provide evidence about any attempt in the past, to predict these three trading signals corresponding to the AORD.

Unlike many previous studies, this study aimed at predicting three trading signals. Consideration of these three classes (signals) resulted in an imbalance of data and this imbalance caused many classification algorithms (which are commonly used) to be less successful. Feedforward neural networks (FNN) provided better results compared to PNN and SVM. However, it is well known that the standard FNN provides solutions far from the global optimal solutions (Section 7.1). Therefore, we developed new neural network algorithms for predicting trading signals.

When developing the new algorithms, main concern was to modify the available error minimisation function in the literature (Section 7.2.1), in a way that made it suitable for the problem of interest: classification of trading signals into three classes, buy, hold and sell.

As mentioned earlier, literature (section 2.2.2) shows only a few studies (for example [11, 40, 44, 56]) which aimed at predicting the trading signals of the other international stock markets. Since, every stock market is different, and has its own unique 'personality' and unique position in the international economic systems [65], the comparison of the results of this study with those of other studies, is not appropriate.

This study is novel in aiming to carry out a formal quantification of intermarket influences from the world's major stock markets on the AORD and then predicting the direction of the Close price of the AORD as well as its trading signals, using the quantified intermarket influences as input features to the prediction models.

8.2.1 Contribution to the Knowledge

This study made the following contributions to the knowledge:

- Developed a new technique for quantifying intermarket influences from a set of potential influential stock markets on a given stock market.
- Quantified influence from major global stock markets on the AORD using this quantification technique.
- Identified how the quantified intermarket influences on the AORD can be incorporated for predicting the direction (up or down) of the Close price of day t as well as trading signals (buy, hold or sell) of the AORD.
- Developed neural network algorithms to predict the daily trading signals of a given stock market.
- The proposed prediction approach can be applied to do similar predictions related to any stock market.

8.3 Further Studies

In this study, the quantified relative return of the Close price of a given market is defined as the actual relative return of the Close price of this market multiplied by the respective quantification coefficient. We used the sum of the quantified relative returns of the Close prices of a set of influential markets as a single input feature as well as the quantified relative returns of the Close prices of these markets as separate input features, for predictions. There may be other alternative ways that the quantification coefficients (that is the strength of the influence) can be incorporated into predictive models. However, this may depend on the predictive models (or algorithms) that are applied to perform the predictions. The algorithms designed for time series predictions might be of special interest.

Another alternative approach for future research is that the consideration of a moving average (for example five day, 22 days, etc.), instead of the relative return of the Close prices of the stock markets, when quantifying the intermarket influences. Use of moving averages would be appropriate, since the moving average smooths the series removing the random fluctuations (noise), and therefore, results in higher correlation between series.

Traders' concerns may be to identify five trading signals: strong buy, buy, hold, sell, strong sell (instead of buy, hold and sell). Therefore, another direction of further research is to consider five trading signals for predictions and modify the error function (7.10) accordingly.

Appendices

Appendix A

Table A.1 to A.6 present the quantification coefficients corresponding to different market combinations, at time lag 1. In these tables, '-' indicates that a particular stock index is not included in the combination of interest. For example, the first row in Table A.1 shows the quantification coefficients relevant to the market combination which includes only the GSPC and the GDAXI while the last row in the same table shows those relevant to the market combination of the GSPC, FTSE, FCHI, HSI, and AORD. Also it is noteworthy that any one of the indices: STI, N225, SSEC, and TWII, did not include in any of the combinations considered. This is because that the Close price of these markets at time lag 1 did not show any significant impact on the Close price of the AORD.

Table A.1: Quantification coefficients at time lag 1 for different market combinations for the 1st training window

