Hindawi Publishing Corporation
Journal of Applied Mathematics and Decision Sciences
Volume 2009, Article ID 125308, 22 pages
doi:10.1155/2009/125308

Research Article

Modified Neural Network Algorithms for Predicting Trading Signals of Stock Market Indices

C. D. Tilakaratne,[1] M. A. Mammadov,[2] and S. A. Morris[2]

[1] *Department of Statistics, University of Colombo, P.O. Box 1490, Colombo 3, Sri Lanka*
[2] *Graduate School of Information Technology and Mathematical Sciences, University of Ballarat, P.O. Box 663, Ballarat, Victoria 3353, Australia*

Correspondence should be addressed to C. D. Tilakaratne, cdt@stat.cmb.ac.lk

Received 29 November 2008; Revised 17 February 2009; Accepted 8 April 2009

Recommended by Lean Yu

The aim of this paper is to present modified neural network algorithms to predict whether it is best to buy, hold, or sell shares (trading signals) of stock market indices. Most commonly used classification techniques are not successful in predicting trading signals when the distribution of the actual trading signals, among these three classes, is imbalanced. The modified network algorithms are based on the structure of feedforward neural networks and a modified Ordinary Least Squares (OLSs) error function. An adjustment relating to the contribution from the historical data used for training the networks and penalisation of incorrectly classified trading signals were accounted for, when modifying the OLS function. A global optimization algorithm was employed to train these networks. These algorithms were employed to predict the trading signals of the Australian All Ordinary Index. The algorithms with the modified error functions introduced by this study produced better predictions.

1. Introduction

A number of previous studies have attempted to predict the price levels of stock market indices [1–4]. However, in the last few decades, there have been a growing number of studies attempting to predict the direction or the trend movements of financial market indices [5–11]. 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 [10]. Leung et al. [12] also found that the classification models based on the direction of stock return outperform those based on the level of stock return in terms of both predictability and profitability.

The most commonly used techniques to predict the trading signals of stock market indices are feedforward neural networks (FNNs) [9, 11, 13], probabilistic neural networks (PNNs) [7, 12], and support vector machines (SVMs) [5, 6]. FNN outputs the value of the stock market index (or a derivative), and subsequently this value is classified into classes (or direction). Unlike FNN, PNN and SVM directly output the corresponding class.

Almost all of the above mentioned studies considered only two classes: the upward and the downward trends of the stock market movement, which were considered as buy and sell signals [5–7, 9, 11]. It was noticed that the time series data used for these studies are approximately equally distributied among these two classes.

In practice, the traders do not participate in trading (either buy or sell shares) if there is no substantial change in the price level. Instead of buying/selling, they will hold the money/shares in hand. In such a case it is important to consider the additional class which represents a hold signal. For instance, the following criterion can be applied to define three trading signals, buy, hold, and sell.

Criterion A.

$$\begin{aligned} &\textit{buy} \quad \textit{if} \;\; Y(t+1) \geq l_u, \\ &\textit{hold} \;\; \textit{if} \;\;\; l_l \; < \; Y(t+1) < l_u, \\ &\textit{sell} \;\; \textit{if} \;\; Y(t+1) \leq l_l, \end{aligned} \tag{1.1}$$

where $Y(t+1)$ is the relative return of the Close price of day $(t+1)$ of the stock market index of interest, while l_l and l_u are thresholds.

The values of l_l and l_u depend on the traders' choice. There is no standard criterion found in the literature how to decide the values of l_l and l_u , and these 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 sensitivity analysis. The Australian All Ordinary Index (AORD) was selected as the target stock market index for this study. We experimented different pairs of values for l_l and l_u [14]. 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 fixed.

By examining the data distribution (during the study period, the minimum, maximum, and average for the relative returns of the Close price of the AORD are −0.0687, 0.0573, and 0.0003, resp.), we chose $l_u = -l_l = 0.005$ for this study, 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. It is unlikely that a change in the values of l_l and l_u would make a qualitative change in the prediction results obtained.

According to Criterion A with $l_u = -l_l = 0.005$, one cannot expect a balanced distribution of data among the three classes (trading signals) because more data falls into the hold class while less data falls into the other two classes.

Due to the imbalance of data, the most classification techniques such as SVM and PNN produce less precise results [15–17]. FNN can be identified as a suitable alternative technique for classification when the data to be studied has an imbalanced distribution. However, a standard FNN itself shows some disadvantages: (a) use of local optimization methods which do not guarantee a deep local optimal solution; (b) because of (a), FNN needs to be trained many times with different initial weights and biases (multiple training results in more than one solution and having many solutions for network parameters prevent getting a clear picture about the influence of input variables); (c) use of the ordinary least squares (OLS; see (2.1)) as an error function to be minimised may not be suitable for classification problems.

To overcome the problem of being stuck in a local minimum, finding a global solution to the error minimisation function is required. Several past studies attempted to find global solutions for the parameters of the FNNs, by developing new algorithms (e.g., [18–21]). Minghu et al. [19] proposed a hybrid algorithm of global optimization of dynamic learning rate for FNNs, and this algorithm shown to have global convergence for error backpropagation multilayer FNNs (MLFNNs). The study done by Ye and Lin [21] presented a new approach to supervised training of weights in MLFNNs. Their algorithm is based on a "subenergy tunneling function" to reject searching in unpromising regions and a "ripple-like" global search to avoid local minima. Jordanov [18] proposed an algorithm which makes use of a stochastic optimization technique based on the so-called low-discrepancy sequences to trained FNNs. Toh et al. [20] also proposed an iterative algorithm for global FNN learning.

This study aims at modifying neural network algorithms to predict whether it is best buy, hold, or sell the shares (trading signals) of a given stock market index. This trading system is designed for short-term traders to trade under normal conditions. It assumes stock market behaviour is normal and does not take unexceptional conditions such as bottlenecks into consideration.

When modifying algorithms, two matters were taken into account: (1) using a global optimization algorithm for network training and (2) modifying the ordinary least squares error function. By using a global optimization algorithm for network training, this study expected to find deep solutions to the error function. Also this study attempted to modify the OLS error function in a way suitable for the classification problem of interest.

Many previous studies [5–7, 9, 11] have used technical indicators of the local markets or economical variables to predict the stock market time series. The other novel idea of this study is the incorporation of the intermarket influence [22, 23] to predict the trading signals.

The organisation of the paper is as follows. Section 2 explains the modification of neural network algorithms. Section 3 describes the network training, quantification of intermarket influence, and the measures of evaluating the performance of the algorithms. Section 4 presents the results obtained from the proposed algorithms together with their interpretations. This section also compares the performance of the modified neural network algorithms with that of the standard FNN algorithm. The last section is the conclusion of the study.

