# A FORECASTING OF INDICES AND CORRESPONDING INVESTMENT DECISION MAKING APPLICATION.

**Pretesh Bhoola Patel.**

A Dissertation submitted to the Faculty of Engineering and the Built Environment, University of the Witwatersrand, in fulfillment of the requirements of the degree of Master of Science.

Johannesburg 2006.

# Preface.

The work described in this dissertation has been carried out at the University of Witwatersrand, School of Electrical and Information Engineering in 2005 and 2006. This work has been sponsored in part by the National Research Foundation.

I would like to acknowledge a few people that have made this research a possibility due to their extensive support. Firstly, Prof. Tshilidzi Marwala. He has been my supervisor for this project and has provided plenty of insight into the development of this application. I also thank him for all the support he has provided.

I would like to thank my parents, Mr. Bhoola Patel and Mrs. Yoginiben Patel, as well as my brother, Mr. Ketan Patel and my sister, Mrs. Anisha Patel. They have supported me throughout my educational years both emotionally and financially.

Lastly, I would like to dedicate my research in memory of a very close friend, Mr. Sanjay Ramasamy. He has provided me with support, guidance as well as encouragement throughout my undergraduate degree. He will be missed.

Aside from references that have been made to the work of others, I affirm that this research is a result of my own original work. No part of this work has already been, or is currently being, submitted for any other degree, diploma or other qualification.

P. B. Patel.
April 2006.

# Abstract.

Due to the volatile nature of the world economies, investing is crucial in ensuring an individual is prepared for future financial necessities. This research proposes an application, which employs computational intelligent methods that could assist investors in making financial decisions. This system consists of 2 components. The Forecasting Component (FC) is employed to predict the closing index price performance. Based on these predictions, the Stock Quantity Selection Component (SQSC) recommends the investor to purchase stocks, hold the current investment position or sell stocks in possession. The development of the FC module involved the creation of Multi-Layer Perceptron (MLP) as well as Radial Basis Function (RBF) neural network classifiers. Categorizes that these networks classify are based on a profitable trading strategy that outperforms the long-term “Buy and hold” trading strategy. The Dow Jones Industrial Average, Johannesburg Stock Exchange (JSE) All Share, Nasdaq 100 and the Nikkei 225 Stock Average indices are considered. It has been determined that the MLP neural network architecture is particularly suited in the prediction of closing index price performance. Accuracies of 72%, 68%, 69% and 64% were obtained for the prediction of closing price performance of the Dow Jones Industrial Average, JSE All Share, Nasdaq 100 and Nikkei 225 Stock Average indices, respectively. Three designs of the Stock Quantity Selection Component were implemented and compared in terms of their complexity as well as scalability. Complexity is defined as the number of classifiers employed by the design. Scalability is defined as the ability of the design to accommodate the classification of additional investment recommendations. Designs that utilized 1, 4 and 16 classifiers, respectively, were developed. These designs were implemented using MLP neural networks, RBF neural networks, Fuzzy Inference Systems as well as Adaptive Neuro-Fuzzy Inference Systems. The design that employed 4 classifiers achieved low complexity and high scalability. As a result, this design is most appropriate for the application of concern. It has also been determined that the neural network architecture as well as the Fuzzy Inference System implementation of this design performed equally well.

# Table of contents.

## List of figures.

# List of Tables.

# List of Abbreviations.

| | |
|---|---|
| ANFIS | Adaptive Neuro-Fuzzy Inference System. |
| ANN | Artificial Neural Network. |
| EMH | Efficient Market Hypothesis. |
| FC | Forecasting Component. |
| FIS | Fuzzy Inference System. |
| HR | Hit Rate. |
| JSE | Johannesburg Stock Exchange or JSE Securities Exchange. |
| LD | Large Drop. |
| LR | Large Rise. |
| Nasdaq | National Association of Securities Dealers Automated Quotation. |
| NASD | National Association of Securities Dealers. |
| MLP | Multi-Layer Perceptron. |
| RBF | Radial Basis Function. |
| RMS | Root Mean Square. |
| ROC | Receiver Operating Characteristic. |
| RWH | Random Walk Hypothesis. |
| SD | Slight Drop. |
| SR | Slight Rise. |
| SQSC | Stock Quantity Selection Component. |

# 1. Introduction.

Trading in stock market indices has gained exceptional popularity in major financial markets worldwide. Due to the increasing diversity of financial index related instruments as well as the economic growth experienced during the past years, the extent of global investment opportunities for both individual and institutional investors has broadened [1]. As a result, it is of critical importance that applications which assist investors in making financial decisions be developed.