Stock Market Index						
GSPC	IXIC	FTSE	FCHI	GDAXI	HSI	AORD
0.9254	-	-	-	0.0746	-	-
0.6856	-	0.3144	-	-	-	-
0.9129	-	-	0.0871	-	-	-
0.7228	0.2772	-	-	-	-	-
0.9486	-	-	-	-	0.0514	-
0.6218	-	0.3309	-	0.0473	-	-
0.7584	-	-	0.1913	0.0503	-	-
0.6099	-	0.3374	0.0528	-	-	-
0.6856	0.0000	0.3144	-	-	-	-
0.9767	-	-	-	0.0122	0.0111	-
0.8431	0.0546	-	-	0.1023	-	-
0.5720	-	0.2905	0.1140	0.0233	-	-
0.5715	0.0090	0.3005	0.1190	-	-	-
0.5716	-	0.2961	0.1261	-	0.0061	-
0.6208	-	0.3327	-	0.0465	0.0000	-
0.6308	0.0000	0.3316	-	-	0.0376	-
0.5729	-	0.2882	0.1143	0.0246	0.0000	-
0.5310	0.0520	0.2879	0.1041	0.0249	-	-
0.5729	-	0.2882	0.1143	0.0246	0.0000	-
0.6857	-	0.3143	-	-	-	0.0000
0.9305	-	-	-	0.0288	-	0.0407
0.5387	-	0.3083	0.1118	-	-	0.0413
0.6377	-	0.3221	-	-	0.0402	0.0000
0.5621	-	0.2929	0.0997	0.0283	-	0.0170
0.6174	-	0.3425	0.0320	-	0.0000	0.0080

Table A.2: Quantification coefficients at time lag 1 for different market combinations for the 2nd training window

Stock Market Index						
GSPC	IXIC	FTSE	FCHI	GDAXI	HSI	AORD
0.7860	-	-	-	0.2140	-	-
0.7457	-	0.2543	-	-	-	-
0.7275	-	-	0.2725	-	-	-
0.8974	0.1026	-	-	-	-	-
0.9435	-	-	-	-	0.0565	-
0.5802	-	0.2589	-	0.1609	-	-
0.6150	-	-	0.0837	0.3013	-	-
0.5947	-	0.0857	0.3197	-	-	-
0.7309	0.0129	0.2562	-	-	-	-
0.7394	-	-	-	0.2086	0.0521	-
0.7720	0.0576	-	-	0.1704	-	-
0.6124	-	0.1825	0.0787	0.1264	-	-
0.6161	0.1011	0.0426	0.2402	-	-	-
0.7749	-	0.1492	0.0344	-	0.0414	-
0.5809	-	0.2557	-	0.1633	0.0000	-
0.7666	0.0000	0.1867	-	-	0.0467	-
0.6128	-	0.1658	0.0812	0.1402	0.0000	-
0.5886	0.0000	0.1886	0.0639	0.1590	-	-
0.6128	-	0.1658	0.0812	0.1402	0.0000	-
0.7356	-	0.2535	-	-	-	0.0109
0.7536	-	-	-	0.2178	-	0.0286
0.7618	-	0.1285	0.0670	-	-	0.0427
0.7562	-	0.1654	-	-	0.0301	0.0483
0.5780	-	0.1083	0.1266	0.1688	-	0.0183
0.5812	-	0.2564	0.1624	-	0.0000	0.0000

Table A.3: Quantification coefficients at time lag 1 for different market combinations for the 3rd training window

Stock Market Index						
GSPC	IXIC	FTSE	FCHI	GDAXI	HSI	AORD
0.8376	-	-	-	0.1624	-	-
0.7816	-	0.2184	-	-	-	-
0.7127	-	-	0.2873	-	-	-
0.9300	0.0701	-	-	-	-	-
0.9792	-	-	-	-	0.0208	-
0.7824	-	0.1575	-	0.0601	-	-
0.6624	-	-	0.3205	0.0171	-	-
0.7520	-	0.1565	0.0915	-	-	-
0.7827	0.0000	0.2173	-	-	-	-
0.8869	-	-	-	0.0982	0.0150	-
0.8809	0.0873	-	-	0.0318	-	-
0.7656	-	0.0931	0.1328	0.0085	-	-
0.7469	0.0151	0.1278	0.1102	-	-	-
0.7534	-	0.1544	0.0922	-	0.0000	-
0.7887	-	0.1547	-	0.0566	0.0000	-
0.7822	0.0000	0.2178	-	-	0.0000	-
0.7525	-	0.1545	0.0930	0.0000	0.0000	-
0.4864	0.0353	0.3308	0.1267	0.0207	-	-
0.7525	-	0.1545	0.0930	0.0000	0.0000	-
0.7209	-	0.1280	-	-	-	0.1511
0.6794	-	-	-	0.1446	-	0.1760
0.7960	-	0.0108	0.1316	-	-	0.0616
0.7191	-	0.1303	-	-	0.0000	0.1507
0.7369	-	0.0000	0.1730	0.0175	-	0.0726
0.7193	-	0.0489	0.1225	-	0.0033	0.1060