2. Modified Neural Network Algorithms

In this paper, we used modified neural network algorithms for forecasting the trading signals of stock market indices. We used the standard FNN algorithm as the basis of these modified algorithms.

A standard FNN is a fully connected network with every node in the lower layer linked to every node in the next higher layer. These linkages are attached with some weights, $w = (w_1, \ldots, w_M)$, where M is the number of all possible linkages. Given weight, w, the network produces an output for each input vector. The output corresponding to the ith input vector will be denoted by $o_i \equiv o_i(w)$.

FNNs adopt the backpropagation learning that finds optimal weights w by minimising an error between the network outputs and given targets [24]. The most commonly used error function is the Ordinary Least Squares function (OLS):

$$E_{\text{OLS}} = \frac{1}{N}\sum_{i=1}^{N}(a_i - o_i)^2, \tag{2.1}$$

where N is the total number of observations in the training set, while a_i and o_i are the target and the output corresponding to the ith observation in the training set.

2.1. Alternative Error Functions

As described in the Introduction (see Section 1), 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 accuracy of predictions [24, 25]. Having this idea in mind, some past studies aimed to modify the error function associated with the FNNs (e.g., [24–27]). These studies incorporated factors which represent the direction of the prediction (e.g., [24–26]) and the contribution from the historical data that used as inputs (e.g., [24, 25, 27]).

The functions proposed in [24–26] 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 viceversa.

Caldwell [26] proposed the Weighted Directional Symmetry (WDS) function which is given as follows:

$$f_{\text{WDS}}(i) = \frac{100}{N}\sum_{i=1}^{N} w_{ds}(i)\left|a_i - o_i\right|, \tag{2.2}$$

where

$$w_{ds}(i) = \begin{cases} 1.5 & \text{if } (a_i - a_{i-1})(o_i - o_{i-1}) \leq 0, \\ 0.5, & \text{otherwise,} \end{cases} \tag{2.3}$$

and N is the total number of observations.

Yao and Tan [24, 25] argued that the weight associated with f_{WDS} (i.e., $w_{ds}(i)$) should be heavily adjusted if a wrong direction is predicted for a larger change, while it should be slightly adjusted if a wrong direction is predicted for a smaller change and so on. Based on this argument, they proposed the Directional Profit adjustment factor:

$$f_{\text{DP}}(i) = \begin{cases} c_1 & \text{if } (\Delta a_i \times \Delta o_i) > 0\,,\ \Delta a_i \leq \sigma, \\ c_2 & \text{if } (\Delta a_i \times \Delta o_i) > 0,\ \Delta a_i > \sigma, \\ c_3 & \text{if } (\Delta a_i \times \Delta o_i) < 0,\ \Delta a_i \leq \sigma, \\ c_4 & \text{if } (\Delta a_i \times \Delta o_i) < 0,\ \Delta a_i > \sigma, \end{cases} \tag{2.4}$$

where $\Delta a_i = a_i - a_{i-1}$, $\Delta o_i = o_i - o_{i-1}$, and σ is the standard deviation of the training data (including validation set). For the experiments authors used $c_1 = 0.5$, $c_2 = 0.8$, $c_3 = 1.2$, and $c_4 = 1.5$ [24, 25]. By giving these weights, they tried to impose a higher penalty the predictions whose direction is wrong and the magnitude of the error is lager, than the other predictions.

Based on this Directional Profit adjustment factor (2.4), Yao and Tan [24, 25] proposed Directional Profit (DP) model [24, 25]:

$$E_{\mathrm{DP}} = \frac{1}{N}\sum_{i=1}^{N} f_{\mathrm{DP}}(i)(a_i - o_i)^2. \tag{2.5}$$

Refenes et al. [27] proposed Discounted Least Squares (LDSs) function by taking the contribution from the historical data into accounts as follows:

$$E_{\mathrm{DLS}} = \frac{1}{N}\sum_{i=1}^{N} w_b(i)(a_i - o_i)^2, \tag{2.6}$$

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

$$w_b(i) = \frac{1}{1 + \exp(b - 2bi/N)}. \tag{2.7}$$

Discount rate b denotes the contribution from the historical data. Refenes et al. [27] suggested $b = 6$.

Yao and Tan [24, 25] proposed another error function, Time Dependent directional Profit (TDP) model, by incorporating the approach suggested by Refenes et al. [27] to their Directional Profit Model (2.5):

$$E_{\mathrm{TDP}} = \frac{1}{N}\sum_{i=1}^{N} f_{\mathrm{TDP}}(i)(a_i - o_i)^2, \tag{2.8}$$

where $f_{\mathrm{TDP}}(i) = f_{\mathrm{DP}}(i) \times w_b(i)$. $f_{\mathrm{DP}}(i)$ and $w_b(i)$ are described by (2.4) and (2.7), respectively.

Note. Refenes et al. [27] and Yao and Tan [24, 25] used $1/2N$ instead of $1/N$ in the formulas given by (2.5), (2.6), and (2.8).

2.2. Modified Error Functions

We are interested in classifying trading signals into three classes: buy, hold, and sell. The hold class includes both positive and negative values (see Criterion A in Section 1). Therefore, the least squares functions, in which the cases with incorrectly predicted directions (positive or negative) are penalised (e.g., the error functions given by (2.5) and (2.8)), will not give the desired prediction accuracy. For example, suppose that $a_i = 0.0045$ and $o_i = -0.0049$. In this case the predicted signal is correct, according to Criterion A. However, the algorithms used in [24, 25] try to minimise error function as $\Delta a_i \times \Delta o_i < 0$ (refer (2.8)). In fact such a minimisation is not necessary, as the predicted signal is correct. Therefore, instead of the weighing schemes suggested by previous studies, we proposed a different scheme of weighing.

Unlike the weighing schemes suggested in [24, 25], which impose a higher penalty on the predictions whose sign (i.e., negative or positive) is incorrect, 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 and, otherwise, assign a weight equal to 1. Therefore, the proposed weighing scheme is

$$w_d(i) = \begin{cases} \delta & \text{if the predicted trading signal is correct,} \\ 1, & \text{otherwise,} \end{cases} \tag{2.9}$$

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

2.2.1. Proposed Error Function 1

The weighing scheme, $f_{\mathrm{DP}}(i)$, incorporated in the Directional Profit (DP) error function (2.5) considers only two classes, upward and downward trends (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 $f_{\mathrm{DP}}(i)$ with the new weighing scheme $w_d(i)$ (see (2.9)). Hence, the new error function (E_{CC}) is defined as

$$E_{CC} = \frac{1}{N}\sum_{i=1}^{N} w_d(i)(a_i - o_i)^2. \tag{2.10}$$

When training backpropagation neural networks using (2.10) 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.