Investing in the stock market can be exciting and it could provide huge benefits. Due to the volatile nature of the world economies, investing is crucial in ensuring an individual is financially prepared for retirement. Investing is perceived as owning assets, such as stocks or real estate, which over time produce substantial earnings.

One of the major benefits of investing is the protection it offers against inflation. Inflation is the rate at which the general level of prices for goods and services rise, thus reducing purchasing power. It provides a technique that can make more money than an individual would lose through inflation. This would allow individuals to save for expenses such as tertiary education for their children or retirement.

However, there are numerous risks involved in investing in the stock market. The capital that is invested in the stock market is not guaranteed. For instance, one may purchase a stock expecting a certain rate of share price increase. If the company experiences financial problems, it may not live up to the expected share price growth. If the company files for bankruptcy, it is possible that an individual would lose all invested capital. Due to the uncertainty of the outcome, one bears a certain amount of risk when purchasing a stock.

A major risk in investing in the stock markets is the reaction of an investment instrument to news items about a certain industry sector. Depending on the interpretation of the news, investors could be influenced to purchase or sell stocks. If sufficient number of investors begins to purchase or sell shares simultaneously, this could cause the stock price to rise or drop.

The aim of this research is to develop a system, using computational intelligent methods, which could assist investors in making financial decisions. It is anticipated that this application would be used in conjunction with other financial analysis techniques. As a result, this application should be used to confirm an investment decision.

There are many funds that are highly correlated with indices [1]. As a result, the developed system is concerned with the Dow Jones Industrial Average, Johannesburg Stock Exchange or the JSE Securities Exchange (JSE) All Share, Nasdaq 100 and the Nikkei 225 Stock Average indices. This approach would assist in creating a diversified portfolio, as these indices are concerned with different industry sectors. A diversified portfolio is concerned with, among other aspects, investing in a wide range of stocks, instead of having the majority of trading capital invested in one particular share. It is a risk management technique, which ideally results in a lower risk portfolio.

The Dow Jones Industrial Average index is the most extensively used indicator of the overall condition of the stock market. It is a price-weighted average of 30 actively traded blue-chip stocks. The 30 stocks are chosen by the editors of the Wall Street Journal, which is published by Dow Jones and Company [2].

The JSE All Share index is an indicator of the overall performance of the JSE. The JSE is the largest stock exchange in Africa [3]. The exchange is ranked 18$^{th}$ in the world in terms of total market capitalization [4]. In 2003, it had an estimated 472 listed companies [3].

The Nasdaq 100 index comprises of the 100 largest domestic and international non-financial companies on the Nasdaq stock exchange [5]. The Nasdaq (once an acronym for the National Association of Securities Dealers Automated Quotation system) stock exchange is a computerized system established by the National Association of Securities Dealers (NASD) to facilitate trading by providing stock brokers with current prices on over-the-counter and some listed stocks. It does not have a physical trading floor that brings together stock brokers, instead all the trading is accomplished over a network of computers and telephones [5].

The Nikkei 225 Stock Average index or the Nikkei average is comprised of the top 225 blue-chip companies on the Tokyo Stock Exchange. It is a price-weighted index. The Nikkei average is calculated daily by the Nihon Keizai Shimbun newspaper [6].

The following section includes background information on the methods that are used in the development of the application. Thereafter, the developed system is described in detail, illustrating the various components of the system. An examination of the implementation methodology follows. The document ends with a conclusion. Recommendations for possible improvement in forecasting accuracies are also stated.

## *References.*


[1] Yahoo Finance, last accessed: 4 April 2006. [Online]. Available: http://finance.yahoo.com.

[2] J.A. Prestbo, *Markets Measure: An Illustrated History of America Told Through the Dow Jones Industrial Average.*, Dow Jones Publishers, 1999.

[3] "JSE Securities Exchange." Wikipedia. Wikipedia, 2005. *Answers.com* GuruNet Corp. Last accessed: 4 April 2006. [Online]. Available: http://www.answers.com/topic/jse-securities-exchange.

[4] "Economy of South Africa." Wikipedia. Wikipedia, 2005. *Answers.com* GuruNet Corp. Last accessed: 4 April 2006. [Online]. Available: http://www.answers.com/economy-of-south-africa.

[5] L. Heckman, *NASDAQ: A Guide to Information Sources.*, Brunner-Routledge, 2001.

[6] "Nikkei." Investopedia. Investopedia Inc., 2000. *Answers.com* GuruNet Corp. Last accessed: 4 April 2006. [Online]. Available: http://www.answers.com/topic/nikkei-225.