Table A.4: Quantification coefficients at time lag 1 for different market combinations for the 4th training window

Stock Market Index						
GSPC	IXIC	FTSE	FCHI	GDAXI	HSI	AORD
0.7427	-	-	-	0.2573	-	-
0.8062	-	0.1938	-	-	-	-
0.8703	-	-	0.1297	-	-	-
1.0000	0.0000	-	-	-	-	-
0.9872	-	-	-	-	0.0128	-
0.6454	-	0.0836	-	0.2710	-	-
0.8682	-	-	0.1318	0.0000	-	-
0.7932	-	0.0640	0.1429	-	-	-
0.7996	0.0062	0.1942	-	-	-	-
0.7289	-	-	-	0.2514	0.0197	-
0.7282	0.0209	-	-	0.2509	-	-
0.7946	-	0.0562	0.1492	0.0000	-	-
0.7946	0.0000	0.0593	0.1461	-	-	-
0.7778	-	0.0527	0.1491	-	0.0204	-
0.7374	-	0.0232	-	0.2393	0.0000	-
0.8016	0.0052	0.1932	-	-	0.0000	-
0.7795	-	0.0345	0.1687	0.0000	0.0172	-
0.8707	0.0000	0.0000	0.1293	0.0000	-	-
0.7795	-	0.0345	0.1687	0.0000	0.0172	-
0.8065	-	0.1935	-	-	-	0.0000
0.7434	-	-	-	0.2566	-	0.0000
0.7910	-	0.0640	0.1450	-	-	0.0000
0.8049	-	0.1951	-	-	0.0000	0.0000
0.7898	-	0.0700	0.1401	0.0000	-	0.0000
0.6419	-	0.0695	0.2736	-	0.0000	0.0150

Table A.5: Quantification coefficients at time lag 1 for different market combinations for the 5th training window

Stock Market Index						
GSPC	IXIC	FTSE	FCHI	GDAXI	HSI	AORD
0.8043	-	-	-	0.1957	-	-
0.6250	-	0.3750	-	-	-	-
0.8174	-	-	0.1826	-	-	-
0.7270	0.2730	-	-	-	-	-
0.9951	-	-	-	-	0.0049	-
0.5629	-	0.2202	-	0.2169	-	-
0.7039	-	-	0.0661	0.2300	-	-
0.6940	-	0.1369	0.1691	-	-	-
0.4565	0.2190	0.3245	-	-	-	-
0.7939	-	-	-	0.1866	0.0194	-
0.8024	0.0000	-	-	0.1976	-	-
0.5572	-	0.1720	0.0346	0.2362	-	-
0.4307	0.1831	0.2057	0.1805	-	-	-
0.6045	-	0.0922	0.2359	-	0.0674	-
0.5656	-	0.2184	-	0.2160	0.0000	-
0.7492	0.0203	0.1122	-	-	0.1183	-
0.5680	-	0.2115	0.0000	0.2204	0.0000	-
0.4426	0.1870	0.1997	0.1510	0.0197	-	-
0.5680	-	0.2115	0.0000	0.2204	0.0000	-
0.6662	-	0.1885	-	-	-	0.1452
0.7259	-	-	-	0.2556	-	0.0185
0.7240	-	0.0683	0.1378	-	-	0.0700
0.6664	-	0.1887	-	-	0.0000	0.1449
0.5569	-	0.1699	0.0394	0.2338	-	0.0000
0.5662	-	0.2175	0.2163	-	0.0000	0.0000

Table A.6: Quantification coefficients at time lag 1 for different market combinations for the 6th training window

Stock Market Index						
GSPC	IXIC	FTSE	FCHI	GDAXI	HSI	AORD
0.7609	-	-	-	0.2391	-	-
0.6668	-	0.3332	-	-	-	-
0.7833	-	-	0.2167	-	-	-
0.7834	0.2166	-	-	-	-	-
0.9830	-	-	-	-	0.0170	-
0.7219	-	0.2016	-	0.0765	-	-
0.6781	-	-	0.1063	0.2156	-	-
0.7086	-	0.1834	0.1080	-	-	-
0.5497	0.1456	0.3047	-	-	-	-
0.7589	-	-	-	0.2410	0.0000	-
0.7428	0.0226	-	-	0.2346	-	-
0.6658	-	0.0583	0.0795	0.1964	-	-
0.6221	0.0450	0.3160	0.0169	-	-	-
0.7069	-	0.1844	0.1087	-	0.0000	-
0.5904	-	0.2040	-	0.1352	0.0704	-
0.4592	0.2128	0.2393	-	-	0.0887	-
0.6628	-	0.0522	0.0859	0.1926	0.0065	-
0.5473	0.1200	0.3163	0.0163	0.0000	-	-
0.6628	-	0.0522	0.0859	0.1926	0.0065	-
0.6862	-	0.3013	-	-	-	0.0125
0.7583	-	-	-	0.2417	-	0.0000
0.7077	-	0.1818	0.1105	-	-	0.0000
0.6164	-	0.1985	-	-	0.0394	0.1456
0.6592	-	0.0431	0.0900	0.1957	-	0.0120
0.5945	-	0.2057	0.1301	-	0.0697	0.0000