2.2.2. Proposed Error Function 2

The contribution from the historical data also plays an important role in the prediction accuracy of financial time series. Therefore, Yao and Tan [24, 25] went further by combining DP error function (see (2.5)) with DLS error function (see (2.6)) and proposed Time Dependent Directional Profit (TDP) error function (see (2.8)).

Following Yao and Tan [23, 24], this study also proposed a similar error function, E_{TCC}, by combining first new error function (E_{CC}) described by (2.10) with the DLS error function (E_{DLS}). Hence the second proposed error function is

$$E_{TCC} = \frac{1}{N}\sum_{i=1}^{N} w_b(i) \times w_d(i)(a_i - o_i)^2, \tag{2.11}$$

where $w_b(i)$ and $w_d(i)$ are defined by (2.7) and (2.9), respectively.

The difference between the TDP error function (see (2.8)) and this second new error function (2.11) is that $f_{\mathrm{DP}}(i)$ is replaced by $w_d(i)$ in order to deal with three classes: buy, hold, and sell.

2.3. *Modified Neural Network Algorithms*

Modifications to neural network algorithms were done by (i) using the OLS error function as well as the modified least squares error functions; (ii) employing a global optimization algorithm to train the networks.

The importance of using global optimization algorithms for the FNN training was discussed in Section 1. In this paper, we applied the global optimization algorithm, AGOP (introduced in [28, 29]), for training the proposed network algorithms.

As the error function to be minimised, we considered E_{OLS} (see (2.1)) and E_{DLS} (see (2.6)) together with the two modified error functions E_{CC} (see (2.10)) and E_{TCC} (see (2.11)). Based on these four error functions, we proposed the following algorithms:

(i) NN_{OLS}—neural network algorithm based on the Ordinary Least Squares error function, E_{OLS} (see (2.1));

(ii) NN_{DLS}—neural network algorithm based on the Discounted Least Squares error function, E_{DLS} (see (2.6));

(iii) NN_{CC}—neural network algorithm based on the newly proposed error function 1, E_{CC} (see (2.10));

(iv) NN_{TCC}—neural network algorithm based on the newly proposed error function 2, E_{TCC} (see (2.11)).

The layers are connected in the same structure as the FNN (Section 2). 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.

Algorithm NN_{OLS} differs from the standard FNN algorithm since it employs a new global optimization algorithm for training. Similarly, NN_{DLS} also differs from the respective algorithm used in [24, 25] due to the same reason. In addition to the use of new training algorithm, NN_{CC} and NN_{TCC} are based on two different modified error functions. The only way to examine whether these new modified neural network algorithms perform better than the existing ones (in the literature) is to conduct numerical experiments.

3. Network Training and Evaluation

The Australian All Ordinary Index (AORD) was selected as the stock market index whose trading signals are to be predicted. The previous studies done by the authors [22] suggested that the lagged Close prices of the US S\&P 500 Index (GSPC), the UK FTSE 100 Index (FTSE), French CAC 40 Index (FCHI), and German DAX Index (GDAXI) as well as that of the AORD itself showed an impact on the direction of the Close price of day t of the AORD. Also it was found that only the Close prices at lag 1 of these markets influence the Close price of the AORD [22, 23]. Therefore, this study considered the relative return of the Close prices at lag 1 of two combinations of stock market indices when forming input sets: (i) a combination which includes the GSPC, FTSE, FCHI, and the GDAXI; (ii) a combination which includes the AORD in addition to the markets included in (i).

The input sets were formed with and without incorporating the quantified intermarket influence [22, 23, 30] (see Section 3.1). By quantifying intermarket influence, this study tries to identify the influential patterns between the potential influential markets and the AORD. Training the network algorithms with preidentified patterns may enhance their learning. Therefore, it can be expected that the using quantified intermarket influence for training algorithms produces more accurate output.

The quantification of intermarket influence is described in Section 3.1, while Section 3.2 presents the input sets used for network training.

Daily relative returns of the Close prices of the selected stock market indices from 2nd July 1997 to 30th December 2005 were used for this study. If no trading took place on a particular day, the rate of change of price should be zero. Therefore, before calculating the relative returns, the missing values of the Close price were replaced by the corresponding Close price of the last trading day.

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 Section 2.2) as 0.01. If the trading signals are correctly predicted, 0.01 is small enough to set the value of the proposed error functions (see (2.10) and (2.11)) to approximately zero.

Since, influential patterns between markets are likely to vary with time [30], the whole study period was divided into a number of moving windows of a fixed length. Overlapping windows of length three trading years were considered $(1 \text{ trading year} \equiv 256 \text{ trading days})$. A period of three trading years consists of enough data (768 daily relative returns) for neural network experiments. Also the chance that outdated data (which is not relevant for studying current behaviour of the market) being included in the training set is very low.

The most recent 10% of data (the last 76 trading days) in each window were accounted for out of sample predictions, while the remaining 90% of data were allocated for network training. We called the part of the window which allocated for training the *training window*. Different number of neurons for the hidden layer was tested when training the networks with each input set.

As described in Section 2.1, the error function, E_{DLS} (see (2.6)), consists of a parameter b (discount rate) which decides the contribution from the historical data of the observations in the time series. Refenes et al. [27] 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 was selected, and this best value was used as b when training network NN_{TCC}.

3.1. Quantification of Intermarket Influences

Past studies [31–33] confirmed that the most of the world's major stock markets are integrated. Hence, one integrated stock market can be considered as a part of a single global system. The influence from one integrated stock market on a dependent market includes 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 influence from individual influential markets. Instead of measuring the individual influence from one influential market to a dependent market, the relative strength of the influence from this influential market to the dependent market can be measured compared to the influence from the other influential markets. This study used the approach proposed in [22, 23] to quantify intermarket influences. This approach estimates the combined influence of a set of influential markets and also the contribution from each influential market to the combined influence.

Quantification of intermarket influences on the AORD was carried out by finding the coefficients, ξ_i, $i = 1, 2, \ldots$ (see Section 3.1.1), which maximise the median rank correlation between the relative return of the Close of day $(t + 1)$ of the AORD market and the sum of ξ_i multiplied by the relative returns of the Close prices of day t of a combination of influential markets over a number of small nonoverlapping windows of a fixed size. The two combinations of markets, which are previously mentioned this section, were considered. ξ_i measures the contribution from the ith influential market to the combined influence which is estimated by the optimal correlation.