# 2. Background information.

In order to realize the objective of this research, it has been decided to develop a system that will be capable of forecasting the performance of indices as well as recommending the quantity of stocks to purchase and sell.

A stock is an instrument that signifies the ownership position, equity, within a corporation and represents a claim on its proportional share of the assets and profits of the company [1]. There are 2 main types of stock; common and preferred. Common stock typically entitles the owner the right to vote at shareholder meetings and to receive dividends that the corporation has declared. Preferred stock does not entitle the owner to voting rights, but it permits a higher claim on assets and earnings than the common stock. For example, a preferred stock owner receives dividends before a common stock owner and has priority in the event a corporation is bankrupt and, as a result, is liquidated [1].

The stock market is an organization for the trading of stocks and bonds. Such organizations were initially open to all, but at present only members, stock brokers, of the owning association may purchase and sell directly. Stock brokers buy and sell for themselves as well as for others, charging a certain commission for their services. When a stock is listed on an exchange and it meets certain requirements prescribed by the board of governors of the exchange, it may be bought or sold. Stock exchanges can be located at all important financial centers of the world [1].

An index can be defined as a statistical measure of the changes in a portfolio of stocks representing a portion of the overall market [1]. Ideally, a change in the price of an index would characterize an exactly proportional change in the stocks included in the index. It is important to note that an index is merely a list of stocks that anyone could create. The difference between the big indices and the small indices is the reputation of the company that publishes the index [1].

Computational intelligent methods such as Artificial Neural Networks (ANNs), Fuzzy Inference Systems (FISs) and Adaptive Neuro-Fuzzy Inference Systems (ANFISs) have been considered in the creation of the system.

Specifically, the Multi-Layer Perceptron (MLP) and the Radial Basis Function (RBF) neural network architectures have been utilized. These ANNs are feed-forward structured whereby each unit receives inputs only from lower layer units. In the majority of implementations, the network consists of 2 layers of adaptive weights with full connectivity between inputs and hidden units as well as between hidden units and outputs [2].

The training of the network is accomplished through backpropagation and a complex nonlinear hidden as well as output weights optimization. At iterations, the error of the network is assessed and the derivative of this error is calculated with respect to each weight within the network.

The error function generally used in Artificial Neural Network (ANN) computation is the squared difference between the actual and desired outputs. Optimization methods are then used to minimize the error function by altering the weights, initially in the output layer and then the hidden layer. Essentially, the error is backpropagated from the output of the network, through the output weights and to the hidden weights [2]. Detailed explanations on these ANN architectures can be found in Appendix C.

FISs utilize fuzzy inference rules. In each rule, there is a premise and a consequence. The premise is described by a fuzzy proposition and the consequence can be a fuzzy conclusion. Fuzzy inference methods are algorithms that deduce results from the fuzzy inference rules and present inputs. Fuzzy inference methods are based on fuzzy logic. A Fuzzy Inference System (FIS) consists of Fuzzification, Inference and Defuzzification processes. The Fuzzification process is a mapping from the observed input to the fuzzy sets defined in the corresponding universe. Inference process is a decision making logic that utilizes the fuzzy inference rules to determine fuzzy outputs corresponding to fuzzified inputs. Defuzzification produces non-fuzzy outputs [3].

The FIS utilized in the development of the application, employed subtractive clustering to generate the required membership functions and set of fuzzy inference rules. The objective of clustering is to locate “natural classes” in a set of given inputs such that similar inputs are grouped together in the same class [3].

Subtractive clustering is a modified form of the Mountain Method for cluster estimation [4]. Assuming N normalized points in an M-dimensional space, each data point is considered as a potential cluster center and defines a measure of the potential of a data point [5]. The measure of potential for a given point is a function of its distances to all other data points. A point with many neighbouring points will have a high potential value. After the potential of every data point has been computed, the point with the largest potential value is selected as the first cluster center. Thereafter, in order to determine the next cluster and its center, all the data points in the vicinity of the first cluster center, which is determined by a radius of influence or cluster radius, is removed. This process is iterated until all the input data are within a cluster radius of a cluster center [3]. Further information on the FISs utilized in this research can be found in Appendix D.

Another popular clustering technique, c-means clustering, has also been considered. However, this technique requires 2 predefined clusters. The quality of the c-means method depends strongly on the choice of the number of centers and the initial cluster positions [5]. This method is also known to possess the "curse of dimensionality". This implies that the number of rules increases exponentially as the input data increases in size. As a result of these problems, it had been decided to utilize the subtractive clustering technique [5].

Neuro-fuzzy modeling is an approach where the fusion of ANNs and fuzzy logic find their strengths. These 2 techniques complement each other. The neuro-fuzzy approach utilizes heuristic learning strategies, derived from the domain of ANNs, to support the development of a FIS. A union between ANNs and fuzzy logic techniques assist in addressing of short comings of both techniques [6].