Bibliography

[1] Akbani, R., Kwek, S. & Japkowicz, N. (2004). Applying Support Vector Machines to Imbalanced Datasets. *Proceedings of the 15th European Conference on Machine Learning (ECML'04)*, 33-50, Springer-Verlag.

[2] Alexander, C. (2001). *Market Models: A Guide to Financial Data Analysis*. John Wiley & Sons.

[3] Amaral, L. A. N., Plerou, V., Gopikrishnan, P., Meyer, M. & Stanley, E. (2000). The Distribution of Returns of Stock Prices. *International Journal of Theoretical and Applied Finance*, 3, 365-369.

[4] Becker, K. G., Finnerty, J. E., & Gupta, M. (1990). The Intertemporal Relation Between the U. S. and Japanese Stock markets. *The Journal of Finance*, XLV, 1297-1306.

[5] Bhattacharyya M. & Banerjee, A. (2004). Integration of Global Capital Markets: An Empirical Exploration. *International Journal of Theoretical and Applied Finance*, 7, 385-405.

[6] Burges, C. J. C. (1998). A tutorial on Support Vector Machines for Pattern Recognition. *Data Mining and Knowledge Discovery*, 2, 121-167.

[7] Caldwell, R. B. (1995). Performances Matrics for Neural Network-based Trading System Development. *NeuroVe$t Journal*, 3, 22-26.

[8] Cao, L. & Tay, F. E. H. (2001). Financial forecasting using support vector machines. *Neural Computing and Applications*, 10, 184-192.

[9] Chapman, A. J. (1994). Stock market trading systems through neural networks: developing a model. *International Journal of Applied Expert Systems*, 2, 88-100.

[10] Chawla, N. V., Bowyer, K. W., Hall, L. O. & Kegelmeyer, W. P. (2002). SMOTE: Synthetic Minority Over-Sampling Technique. *Journal of Artificial Intelligence Research*, 16, 321-357.

[11] Chen, A. S., Leung, M. K. & Daouk, H. (2003). Application of neural networks to an emerging financial market: forecasting and trading the Taiwan Stock Index. *Computers and Operations Research*, 30, 901-923.

[12] Chenoweth, T., & Obradovic, Z. (1995). A multi-component approach to stock market predictions. *Proceedings of the Third international conference on artificial intelligence applications on Wall Street*, 74-79.

[13] Chenoweth, T., Obradovic Z. & Stephenlee, S. (1996). Embedding Technical Analysis into Neural Network Based Trading Systems. *Applied Artificial Intelligence*, 10, 523-541.

[14] Demuth, H., Beale, M. & Hagan, M. (2006). *Neural Network Toolbox User's Guide*, The Math Works Inc..

[15] Dorsey, R & Sexton, R. (1998). The Use of Parsimonious Neural Networks for Forecasting Financial Time Series. *Journal of Computational Intlligence in Finance*, 6, 24-31.

[16] Egeli, B., Ozturan, M. & Badur, B. (2003). Stock Market Prediction Using Artificial Neural Networks. *Proceeding of the Hawaii International Conference on Business*, Honolulu, Hawaii, USA.

[17] Elder, A. (1993), *Trading for a Living*, Traders Press, Inc..

[18] Eun C. S. & Shim, S. (1989). International Transmission of Stock Market Movements. *The Journal of Financial and Quantitative Analysis*, 24, 241-256.

[19] Fama, E. F. (1965). The Behaviour of Stock-Market Prices. *The Journal of Business*, 38, 34-105.