There is a possibility that the maximum value leads to a conclusion about a relationship which does not exist in reality. In contrast, the median is more conservative in this respect. Therefore, instead of selecting the maximum of the optimal rank correlation, the median was considered.

Spearman's rank correlation coefficient was used as the rank correlation measure. For two variables X and Y, Spearman's rank correlation coefficient, r_s, can be defined as

$$r_s = \frac{n(n^2 - 1) - 6\sum {d_i}^2 - (T_x - T_y)/2}{\sqrt{(n(n^2 - 1) - T_x)(n(n^2 - 1) - T_Y)}}, \tag{3.1}$$

where n is the total number of bivariate observations of x and y, d_i is the difference between the rank of x and the rank of y in the ith observation, and T_x and T_y are the number of tied observations of X and Y, respectively.

The same six training windows employed for the network training were considered for the quantification of intermarket influence on the AORD. The correlation structure between stock markets also changes with time [31]. Therefore, each moving window was further divided into a number of small windows of length 22 days. 22 days of a stock market time series represent a trading month. Spearman's rank correlation coefficients (see (3.1)) were calculated for these smaller windows within each moving window.

The absolute value of the correlation coefficient was considered when finding the median optimal correlation. This is appropriate as the main concern is the strength rather than the direction of the correlation (i.e., either positively or negatively correlated).

The objective function to be maximised (see Section 3.1.1 given below) is defined by Spearman's correlation coefficient, which uses ranks of data. Therefore, the objective function is discontinuous. Solving such a global optimization problem is extremely difficult because of the unavailability of gradients. We used the same global optimization algorithm, AGOP, which was used for training the proposed algorithms (see Section 2.3) to solve this optimization problem.

3.1.1. Optimization Problem

Let $Y(t + 1)$ be the relative return of the Close price of a selected dependent market at time $t + 1$, and let $X_j(t)$ be the relative return of the Close price of the jth influential market at time t. Define $X_\xi(t)$ as

$$X_\xi(t) = \sum_j \xi_j X_j(t), \tag{3.2}$$

where the coefficient $\xi_j \geq 0$, $j = 1, 2, \ldots, m$ measures the strength of influence from each influential market X_j, while m is the total number of influential markets.

The aim is to find the optimal values of the coefficients, $\xi = (\xi_1, \ldots, \xi_m)$, which maximise the rank correlation between $Y(t+1)$ and $X_\xi(t)$ for a given window.

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

$$T\left(t^0, l\right) = \left\{t^0,\ t^0 + 1, \ldots,\ t^0 + (l-1)\right\}, \tag{3.3}$$

where t^0 is the starting date of the window, and l is its size (in days). This study sets $l = 22$ days.

Spearman's correlation (see (3.1)) between the variables $Y(t+1)$, $X_\xi(t)$, $t \in T(t^0, l)$, defined on the window $T(t^0, l)$, will be denoted as

$$C(\xi) = \text{Corr}\left(Y(t+1),\ X_\xi(t) \| \ T\left(t^0, l\right)\right). \tag{3.4}$$

To define optimal values of the coefficients for a long time period, the following method is applied. Let $[1, T] = \{1, 2, \ldots, T\}$ be a given period (e.g., a large window). This period is divided into n windows of size l (we assume that $T = l \times n$, $n > 1$ is an integer) as follows:

$$T(t_k, l), \quad k = 1, 2, 3, \ldots, n, \tag{3.5}$$

so that,

$$\begin{gathered} T(t_k, l) \cap T(t_{k'}, l) = \phi \quad \text{for} \ \ \forall k \neq k', \\ \bigcup_{k=1}^{n} T(t_k, l) = [1, T]. \end{gathered} \tag{3.6}$$

The correlation coefficient between $Y(t+1)$ and $X_\xi(t)$ defined on the window $T(t_k, l)$ is denoted as

$$C_k(\xi) = \text{Corr}\left(Y(t+1),\ X_\xi(t) \| \ T(t_k, l)\right), \quad k = 1, \ldots, n. \tag{3.7}$$

To define an objective function over the period $[1, T]$, the median of the vector, $(C_1(\xi), \ldots, C_n(\xi))$, is used. Therefore, the optimization problem can be defined as

$$\begin{aligned} &\text{Maximise} \quad f(\xi) = \text{Median}(C_1(\xi), \ldots, C_n(\xi)), \\ &\text{s. t.} \quad \sum_j \xi_j = 1, \quad \xi_j \geq 0, \ j = 1, 2, \ldots, m. \end{aligned} \tag{3.8}$$

The solution to (3.8) is a vector, $\xi = (\xi_1, \ldots, \xi_m)$, where ξ_j, $j = 1, 2, \ldots, m$ denotes the strength of the influence from the jth influential market.

In this paper, the quantity, $\xi_j X_j$, is called the quantified relative return corresponding to the jth influential market.

3.2. Input Sets

The following six sets of inputs were used to train the modified network algorithms introduced in Section 2.3.

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

(2) Four input features of the quantified relative returns of the Close prices of day t of the market combination (i) (i.e., ξ_1 GSPC(t), ξ_2 FTSE(t), ξ_3 FCHI(t), and ξ_4 GDAXI(t))—denoted by **GFFG-q**.

(3) Single input feature consists of the sum of the quantified relative returns of the Close prices of day t of the market combination (i) (i.e., ξ_1 GSPC(t) + ξ_2 FTSE(t) + ξ_3 FCHI(t) + ξ_4 GDAXI(t))—denoted by **GFFG-sq**.

(4) Five input features of the relative returns of the Close prices of day t of the market combination (ii) (i.e., GSPC(t), FTSE(t), FCHI(t), GDAXI(t), and AORD(t))—denoted by **GFFGA**.

(5) Five input features of the quantified relative returns of the Close prices of day t of the market combination (ii) (i.e., ξ_1^A GSPC(t), ξ_2^A FTSE(t), ξ_3^A FCHI(t), ξ_4^A GDAXI(t), and ξ_5^A AORD(t))—denoted by **GFFGA-q**.

(6) Single input feature consists of the sum of the quantified relative returns of the Close prices of day t of the market combination (ii) (i.e., ξ_1^A GSPC(t) + ξ_2^A FTSE(t) + ξ_3^A FCHI(t) + ξ_4^A GDAXI(t) + ξ_5^A AORD(t))—denoted by **GFFGA-sq**.