Neuro-fuzzy techniques can learn the behaviour of the system from a sufficiently large data set and automatically generate fuzzy inference rules as well as membership functions to a pre-specified accuracy level. They are also capable of generalization, thus overcoming the key disadvantages of the fuzzy logic-based approaches. These key disadvantages are self-learning, inability to meet pre-specified accuracy and the lack of generalization capability [7].

The specific neuro-fuzzy method employed in this research is the Adaptive Neuro-Fuzzy Inference System (ANFIS). It utilizes a first order Sugeno-type inference process. Similar to ANNs, the ANFIS is presented with a training data set. The membership functions are extracted from the data set. The ANFIS learns features in the data set and adjusts the system parameters according to a given criterion. Further information on the ANFIS utilized can be found in [8].

## *References.*


[1] G. A. Fontanills, *The Stock Market Course.*, John Wiley & Sons Ltd, 2001.

[2] C. M. Bishop, *Neural Networks for Pattern Recognition.*, Oxford University Press, 1995.

[3] The Mathworks, Inc., *Fuzzy Logic Toolbox User's Guide Version 2.*, The Mathworks, Inc, 1995.

[4] J. Yen, L. Wang, "Constructing optimal fuzzy models using statistical information criteria.", *Journal of Intelligent and Fuzzy Systems.*, vol. 7, pp. 185-201, 1999.

[5] S. Chiu, "Fuzzy Model Identification Based on Cluster Estimation.", *Journal of Intelligent and Fuzzy Systems.*, vol. 2, no. 3, pp. 267-278, 1994.

[6] R. E. Uhrig, L. H. Tsoukala, *Fuzzy and Neural Approaches in Engineering.*, John Wiley & Sons Ltd, 1997.

[7] N. K. Kasabov, *Foundations of Neural Netwoks, Fuzzy Systems, and Knowledge Engineering.*, MIT Press, 1996.

[8] J-S. R. Jang, "ANFIS: Adaptive-Network-Based Fuzzy Inference System.", *IEEE Transaction on System, Man and Cybernetics.*, vol. 23, no. 3, pp. 665-685, 1993.

# A forecasting of indices and corresponding investment decision making application.

Pretesh B. Patel.

*Abstract*—**Accurate financial prediction is of great practical interest to both individual and institutional investors. This project proposes an application, which employs computational intelligent techniques that could be used to assist investors in making financial decisions. The Multi-Layer Perceptron (MLP) as well as Radial Basis Function (RBF) neural network architectures are implemented as classifiers to forecast the closing index price performance. Categorizes that these networks classify are based on a profitable trading strategy that outperforms the long-term "Buy and hold" trading strategy. The Dow Jones Industrial Average, Johannesburg Stock Exchange All Share, Nasdaq 100 and the Nikkei 225 Stock Average indices are considered. The best and worst forecasting classification accuracies obtained were 72% and 64%, respectively. These accuracy levels were attained for the Dow Jones Industrial Average and the Nikkei 225 Stock Average indices, respectively. Based on the forecasted performance of the indices considered, the system recommends the investor to purchase stocks, hold the current investment position or sell stocks in possession. Three designs of the Stock Quantity Selection Component were implemented and compared in terms of their complexity as well as scalability. Designs that utilized 1, 4 and 16 classifiers, respectively, were developed. These designs were implemented using MLP neural networks, RBF neural networks, Fuzzy Inference Systems as well as Adaptive Neuro-Fuzzy Inference Systems. The design that employed 4 classifiers achieved low complexity and high scalability. As a result, this design is most appropriate for the application of concern. It has also been determined that the neural network architecture as well as the Fuzzy Inference System implementation of this design performed equally well.**

## I. Introduction.

Trading in stock market indices has gained exceptional popularity in major financial markets worldwide. Due to the increasing diversity of financial index related instruments as well as the economic growth experienced during the past years, the extent of global investment opportunities for both individual and institutional investors has broadened [1]. As a result, it is of critical importance that applications which assist investors in making financial decisions be developed.

The aim of this research is to develop such an application, using computational intelligent methods, which could assist investors in making financial decisions. It is anticipated that this application would be used in conjunction with other financial analysis techniques. As a result, this application should be used to confirm an investment decision.