[20] Fama, E. F. (1970). Efficient Capital Markets: A Review of Thoery and Empirical Work. *Journal of Finance*, 25, 383-417.

[21] Fernando, F. R, Christian, G. M. & Simon, S. R. (2000). On the profitability of technical trading rules based on artificial neural networks: Evidence from the Madrid stock market. *Economics Letters*, 69, 89-94.

[22] Gately, E. (1996). *Neural Networks for Financial Forecasting*. New York: John Wiley & Sons, Inc.

[23] Gencay, R. & Stengos, T. (1998). Moving Average Rules, Volume and the Predictability of Security Returns with Feedforward Networks. *Journal of Forecasting*, 17, 401-414.

[24] Gopikrishnan, P., Plerou, V., Amaral, L. A. N., Meyer, M. & Stanley, E. (1999). Scaling of the distribution of fluctuations of financial market indices. *Physical Review E*, 60, 5305-5316.

[25] Gultekin, M. N., Gultekin, N. B., & Penati, A. (1989). Capital Controls and International Capital Market Segmentation: The Evidence from the Japanese and American Stock Markets. *The Journal of Finance*, XLIV(4), 849-869.

[26] Huang, W., Nakamori, Y. & Wang, S. Y. (2005). Forecasting stock market movement direction with support vector machine. *Computers and Operations Research*, 32,1-10.

[27] Imam, T., Ting, K. M. & Kamruzzaman, J. (2006). z-SVM: An SVM for Improved Classification of Imabalanced Data. *Proceedings of the 19th Australian Joint Conference on Artificial Intelligence 2006*, Springer.

[28] Joachims, T. (1998). Text Categorization with Support Vector Machines: Learning with Many Relevant Features. *Proceedings of the European Conference on Machine Learning*, Springer.

[29] Joachims, T. (1999). Making Large-Scale SVM Learning Practical. In B. Schölkopf and C. Burges and A. Smola (eds.), *Advances in Kernel Methods - Support Vector Learning*, MIT Press. `http://svmlight.joachims.org/`

[30] Joachims, T. (2000). Estimating the Generalization Performance of a SVM Efficiently. *Proceedings of the International Conference on Machine Learning*, Morgan Kaufman.

[31] Joachims, T. (2002). *Learning to Classify Text Using Support Vector Machines*, Dissertation, Kluwer.

[32] Jordanov, I. (2002). Neural Network Training and Stochastic Global Optimization. *Proceedings of the 9th International Conference on Neural Information Processing (ICONIP'02)*, 1, 488-492.

[33] Kaastra, I. & Boyd, M. (1996). Designing a neural network for forecasting financial and economic time series. *Neurocomputing*, 10, 215-236.

[34] Kim, K-J. (2003). Financial time series forecasting using support vector machines. *Neurocomputing*, 55, 307-319.

[35] Kim, S. H. & Chun S. H. (1998). Graded forecasting using an array of bipolar predictions: Application of probabilistic neural networks to a stock market index. *International Journal of Forecasting*, 14, 323-337.

[36] Kolarik, T., & Rudorfer, G. (1994). Time Series Forecasting Using Neural Networks. *Proceedings of the International Conference on APL: the language and its application*, 86-94.

[37] Kolda, T. G., Lewis R. M. & Torczon, V. (2003). Optimization by Direct Search: New Perspectives on Some Classical and Modern Methods. *SIAM Review*, 45, 385-482.

[38] Kohara, K., Ishikawa, T., Fukuhara, Y., & Nakamura, Y. (1997). Stock Price Prediction Using Prior Knowledge and Neural Networks. *Intelligent Systems in Accounting, Finance and Management*, 6, 11-22.

[39] Koubor, S., Ugon, J., Mammadov, M. A., Rubinov, A. M. & Kruger, A. (2006). Coverage in WLAN: Optimization Model and Algorithm, *Proceeding of the First IEEE International Conference on Wireless Broadband and Ultra Wideband Communications*, Sydney, Australia.

[40] Kuo, R. J. (1998). A Decision Support System for the Stock Market Through Integration of Fuzzy Neural Networks and Fuzzy Delphi. *Applied Artificial Intelligence*, 12, 501-520.

[41] LaBeau, C. & Lucas, D. (1992). *Computer Analysis of the Futures Market*, Traders Press, Inc..

[42] Lam, K. & Lam, K. C. (200). Forecasting for the Generation of TradingSignals in Financial Markets. *Journal of Forecasting*, 19, 39-52.