$(\xi_1, \xi_2, \xi_3, \xi_4)$ and $(\xi_1^A, \xi_2^A, \xi_3^A, \xi_4^A)$ are solutions to (3.8) corresponding to the market combinations (i) and (ii), previously mentioned in Section 3. These solutions relating to the market combinations (i) and (ii) are shown in the Tables 1 and 2, respectively. We note that ξ_i and ξ_i^A, $i = 1, 2, 3, 4$ are not necessarily be equal.

3.3. Evaluation Measures

The networks proposed in Section 2.3 output the $(t+1)$th day relative returns of the Close price of the AORD. Subsequently, the output was classified into trading signals according to Criterion A (see Section 1).

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:

$$r_{CA} = \frac{N_0}{N_T} \times 100, \tag{3.9}$$

where N_0 and N_T are the number of test cases with correct predictions and the total number of cases in the test sample, respectively, as follows:

$$\begin{aligned} r_{E1} &= \frac{N_1}{N_T} \times 100, \\ r_{E2} &= \frac{N_2}{N_T} \times 100, \end{aligned} \tag{3.10}$$

Table 1: Optimal values of quantification coefficients (ξ) and the median optimal Spearman's correlations corresponding to market combination (i) for different training windows.

Training window no.	Optimal values of ξ				Optimal median Spearman's correlation
	GSPC	FTSE	FCHI	GDAXI	
1	0.57	0.30	0.11	0.02	0.5782*
2	0.61	0.18	0.08	0.13	0.5478*
3	0.77	0.09	0.13	0.01	0.5680*
4	0.79	0.06	0.15	0.00	0.5790*
5	0.56	0.17	0.03	0.24	0.5904*
6	0.66	0.06	0.08	0.20	0.5359*

* Significant at 5% level

Table 2: Optimal values of quantification coefficients (ξ) and the median optimal Spearman's correlations corresponding to market combination (ii) for different training windows.

Training window no.	Optimal values of ξ					Optimal median Spearman's correlation
	GSPC	FTSE	FCHI	GDAXI	AORD	
1	0.56	0.29	0.10	0.03	0.02	0.5805*
2	0.58	0.11	0.12	0.17	0.02	0.5500*
3	0.74	0.00	0.17	0.02	0.07	0.5697*
4	0.79	0.07	0.14	0.00	0.00	0.5799*
5	0.56	0.17	0.04	0.23	0.00	0.5904*
6	0.66	0.04	0.09	0.20	0.01	0.5368*

* Significant at 5% level

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.

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 lose the opportunity of making a profit, but no monetary loss. The most serious monetary loss occurs when a buy signal is misclassified as a sell signal and viceversa. Because of the seriousness of the mistake, r_{E2} plays a more important role in performance evaluation than r_{E1}.

4. Results Obtained from Network Training

As mentioned in Section 3, different values for the discount rate, b, were tested. $b = 1, 2, \ldots, 12$ was considered when training NN_{DLS}. The prediction results improved with the value of b up to 5. For $b > 5$ the prediction results remained unchanged. Therefore, the value of b was fixed at 5. As previously mentioned (see Section 3), $b = 5$ was used as the discount rate also in NN_{TCC} algorithm.

We trained the four neural network algorithms by varying the structure of the network; that is by changing the number of hidden layers as well as the number of neurons per hidden layer. The best four prediction results corresponding to the four networks were obtained when the number of hidden layers equal to one is and, the number of neurons per hidden layer is equal to two (results are shown in Tables 12, 13, 14, 15). Therefore, only the

Table 3: Results obtained from training neural network, NN_{OLS}. The best prediction results are 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 4: Results obtained from training neural network, NN_{DLS}. The best prediction results are 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 5: Results obtained from training neural network, NN_{CC}. The best prediction results are 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 6: Results obtained from training neural network, NN_{TCC}. The best prediction results are 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**	**33.37**
GFFGA-q	63.82	0.22	35.96
GFFG-sq	64.25	0.44	35.31
GFFGA-sq	64.69	0.22	35.09

results relevant to networks with two hidden neurons are presented in this section. Table 3 to Table 6 present the results relating to neural networks, NN_{OLS}, NN_{DLS}, NN_{CC}, and NN_{TCC}, respectively.

The best prediction results from NN_{OLS} were obtained when the input set GFFG-q (see Section 3.2) was used as the input features (see Table 3). This input set consists of four inputs of the quantified relative returns of the Close price of day t of the GSPC and the three European stock indices.

Table 7: Results obtained from training standard FNN algorithms. The best prediction results are shown in bold colour.

Input set	Average r_{CA}	Average r_{E2}	Average r_{E1}
GFFG	62.06	0.22	37.72
GFFGA	62.06	0.22	37.72
GFFG-q	**62.72**	**0.00**	**37.28**
GFFGA-q	**62.72**	**0.00**	**37.28**
GFFG-sq	62.28	0.00	37.72
GFFGA-sq	62.50	0.00	37.50

Table 8: Average (over six windows) classification and misclassification rates of the best prediction results corresponding to NN_{OLS} (trained with input set GFFG-q; refer Table 3).

Actual class	Average classification (misclassification) rates Predicted class		
	Buy	Hold	Sell
Buy	23.46%	(76.54%)	(0.00%)
Hold	(5.00%)	88.74%	(6.27%)
Sell	(0.00%)	(79.79%)	20.21%

Table 9: Average (over six windows) classification and misclassification rates of the best prediction results corresponding to NN_{DLS} (trained with input set GFFGA-sq; refer Table 4).

Actual class	Average classification (misclassification) rates Predicted class		
	Buy	Hold	Sell
Buy	22.10%	(77.90%)	(0.00%)
Hold	(4.97%)	89.20%	(5.83%)
Sell	(0.00%)	(83.06%)	16.94%

NN_{DLS} yielded nonzero values for the more serious classification error, r_{E2}, when the multiple inputs (either quantified or not) were used as the input features (see Table 4). The best results were obtained when the networks were trained with the 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; see Section 3.2). When the networks were trained with the single inputs (input sets GFFG-sq and GFFGA-sq; see Section 3.2) the serious misclassifications were prevented.

The overall prediction results obtained from the NN_{OLS} seem to be better than those relating to NN_{DLS}, (see Tables 3 and 4).

Compared to the predictions obtained from NN_{DLS}, those relating to NN_{CC} are better (see Tables 4 and 5). In this case the best prediction results were obtained when the relative returns of day t of the GSPC and the three European stock market indices (input set GFFG) were used as the input features (see Table 5). The classification rate was increased by 1.02% compared to that of the best prediction results produced by NN_{OLS} (see Tables 3 and 5).