There are many funds that are highly correlated with indices [1]. As a result, the developed system is concerned with the Dow Jones Industrial Average, Johannesburg Stock Exchange or the JSE Securities Exchange (JSE) All Share, Nasdaq 100 and the Nikkei 225 Stock Average indices. This approach would assist in creating a diversified portfolio, as these indices are concerned with different industry sectors. A diversified portfolio is concerned with, among other aspects, investing in a wide range of stocks, instead of having the majority of trading capital invested in 1 particular share. It is a risk management technique, which ideally results in a lower risk portfolio.

Predicting stock market performance has been a major research area for many years. There are various schools of thought in terms of the ability to profit from the equity markets. Some believe that no investor can obtain above average trading advantages based on the historical and present information. The Random Walk Hypothesis (RWH) states that prices on the stock market wander in a purely random and unpredictable manner [2]. As a result, according to this theory, every price change occurs without any influence from past prices. The Efficient Market Hypothesis (EMH) states that the markets incorporate all available information and prices are adjusted immediately once new information becomes available [2]. If these theories are true, there should not be any advantage in predicting stock performance, as the market would react and compensate for any actions performed due to the predicted information.

These theories have been met with a great deal of opposition. The argument against the EMH is that many investors base their expectations on past prices, past earnings, track record as well as other indicators. Since stock prices are largely influenced by investor expectations, many believe it only makes sense that past prices do affect future prices.

Compelling evidence has also been provided that rejects the RWH [3]. It has been illustrated that stock market price movements, of the United States [4] as well as Japan [5], have conformed only to the weak form of the EMH. There has also been a study of 234 stocks from 8 major European stock markets, which indicated that these stock markets exhibited a slight departure from the RWH [6]. As a result, the above offers encouragement for research into developing market prediction applications.

Traditionally, moving average, exponential smoothing and linear regression statistical methods have been used in the prediction of stock prices [7]. Regression models have been used to identify cycles and trends [7].

Recently, artificial neural networks (ANNs) have been applied to solve problems of predicting future stock indices [9][10][11]. Advanced methods such as genetic algorithms [12], Markov models [13] and fuzzy methods [12] have also been frequently used. ANNs together with pattern recognition techniques for stock market forecasting have also been employed [14]. Research has also been conducted in the prediction of stocks using case-based reasoning [8]. Random subspace classifier networks have also been used to predict the next day stock price return [15].

In this research, the Multi-Layer Perceptron (MLP) and the Radial Basis Function (RBF) neural network architectures have been applied to closing index price performance forecasting. Detailed explanations on these Artificial Neural Network (ANN) architectures can be found in [16].

The system is to also recommend to the investor to purchase additional stocks, hold the current investment position or sell stocks in possession, based on the forecasted performance of the indices considered. As a result, this is a pattern classification problem.

The classification of data into various classes has been an important research area for many years. ANNs have been applied to pattern classification [17]. Research has also been conducted on fuzzy classification. This resulted in many algorithms, such as fuzzy K-nearest neighbour [18] and fuzzy c-means [19], being applied to decision making systems. Fuzzy systems constructed using genetic algorithms have been utilized [20][21][22]. Fuzzy neural networks have also been employed in pattern classification applications [23][24][25]. Support Vector Machines have been applied to multi-category classification problems [26]. These classification tasks have also been implemented by combining multiple simpler specialized classifiers [27][28][29].

In this project, the MLP and RBF neural network architectures, Fuzzy Inference Systems (FISs) as well as Adaptive Neuro-Fuzzy Inference Systems (ANFISs) have been applied to the classification of investment recommendations. The FISs developed employed subtractive clustering to generate the required membership functions and set of fuzzy inference rules. Information on these computational intelligent techniques can be found in [16], [30] and [31], respectively.

The section to follow examines the developed application. Thereafter, the forecasting classifier implementation methodology is described. The comparison of the various forecasting classifiers developed and the selection of the superior network follows. The investment recommendation classifier development methodology examination follows. The document ends with a comparison of the various investment recommendation classification models developed and the identification of the best performing classifiers. Recommendations for possible improvement in forecasting accuracies are also stated.

## II. THE DEVELOPED SYSTEM.

The primary aim of this project is to develop an application that could be used to assist an individual in making an investment decision concerning certain index funds. These index funds are based on the Dow Jones Industrial Average, JSE All Share, Nasdaq 100 and the Nikkei 225 Stock Average indices so as to mirror their performance.

In order to create such a system, it had been decided to develop an application that would be capable of forecasting the performance of the indices concerned as well as recommending the quantity of stocks to purchase and sell. Fig. 1 illustrates such a system.

As the developed system is to be used in assisting an investor in making financial decisions, the application should be based on a profitable trading strategy. There are numerous trading strategies available [32]. This research focuses on the "Buy low, sell high" trading strategy. The strategy has been implemented as well as compared to the "Buy and hold" trading strategy in terms of profits generated.