[43] Lawrence, R. (1997). Using Neural Networks to Forecast Stock Market Prices. `http://people.ok.ubc.ca/rlawrenc/research/Papers/nn.pdf`

[44] Leung, M. T., Daouk, H. & Chen, A. S. (2000). Forecasting stock indices: a comparison of classification and level estimation models. *International Journal of Forecasting*, 16, 173-190.

[45] Mahfoud S. & Mani, G. (1996). Financial Forecasting Using Genetic Algorithms. *Applied Artificial Intelligence*, 10, 543-565.

[46] Malkiel, B. G. (1985). *A Random Walk Down Wall Street*. W. W. Nortan & Company, New York.

[47] Mammadov, M. A. (2004). A new global optimization algorithm based on dynamical systems approach. In A. Rubinov & M. Sniedovich (Eds.), *Proceedings of The Sixth International Conference on Optimization: Techniques and Applications (ICOTA6)*, Ballarat, Australia.

[48] Mammadov, M. A., Rubinov A. M. & Yearwood, J. (2005). Dynamical systems described by relational elasticities with applications to global optimization. In V. Jeyakumar & A. Rubinov (Eds.) *Continuous Optimisation: Current Trends and Applications*, Springer, 365-387.

[49] Mangarasian, O. L. (1969). *Nonlinear Programming*, McGraw Hill, New York.

[50] Mantegna, R. N. & Stanley, H. E. (1995). Scaling behaviour in the dynamics of an economic index. *Letters to Nature*, 376, 46-49.

[51] Margarita, S. (1992). Genetic Neural Networks for Financial Markets: Some Results. In B. Nuemann (Eds.) *10th European Conference on Atificial Intelligence (ECL 92)*, John Wiley & Sons Ltd. 211-213.

[52] Mehrotra, K., Mohan, C. K., & Ranka, S. (1997). *Elements of Artificial Neural Networks*, Massachusetts: MIT Press.

[53] Mendelsohn, L. B. (2000). *Trend Forecating with Technical Analysis*, Market Technologies Corporation.

[54] Minghu, J., Xiaoyan, Z., Baozong, Y., Xiaofang, T., Biqin, L., Qiuqi, R. & Mingyan, J. (2000). A fast hybrid algorithm of global optimization for feedforward neural networks. *Proceedings of the 5th International Conference on Signal Process-*

ing (WCCC-ICSP 2000), Beijing, China, 3, 1609-1612. `http://ieeexplore.ieee.org/xpl`

[55] Mittoo, U. (1992). Additional Evidence on Integration in the Canadian Stock Market. *The Journal of Finance*, XLVII(5), 2035-2054.

[56] Mizuno, H., Kosaka, M. & Yajima, H. (1998). Application of Neural Network to Technical Analysis of Stock Market Prediction. *Studies in Information and Control*, 7, 111-120.

[57] Morik, K., Brockhausen, P. & Joachims, T. (1999). Combining statistical learninh with a knowledge-based approach - A case study in intensive care monitoring. *Proceedings of the 16th International Conference on Machine Learning (ICML-99).* `http://svmlight.joachims.org/`

[58] Murphy, J. J. (1991). *Intermarket Technical Analysis*, John Wiley and Sons.

[59] Murphy, J. J. (1999). Technical Analysis of the Financial Markets. New York Institute of Finance, New York.

[60] Murphy, J. J. (2004). *Intermarket Analysis: Profiting from Global Market Relationships*. John Wiley and Sons.

[61] Nguyen, D. & Widrow, B. (1990). Improving the learning speed of 2-layer neural networks by choosing initial values of the adaptive weights. *Proceedings of the Internationsl Joint Conference on Neural Networks (IJCNN)*, 3, 21-26.

[62] O'Connor, M., Remus, W. & Griggs, K. (1997). Going up-going down: How good are people at forecasting trends and changes in trend?. *Journal of Forecasting*, 16, 165-176.

[63] Olson D. & Mossaman, C. (2001). Cross-correlation and Predictability of Stock Returns., *Journal of Forecasting*, 20, 145-160.

[64] Olson D. & Mossaman, C. (2003). Neural network forecasts of Canadian stock returns using accounting ratios. *International Journal of Forecasting*, 19, 453-465.