Table 6 shows that NN_{TCC} also produced serious misclassifications. However, these networks produced high overall classification accuracy and also 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.

Table 10: Average (over six windows) classification and misclassification rates of the best prediction results corresponding to NN_{CC} (trained with input set GFFG; refer Table 5).

Actual class	Average classification (misclassification) rates Predicted class		
	Buy	Hold	Sell
Buy	23.94%	(76.06%)	(0.00%)
Hold	(5.00%)	89.59%	(6.66%)
Sell	(0.00%)	(77.71%)	22.29%

Table 11: Average (over six windows) classification and misclassification rates of the best prediction results corresponding to NN_{TCC} (trained with input set GFFG-q; refer Table 6).

Actual class	Average classification (misclassification) rates Predicted class		
	Buy	Hold	Sell
Buy	27.00%	(73.00%)	(0.00%)
Hold	(4.56%)	89.22%	(6.22%)
Sell	(0.00%)	(75.49%)	24.51%

The accuracy was the best among all four types of neural network algorithms considered in this study.

NN_{TCC} provided 1.34% increase in the overall classification rate compared to NN_{CC}. When compared with the 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.

4.1. Comparison of the Performance of Modified Algorithms with that of the Standard FNN Algorithm

Table 7 presents the average (over six windows) classification rates, and misclassification rates related to prediction results obtained by training the standard FNN algorithm which consists of one hidden layer with two neurons. In order to compare the prediction results with those of the modified neural network algorithms, the number of hidden layers was fixed as one, while the number of hidden neurons were fixed as two. These FNNs was trained for the same six windows (see Section 3) with the same six input sets (see Section 3.2). The transfer functions employed are same as those of the modified neural network algorithms (see Section 2.3).

When the overall classification and overall misclassification rates given in Table 7 are compared with the respective rates (see Tables 3 to 6) corresponding to the modified neural network algorithms, it is clear that the standard FNN algorithm shows poorer performance than those of all four modified neural network algorithms. Therefore, it can be suggested that all modified neural network algorithms perform better when predicting the trading signals of the AORD.

4.2. Comparison of the Performance of the Modified Algorithms

The best predictions obtained by each algorithm were compared by using classification and misclassification rates. 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,

Table 12: Results obtained from training neural network, NN_{OLS} with different number of hidden neurons.

Input set	No. of hidden neurons	Average r_{CA}	Average r_{E2}	Average r_{E2}
GFFG	1	64.25	0.00	35.75
	2	64.25	0.00	35.75
	3	64.25	0.00	35.75
	4	64.25	0.22	35.53
	5	64.25	0.00	35.75
	6	64.25	0.00	35.75
GFFGA	1	64.25	0.00	35.75
	2	64.25	0.00	35.75
	3	64.04	0.00	35.96
	4	64.25	0.00	35.75
	5	64.25	0.00	35.75
	6	64.25	0.00	35.75
GFFG-q	1	64.47	0.00	35.53
	2	**64.69**	**0.00**	**35.31**
	3	64.47	0.00	35.53
	4	64.04	0.00	35.96
	5	**64.69**	**0.00**	**35.31**
	6	64.25	0.00	35.75
GFFGA-q	1	64.25	0.00	35.75
	2	64.04	0.00	35.96
	3	63.60	0.22	36.18
	4	64.04	0.00	35.96
	5	64.25	0.00	35.75
	6	63.82	0.00	36.18
GFFG-sq	1	63.82	0.00	36.18
	2	63.82	0.00	36.18
	3	63.82	0.00	36.18
	4	63.82	0.00	36.18
	5	63.82	0.00	36.18
	6	63.82	0.00	36.18
GFFGA-sq	1	63.60	0.00	36.40
	2	63.60	0.00	36.40
	3	63.60	0.00	36.40
	4	63.60	0.00	36.40
	5	63.60	0.00	36.40
	6	63.60	0.00	36.40

the 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.

4.2.1. Prediction Accuracy

The average (over six windows) classification and misclassification rates related to the best prediction results obtained from NN_{OLS}, NN_{DLS}, NN_{CC}, and NN_{TCC} are shown in Tables 8 to 11, respectively.

Table 13: Results obtained from training neural network, NN_{DLS} with different number of hidden neurons.

Input set	No. of hidden neurons	Average r_{CA}	Average r_{E2}	Average r_{E1}
GFFG	1	64.47	0.44	35.09
	2	64.25	0.44	35.71
	3	64.03	0.44	35.53
	4	64.25	0.44	35.31
	5	64.25	0.44	35.31
	6	64.25	0.44	35.31
GFFGA	1	64.03	0.44	35.53
	2	64.03	0.44	35.53
	3	64.03	0.44	35.53
	4	64.03	0.44	35.53
	5	64.03	0.44	35.53
	6	64.03	0.44	35.53
GFFG-q	1	64.47	0.22	35.31
	2	64.47	0.22	35.31
	3	64.69	0.22	35.09
	4	64.47	0.22	35.31
	5	64.25	0.22	35.53
	6	64.47	0.22	35.31
GFFGA-q	1	64.69	0.22	35.09
	2	64.25	0.22	35.53
	3	63.82	0.22	35.96
	4	64.25	0.44	35.31
	5	64.47	0.44	35.09
	6	64.25	0.22	35.53
GFFG-sq	1	63.82	0.00	36.18
	2	63.82	0.00	36.18
	3	63.82	0.00	36.18
	4	63.82	0.00	36.18
	5	63.82	0.00	36.18
	6	63.82	0.00	36.18
GFFGA-sq	**1**	**64.04**	**0.00**	**35.96**
	2	**64.04**	**0.00**	**35.96**
	3	**64.04**	**0.00**	**35.96**
	4	**64.04**	**0.00**	**35.96**
	5	**64.04**	**0.00**	**35.96**
	6	**64.04**	**0.00**	**35.96**

Among the best networks corresponding to the four algorithms considered, the best network of the algorithm based on the proposed error function 2 (see (2.11)) showed the best classification accuracies relating to buy and sell signals (27% and 25%, resp.; see Tables 8 to 11). Also this network classified more than 89% of the hold signals accurately and it is the second best rate for the hold signal. The rate of misclassification from hold signals to buy is the lowest when this network was used for prediction. The rate of misclassification from hold class to sell class is also comparatively low (6.22%, which is the second lowest among the four best predictions).

Table 14: Results obtained from training neural network, NN_{CC} with different number of hidden neurons.