The "Buy low, sell high" trading strategy entails

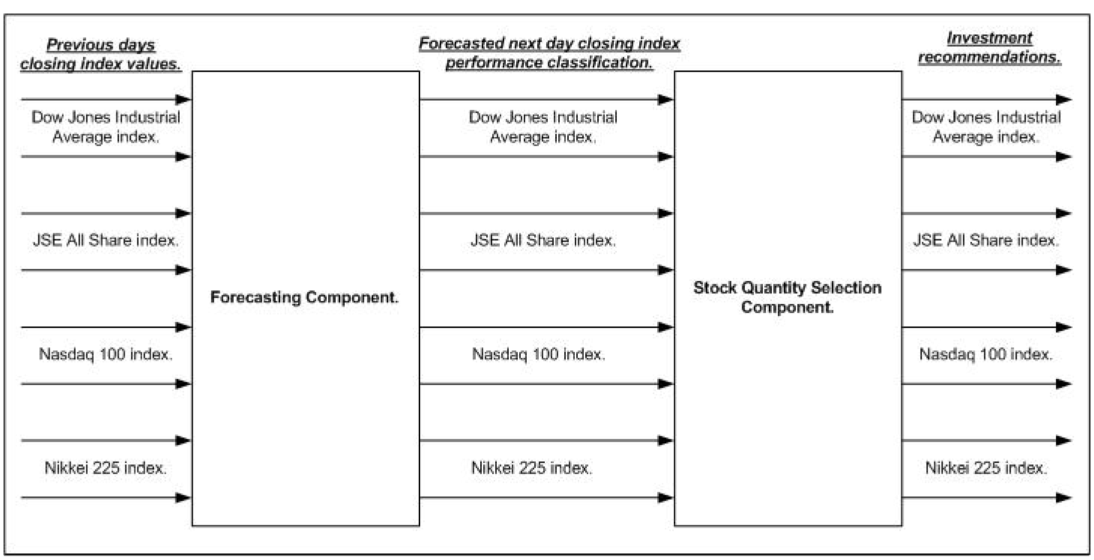


Fig. 1. The developed system.

purchasing certain stocks at a low price and selling these stocks when the price is high. The "Buy and hold" trading strategy, as the name suggests, involves an investor purchasing certain stocks and retaining them for a particular duration.

The strategies were implemented for the period 5 January 2004 to 16 September 2004 using the available data of the indices concerned. It has been assumed that the investment instrument price is equal to the index closing value divided by hundred. This provided a realistic index fund price. It has also been assumed that the investor had a limited trading capital of 100 000 available. This also assisted in simulating a real world scenario. Furthermore, the initial limited trading capital is divided into 4 equal portions, which were used to purchase units in the indices concerned. Thereafter, the trading strategies began.

The method used to implement the "Buy low, sell high" trading strategy involved classifying the change in index or delta into certain categorizes. Delta is defined as the difference between the closing index value for the next day and the closing index value for the previous day. This functionality has been implemented within the Forecasting Component (FC). Depending on the predicted index performance output of this component, the strategy would recommend the investor to purchase stocks, hold the current investment position or sell stocks. This responsibility can be found in the Stock Quantity Selection Component (SQSC).

Table I illustrates these classes as well as the corresponding strategy recommendation.

TABLE I:
THE "BUY LOW, SELL HIGH" TRADING STRATEGY CATEGORIZES

| Class | Requirement | Strategy recommendation |
|---|---|---|
| Large Rise (LR). | Delta > Positive threshold percentage of previous day closing price. | If LR is forecasted for the next day, sell stocks in possession at the next day closing price. |
| Slight Rise (SR). | 0 < Delta <= Positive threshold percentage of previous day closing price. | If SR is forecasted for the next day, hold current investment position. |
| Slight Drop (SD). | Negative threshold percentage of previous day closing price <= Delta <= 0. | If SD is forecasted for the next day, buy stocks to the value of 15 % of available trading capital at the next day closing price. |
| Large Drop (LD). | Delta < Negative threshold percentage of previous day closing price. | If LD is forecasted for the next day, buy stocks to the value of 25 % of available trading capital at the next day closing price. |

The percentage thresholds shown in Table II were determined through experimentation. The percentage thresholds were initially kept constant, for example 0.1% and -0.1% of the closing index value for the previous day. Table II illustrates some of these results. When compared to the "Buy and hold" trading strategy, the percentage threshold combination of 0.2% and -0.2% of the closing index value for the previous day performed well.