[65] Pan, H., Tilakaratne, C., and Yearwood, J. (2005). Predicting the Australian Stock Market Index Using Neural networks Exploiting Dynamical Swings and Intermarket Influences. *Journal of Research and Practice in Information Technology*, 37(1), 43-55. `https://www.acs.org.au/jrpit/JRPIT_Volumes.html`

[66] Poddig, T. & Rehkugler, H. (1995). A 'world' model of integrated financial markets using artificial neural networks. *Neurocomputing*, 10, 251-273.

[67] Qi, M. (1999). Nonlinear Predictability of Stock Returns Using Financial and Economic Variables. *Journal of Business & Economic Statistics* 17, 419-429.

[68] Qi M. & Maddala, G. S. (1999). Economic factors and the stock market: A new perspective. *Journal of Forecasting*, 18, 151-166.

[69] Refenes, A. N., Bentz, Y., Bunn, D. W., Burgess, A. N., & Zapranis, A. D. (1995). Financial Time Series Modelling with Discounted Least Squares Backpropagation. *Neurocomputing*, 14, 123-138.

[70] Refenes, N. A., Zapranis, A. and Francis, G. (1994). Stock performance modelling using neural networks: A Comparative Study with Regression Models. *Neural Networks*, 7, 375-388.

[71] Riedmiller, M. & Braun, H. (1993). A direct adaptive method for faster backpropagation learning: The RPROP algorithm, *Proceedings of the IEEE International Conference on Neural Networks*, California, USA.

[72] Rose-Pehrsson, S. L., Hart, S. J., Street, T. T., Tatem, P. A., Williams, F. W., Hammond, M. H., Gottuk, D. T., Wright, M. T. & Wong, J. T. (2001). Real-Time Probabilistic Neural Network Performance and Optimization for Fire Ditection and

Nuisance Alarm Rejection, In K. Beall, W. Grosshandler and H. Luck (Eds.) *Proc-ceding of the 12th International Conference on Automatic Fire Detection*, Maryland, USA.

[73] Ruggiero (Jr.), M. A. (2001). Rolling Your Own Intermarket Index. *Futures: News, Analysis and Strategies for Futures, Options and Derivative Traders*, 30(1), 56-59.

[74] Saad, E. W., Prokhorov, D. V. & Wunsch, D. C. (1998). Comparative study of stock trend prediction using time delay, recurrent and probabilistic neural networks. *IEEE Transactions on Neural Networks*, 9, 1456-1470.

[75] Safer, A. M. (2003). A comparision of two data mining techniques to predict abnormal stock market returns. *Intelligent Data Analysis*, 7, 3-13.

[76] Siegel S. & Castellan, N. J. (Jr.) (1988). *Nonparametric Statistics for the Behavioral Sciences*, McGraw-Hill, Singapore.

[77] Specht, D. (1990). Probabilistic Neural Networks. *Neural Networks*, 3, 109-118.

[78] Tay, F. E. H. & Cao, L. (2001). Application of support vector machines in financial time series forecasting. *Omega*, 29, 309-317.

[79] Tay, F. E. H. & Cao, L. (2002). Modified support vector machines in financial time series forecasting. *Neurocomputing*, 48, 847-861.

[80] Taylor, M. P. & Tonks, I. (1989). The Internationalisation of Stock Markets and the Abolition of U.K. Exchange Control. *Review of Economics and Statistics*, 71, 332-336.

[81] Thawornwong, S. & Enke, D. (2004). *Forecasting stock returns with artificial neural networks.* In G. P. Zhang, editor, Neural Networks in Business Forecasting, chapter 3, Idea Group Publishing.

[82] Tilakaratne, C. D. (2004). *A Neural Network Approach for Predicting the Direction of the Australian Stock Market Index*, Master thesis, University of Ballarat, Australia . `http://digthesis.ballarat.edu.au/adt/public/adt-ADT20051104.145850/`

[83] Tilakaratne, C. D. (2006). A Study of Intermarket Influence on the Australian All Ordinary Index at Different Time Periods. *Proceedings of the 2nd Australian Business and Behavioural Sciences Association (ABBSA) International Conference*, Adeliade, Australia.