Input set	No. of hidden neurons	Average r_{CA}	Average r_{E2}	Average r_{E1}
GFFG	1	62.72	0.66	36.62
	2	**65.35**	**0.00**	**34.65**
	3	63.60	0.00	36.40
	4	63.38	0.22	36.40
	5	64.25	0.00	35.75
	6	64.69	0.00	35.31
GFFGA	1	64.04	0.00	35.96
	2	64.03	0.22	35.75
	3	63.16	0.00	36.84
	4	64.04	0.00	35.96
	5	64.03	0.44	35.53
	6	64.04	0.00	35.96
GFFG-q	1	63.38	0.00	36.62
	2	63.82	0.00	36.18
	3	63.60	0.00	36.40
	4	64.91	0.22	34.87
	5	64.03	0.22	35.75
	6	64.69	0.00	35.31
GFFGA-q	1	65.35	0.22	34.43
	2	64.04	0.00	35.96
	3	64.04	0.00	35.96
	4	63.38	0.00	36.62
	5	65.13	0.00	34.87
	6	63.82	0.00	36.18
GFFG-sq	1	64.25	0.00	35.75
	2	64.25	0.00	35.75
	3	64.04	0.00	35.96
	4	64.04	0.00	35.96
	5	64.25	0.00	35.75
	6	64.04	0.00	35.96
GFFGA-sq	1	63.82	0.00	36.18
	2	63.82	0.00	36.18
	3	63.82	0.00	36.18
	4	63.82	0.00	36.18
	5	63.82	0.00	36.18
	6	63.82	0.00	36.18

The network corresponding to the algorithm based on the proposed error function 1 (see (2.10)) produced the second best prediction results. This network accounted for the second best prediction accuracies relating to buy and sell signals while it produced the best predictions relating to hold signals (Table 10).

4.3. Comparisons of Results with Other Similar Studies

Most of the studies [8, 9, 11, 13, 22], which used FNN algorithms for predictions, are aimed at predicting the direction (up or down) of a stock market index. Only a few studies [14, 17],

Table 15: Results obtained from training neural network, NN_{TCC} with different number of hidden neurons.

Input set	No. of hidden neurons	Average r_{CA}	Average r_{E2}	Average r_{E1}
GFFG	1	65.57	0.44	33.99
	2	66.67	0.44	32.89
	3	64.47	0.44	35.09
	4	65.57	0.22	34.21
	5	65.13	0.22	34.65
	6	64.91	0.22	34.87
GFFGA	1	64.69	0.22	35.09
	2	64.91	0.22	34.87
	3	65.13	0.00	34.87
	4	65.13	0.22	34.35
	5	64.13	0.22	34.65
	6	65.57	0.22	34.21
GFFG-q	1	64.91	0.22	34.87
	2	**66.23**	**0.00**	**33.77**
	3	65.57	0.00	34.43
	4	65.79	0.22	33.99
	5	65.13	0.22	34.65
	6	66.23	0.22	33.55
GFFGA-q	1	65.57	0.22	34.21
	2	63.82	0.22	35.96
	3	64.91	0.00	35.09
	4	63.82	0.22	35.96
	5	64.69	0.22	35.09
	6	64.47	0.00	35.53
GFFG-sq	1	65.13	0.44	34.43
	2	64.25	0.44	35.31
	3	64.91	0.44	34.65
	4	64.47	0.44	35.09
	5	64.69	0.44	34.87
	6	64.69	0.44	34.87
GFFGA-sq	1	64.69	0.22	35.09
	2	64.69	0.22	35.09
	3	64.69	0.22	35.09
	4	64.91	0.22	34.87
	5	64.91	0.22	34.87
	6	64.69	0.22	35.09

which used the AORD as the target market index, predicted whether to buy, hold or sell stocks. These studies employed the standard FNN algorithm (that is with OLS error function) for prediction. However, the comparison of results obtained from this study with the above mentioned two studies is impossible as they are not in the same form.

5. Conclusions

The results obtained from the experiments show that the modified neural network algorithms introduced by this study perform better than the standard FNN algorithm in predicting the trading signals of the AORD. Furthermore, the neural network algorithms, based on the modified OLS error functions introduced by this study (see (2.10) and (2.11)), produced better predictions of trading signals of the AORD. Of these two algorithms, the one-based on (2.11) showed the better performance. This algorithm produced the best predictions when the network consisted of one hidden layer with two neurons. The quantified relative returns of the Close prices 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 viceversa and also predicted trading signals with a higher degree of accuracy.

Also it can be suggested that the quantified intermarket influence on the AORD can be effectively used to predict its trading signals.

The algorithms proposed in this paper can also be used to predict whether it is best to buy, hold, or sell shares of any company listed under a given sector of the Australian Stock Exchange. For this case, the potential influential variables will be the share price indices of the companies listed under the stock of interest.

Furthermore, the approach proposed by this study can be applied to predict trading signals of any other global stock market index. Such a research direction would be very interesting especially in a period of economic recession, as the stock indices of the world's major economies are strongly correlated during such periods.

Another useful research direction can be found in the area of marketing research. That is the modification of the proposed prediction approach to predict whether market share of a certain product goes up or not. In this case market shares of the competitive brands could be considered as the influential variables.

References

[1] B. Egeli, M. Ozturan, and B. Badur, "Stock market prediction using artificial neural networks," in *Proceedings of the 3rd Hawaii International Conference on Business*, pp. 1–8, Honolulu, Hawaii, USA, June 2003.

[2] R. Gençay and T. Stengos, "Moving average rules, volume and the predictability of security returns with feedforward networks," *Journal of Forecasting*, vol. 17, no. 5-6, pp. 401–414, 1998.

[3] M. Qi, "Nonlinear predictability of stock returns using financial and economic variables," *Journal of Business & Economic Statistics*, vol. 17, no. 4, pp. 419–429, 1999.

[4] M. Safer, "A comparison of two data mining techniques to predict abnormal stock market returns," *Intelligent Data Analysis*, vol. 7, no. 1, pp. 3–13, 2003.

[5] L. Cao and F. E. H. Tay, "Financial forecasting using support vector machines," *Neural Computing & Applications*, vol. 10, no. 2, pp. 184–192, 2001.

[6] W. Huang, Y. Nakamori, and S.-Y. Wang, "Forecasting stock market movement direction with support vector machine," *Computers and Operations Research*, vol. 32, no. 10, pp. 2513–2522, 2005.

[7] S. H. Kim and S. H. Chun, "Graded forecasting using an array of bipolar predictions: application of probabilistic neural networks to a stock market index," *International Journal of Forecasting*, vol. 14, no. 3, pp. 323–337, 1998.