As a result, the investigation that followed kept the positive threshold constant at 0.2% of the closing value for the previous day, whereas the negative threshold had been varied. When compared to the "Buy and hold" strategy, the results indicate that the percentage threshold combination of 0.2% and -0.2% of the closing value for the previous day performed the best. Some of these results are shown in Table II.

Experiments that kept the negative threshold constant at -0.2% of the closing index value for the previous day, but varied the positive threshold were also conducted. It has been determined that the combination of 0.8% and -0.2% of the closing index value for the previous day percentage thresholds were most profitable, when compared to the "Buy and hold" trading strategy.

TABLE II:
RESULTS OF TRADING STRATEGY ANALYSIS

| Positive percentage threshold. | Negative percentage threshold. | Profits/Losses generated by "Buy low, sell high" trading strategy. | Profits/Losses generated by "Buy and hold" trading strategy. |
|---|---|---|---|
| Results of constant percentage threshold combinations. | | | |
| ***0.20*** | ***-0.20*** | ***4939.86*** | ***1137.50*** |
| 0.30 | -0.30 | 4348.64 | 1137.50 |
| 0.80 | -0.80 | 4166.90 | 1137.50 |
| 1.30 | -1.30 | 1974.50 | 1137.50 |
| 2.00 | -2.00 | -(1333.92) | 1137.50 |
| Results of constant positive percentage threshold, but varied negative percentage threshold. | | | |
| 0.20 | -0.40 | 4255.52 | 1137.50 |
| 0.20 | -0.80 | 3431.99 | 1137.50 |
| 0.20 | -0.90 | 3972.67 | 1137.50 |
| 0.20 | -1.4 | 4277.87 | 1137.50 |
| 0.20 | -2.10 | 4205.68 | 1137.50 |
| Results of varied positive percentage threshold, but constant negative percentage threshold. | | | |
| 0.10 | -0.20 | 4850.54 | 1137.50 |
| 0.50 | -0.20 | 3372.29 | 1137.50 |
| ***0.80*** | ***-0.20*** | ***5060.25*** | ***1137.50*** |
| 1.30 | -0.20 | 2426.21 | 1137.50 |
| 1.80 | -0.20 | -(1447.51) | 1137.50 |

As illustrated in Table II, the percentage threshold combination of 0.8% and -0.2% of the closing value for the previous day is most profitable for the period concerned. As a result, this percentage threshold combination has been used in the next analysis. This entailed varying the number of shares sold. It has been determined that, if the strategy categorizes the next day closing index price as a Large Rise (LR), all the shares in possession should be sold to achieve the maximum profit.

The above percentage thresholds were further verified by extending the trading strategy implementation to include 2005 data. The strategies were implemented for the extended period of 5 January 2004 to 31 May 2005. It has been established that the percentage thresholds chosen assisted in generating a profit at the end of this period of 13303.73, whereas the "Buy and hold" strategy produced a profit of 9768.51. It has also been determined that this percentage threshold combination out performed all the threshold combinations considered for this extended period. Table III illustrates some of the results obtained from the analysis. As a result, the system employs the percentage threshold combination of 0.8% and -0.2% of the closing index value for the previous day.

An objective of this application, as previously mentioned, is to predict the next day performance of the closing prices of the indices concerned. This is accomplished by categorizing the change in index or delta into the classes used in the "Buy low, sell high" trading strategy.

The forecasting of the indices can be classified as either multivariate or univariate models. A univariate model utilizes the past values of the time series to generate a prediction [33]. The disadvantage of this approach is that it does not consider the environmental effects and the interactions among different factors other than the outputs. A multivariate model employs additional information such as market indicators, technique indicators or fundamental factors of companies as inputs [33]. The disadvantage of a multivariate model is the difficulty involved in the selection of inputs. There are many factors leading to price fluctuations that cannot be captured precisely or may be too numerous and difficult to be modeled.

Several research outcomes, using the univariate approach, have illustrated acceptable results [33][34]. Networks with root mean square (RMS) errors between 0.0251 and 0.3318 were reported. As a result, this application utilizes the closing prices of the indices, represented by a univariate time series, as inputs.

The FC module employs 4 forecasting classifiers. Each of the classifiers is used to forecast the closing price performance of an index considered. Each of these classifiers utilized by this component has 4 outputs.