[84] Tilakaratne, C. D., Mammadov, M. A. & Hurst, C. P. (2006). Quantification of Intermarket Influence Based on the Global Optimization and Its Application for Stock Market Prediction. In K-L. Ong, K. Smith-Miles, V. Lee & W-K. Ng (Eds.), *Proceedings of the International Workshop on Integrating AI and Data Mining (AIDM'06)*, Hobart, Tasmania, Australia. `http://doi.ieeecomputersociety.org/10.1109/AIDM.2006.14`

[85] Tilakaratne, C. D., Morris, S. A., Mammadov, M. A. & Hurst, C. P. (2007). Quantification of Intermarket Influence on the Australian All Ordinary Index Based on Optimization Techniques. In W. Read (Eds.), *Proceedings of the 13th Biennial Computational Techniques and Applications Conference (CTAC'06)*, *ANZIAM Journal*, 48, C104-C118.

[86] Tilakaratne, C. D., Morris, S. A., Mammadov, M. A. & Hurst, C. P. (2007). Predicting Stock Market Index Trading Signals Using Neural Networks. *Proceedings of the 14th Annual Global Finance Conference (GFC2007)*, Melbourne, Australia.

[87] Toh, K. A., Lu, J. & Yau, W. Y. (2001). Global Feedforward Neural Network Learning for Classification and Regression. *Proceedings of the Third International Workshop on Energy Minimization Methods in Computer Vision and Pattern Recognition (EMMCVPR 2001)*, Shophia Antipolis, France.

[88] Vanstone, B. (2006). Trading in the Australian stockmarket using artificial neural networks, PhD Thesis, ePublications@Bond University, School of Information Technology. `http://epublications.bond.edu.au/theses/vanstone/`

[89] Vapnik, V. N. (1998). *Statistical Learning Theory*, John Wiley and Sons, New York.

[90] Veropoulos, K., Campbell, C. & Cristianini, N. (1999). Controlling the sensivity of support vector machines. *Proceedings of the International Joint Conference on AI*, 55-60.

[91] Weiss, S. M., & Kulikowski, C. A. (1991). *Computer systems that learn.* Morgan Kaufmann, California.

[92] Wasserman, P. (1993). *Advanced Methods in Neural Computing*, Van Nostrand Reinhold, New York.

[93] White, H. (1988). Economic Prediction Using Neural networks: The Case of IBM Daily Stock Returns. *Proceedings of the Second Annual IEEE Conference on Neural Networks*, IEEE Press, 451-458.

[94] Witten, I. H., Frank, E. (2000). *Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations*, Morgan Kaufmann Publishers, San Francisco, CA.

[95] Wood, D. & Dasgupta, B. (1996). Classifying Trend Movements in the MSCI U.S.A. Capital Market Index - A Comparison of Regression, ARIMA and Neural Network Methods. *Computers & Operations Research*, 23, 611-622.

[96] Wu, C. & Su, Y. (1998). Dynamic Relations among International Stock Markets. *International Review of Economic and Finance*, 7, 63-84.

[97] Wu, G. & Chang, E. Y. (2005). KBA: Kernel Boundary Alignment Considering Imbalanced Data Distribution. *IEEE Transactions on Knowledge and Data Engineering*, 17, 786-795.

[98] Wu, Y. & Zhang, H. (1997). Forward Premiums as Unbiased Predictors of Future Currency Depreciation: A Non-parametric Analysis. *Journal of International Money and Finance*, 16, 609-623.

[99] Yang, J., Khan M. M. & Pointer, L. (2003). Increasing Integration Between the United States and Other International Stock Markets?; A Recursive Cointegration Analysis. *Emerging Markets Finance and Trade*, 39, 39-53.

[100] Yao, J., Tan, C. H. & Poh, H. L. (1999). Neural Networks for Technical Analysis: A Study on KLCI. *International Journal of Theoretical and Applied Finance*, 2, 221-241.

[101] Yao, J. & Tan, C. L. (2000). Time Dependent Directional Profit Model for Financial Time Series Forecasting. *Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks (IJCNN'00)*. Como, Italy.

[102] Yao, J. & Tan, C. L. (2001). A Study on Training Criteria for Financial Time Series Forecasting. *Proceedings of the International Conference on Neural Information Processing*. Shanghai, China.

[103] Ye, H. & Lin, Z. (2003). Global optimization of neural network weights using subenergy tunneling function and ripple search. *Proceedings of the 2003 International Symposium on Circuits and Systems: ISCAS'03*, 5, 725-728. `http://ieeexplore.ieee.org/xpl`

[104] Zemke, S. (1999). Nonlinear index prediction. *Physica A*, 269, 177-183.

[105] `http://en.wikipedia.org/wiki/Support_vector_machine`