[8] H. Pan, C. Tilakaratne, and J. Yearwood, "Predicting Australian stock market index using neural networks exploiting dynamical swings and intermarket influences," *Journal of Research and Practice in Information Technology*, vol. 37, no. 1, pp. 43–54, 2005.

[9] M. Qi and G. S. Maddala, "Economic factors and the stock market: a new perspective," *Journal of Forecasting*, vol. 18, no. 3, pp. 151–166, 1999.

[10] Y. Wu and H. Zhang, "Forward premiums as unbiased predictors of future currency depreciation: a non-parametric analysis," *Journal of International Money and Finance*, vol. 16, no. 4, pp. 609–623, 1997.

[11] J. Yao, C. L. Tan, and H. L. Poh, "Neural networks for technical analysis: a study on KLCI," *International Journal of Theoretical and Applied Finance*, vol. 2, no. 2, pp. 221–241, 1999.

[12] M. T. Leung, H. Daouk, and A.-S. Chen, “Forecasting stock indices: a comparison of classification and level estimation models,” *International Journal of Forecasting*, vol. 16, no. 2, pp. 173–190, 2000.

[13] K. Kohara, Y. Fukuhara, and Y. Nakamura, “Selective presentation learning for neural network forecasting of stock markets,” *Neural Computing & Applications*, vol. 4, no. 3, pp. 143–148, 1996.

[14] C. D. Tilakaratne, M. A. Mammadov, and S. A. Morris, “Effectiveness of using quantified intermarket influence for predicting trading signals of stock markets,” in *Proceedings of the 6th Australasian Data Mining Conference (AusDM ′07)*, vol. 70 of *Conferences in Research and Practice in Information Technology*, pp. 167–175, Gold Coast, Australia, December 2007.

[15] R. Akbani, S. Kwek, and N. Japkowicz, “Applying support vector machines to imbalanced datasets,” in *Proceedings of the 15th European Conference on Machine Learning (ECML ′04)*, pp. 39–50, Springer, Pisa, Italy, September 2004.

[16] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, “SMOTE: synthetic minority oversampling technique,” *Journal of Artificial Intelligence Research*, vol. 16, pp. 321–357, 2002.

[17] C. D. Tilakaratne, S. A. Morris, M. A. Mammadov, and C. P. Hurst, “Predicting stock market index trading signals using neural networks,” in *Proceedings of the 14th Annual Global Finance Conference (GFC ′07)*, pp. 171–179, Melbourne, Australia, September 2007.

[18] I. Jordanov, “Neural network training and stochastic global optimization,” in *Proceedings of the 9th International Conference on Neural Information Processing (ICONIP ′02)*, vol. 1, pp. 488–492, Singapore, November 2002.

[19] J. Minghu, Z. Xiaoyan, Y. Baozong, et al., “A fast hybrid algorithm of global optimization for feedforward neural networks,” in *Proceedings of the 5th International Conference on Signal Processing (WCCC-ICSP ′00)*, vol. 3, pp. 1609–1612, Beijing, China, August 2000.

[20] K. A. Toh, J. Lu, and W. Y. Yau, “Global feedforward neural network learning for classification and regression,” in *Proceedings of the 3rd International Workshop on Energy Minimization Methods in Computer Vision and Pattern Recognition (EMMCVPR ′01)*, pp. 407–422, Shophia Antipolis, France, September 2001.

[21] H. Ye and Z. Lin, “Global optimization of neural network weights using subenergy tunneling function and ripple search,” in *Proceedings of the IEEE International Symposium on Circuits and Systems (ISCAS ′03)*, vol. 5, pp. 725–728, Bangkok, Thailand, May 2003.

[22] C. D. Tilakaratne, M. A. Mammadov, and C. P. Hurst, “Quantification of intermarket influence based on the global optimization and its application for stock market prediction,” in *Proceedings of the 1st International Workshop on Integrating AI and Data Mining (AIDM ′06)*, pp. 42–49, Horbart, Australia, December 2006.

[23] C. D. Tilakaratne, S. A. Morris, M. A. Mammadov, and C. P. Hurst, “Quantification of intermarket influence on the Australian all ordinary index based on optimization techniques,” *The ANZIAM Journal*, vol. 48, pp. C104–C118, 2007.

[24] J. Yao and C. L. Tan, “A study on training criteria for financial time series forecasting,” in *Proceedings of the International Conference on Neural Information Processing (ICONIP ′01)*, pp. 1–5, Shanghai, China, November 2001.

[25] J. Yao and C. L. Tan, “Time dependent directional profit model for financial time series forecasting,” in *Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks (IJCNN ′00)*, vol. 5, pp. 291–296, Como, Italy, July 2000.

[26] R. B. Caldwell, “Performances metrics for neural network-based trading system development,” *NeuroVe$t Journal*, vol. 3, no. 2, pp. 22–26, 1995.

[27] A. N. Refenes, Y. Bentz, D. W. Bunn, A. N. Burgess, and A. D. Zapranis, “Financial time series modelling with discounted least squares backpropagation,” *Neurocomputing*, vol. 14, no. 2, pp. 123–138, 1997.

[28] M. A. Mammadov, “A new global optimization algorithm based on dynamical systems approach,” in *Proceedings of the 6th International Conference on Optimization: Techniques and Applications (ICOTA ′04)*, A. Rubinov and M. Sniedovich, Eds., Ballarat, Australia, December 2004.

[29] M. Mammadov, A. Rubinov, and J. Yearwood, “Dynamical systems described by relational elasticities with applications,” in *Continuous Optimization: Current Trends and Applications*, V. Jeyakumar and A. Rubinov, Eds., vol. 99 of *Applied Optimization*, pp. 365–385, Springer, New York, NY, USA, 2005.

[30] C. D. Tilakaratne, “A study of intermarket influence on the Australian all ordinary index at different time periods,” in *Proceedings of the 2nd International Conference for the Australian Business and Behavioural Sciences Association (ABBSA ′06)*, Adelaide, Australia, September 2006.

[31] C. Wu and Y.-C. Su, "Dynamic relations among international stock markets," *International Review of Economics & Finance*, vol. 7, no. 1, pp. 63–84, 1998.

[32] J. Yang, M. M. Khan, and L. Pointer, "Increasing integration between the United States and other international stock markets? A recursive cointegration analysis," *Emerging Markets Finance and Trade*, vol. 39, no. 6, pp. 39–53, 2003.

[33] M. Bhattacharyya and A. Banerjee, "Integration of global capital markets: an empirical exploration," *International Journal of Theoretical and Applied Finance*, vol. 7, no. 4, pp. 385–405, 2004.