TABLE III:
VERIFICATION OF TRADING STRATEGY ANALYSIS.

| Positive percentage threshold. | Negative percentage threshold. | Profits/Losses generated by "Buy low, sell high" trading strategy. | Profits/Losses generated by "Buy and hold" trading strategy. |
|---|---|---|---|
| 0.10 | -0.10 | 11000.72 | 9768.51 |
| 0.20 | -0.20 | 12052.50 | 9768.51 |
| ***0.80*** | ***-0.20*** | ***13303.73*** | ***9768.51*** |
| 1.40 | -0.20 | 5377.86 | 9768.51 |
| 0.20 | -2.10 | 9251.24 | 9768.51 |

The initial design of the FC module entailed using an ANN to predict the next day closing value of the indices concerned, instead of using a classifier to predict the next day closing index performance. However investigations revealed that the ANN predicts a value to the actual closing index approximately a day late. This effect has also been noticed when utilizing committee neural network architectures as well as ANFISs. Several research results, also mention this effect [35], [36].

In order to resolve this effect, several techniques were suggested. One of the techniques entailed using another criterion, such as a Hit Rate (HR), to measure the performance of the networks. It has been suggested that this would measure the number of instances the network correctly predicted the direction of change [36]. The calculation of the HR used a threshold value. However, further investigation revealed that, depending on the threshold value selected, the HR would be large or small. As a result, this technique has not been employed. Another technique involved utilizing a second measurement during training of the ANN. This measurement would penalize the network when a delay has been detected. However, it was suggested that this would reduce the accuracy of the ANN [35]. As a result, this method was not utilized. Due to this effect, it had been decided to use the current design.

As previously stated, the primary objective of the SQSC module is to utilize the forecasted closing index price performance to generate investment recommendations of the quantity of stocks to purchase or sell. Pattern classification problems can be grouped as either dichotomous or polychotomous problems. Dichotomous classification can be interpreted as 2-class classification problems, whereas polychotomous classification involves problems with more than 2 classes to be categorized. As a result, it can be stated that this is a polychotomous classification problem as there are more than 2 classes.

Various classifier designs of the SQSC module were considered. Each of these designs were developed using both ANNs as well as fuzzy logic techniques. The first design employed 1 classifier. This classifier consisted of 16 inputs and 16 outputs. The inputs to the model are the forecasted performance of the closing price of the indices considered. The outputs of the classifier are the investment recommendations for the indices.

The second design involved 4 classifiers. Each classifier has 4 inputs and 4 outputs. Each classifier is used to generate an investment recommendation for an index considered. The input to a classifier is the forecasted performance of the closing price of an index. The output of a classifier is the investment recommendation for the index.

The third and final design considered utilized 16 classifiers. Each classifier has 4 inputs and 1 output. Each classifier is employed to categorize whether or not to execute an investment recommendation. The input to a classifier is the same as the input to the classifiers utilized in the design that employed 4 classifiers above. The outputs of the classifiers are fed into an interpretation function that generates the final investment recommendations for the indices. This design has been implemented to investigate the method of utilizing simpler classifiers to generate a multi-category classifier.

The following section will examine the implementation methodology employed in the creation of the FC and SQSC modules.

## III. IMPLEMENTATION METHODOLOGY.

### A. *The Forecasting Component.*

The data used to develop the FC module can be obtained from the internet [1]. The development process was divided into various stages. The following procedure has been pursued in the creation of the various ANN architectures employed:

1. Selection and processing of data to be used by ANNs during training, validation and testing.
2. Optimization of the number of hidden neurons or nodes within the ANN.
3. Optimization of the input time window using polynomial approximation.
4. Comparison of the various networks developed and the selection of the superior network.

The remainder of this subsection will elaborate on the various stages of the procedure stated above.

#### 1) *Selection and processing of data.*

The data utilized in developing the ANNs included closing price indices values from 5 January 2004 to 31 May 2005.

Before the data sets were analyzed, basic preprocessing of the information was considered. In the case of days with no trading, the missing data maybe required to be manually inserted. It has been stated that there are 3 techniques which could be employed to contend with days with no trading [37]. These are:

1. Ignore the days with no trading and use the data for trading days only.
2. Assign a 0 value for the days with no trading.
3. Build a model that can approximate the value for the days with no trading.

A model was not created to determine the value for the days with no trading because it was feared that the values calculated may contribute significantly to the final error of the networks. Initial experiments were conducted, utilizing techniques 1 and 2 above, and it was found that, technique 1 resulted in lower error values. As a result, technique 1, from the above list, has been employed.

It should also be noted that the data sets utilized during the development of the ANNs consisted of closing values for the indices for the days when trading occurred on all the 4 stock exchanges considered. The closing values for the indices for the days when trading did not occur on all 4 stock exchanges of concern, but occurred on some stock markets considered, were omitted from the above data sets. This assisted in creating a realistic trading scenario.

In order to ensure that over-fitting and under-fitting were avoided, the data is divided into 3 sets. Over-fitting occurs when the network does not generalize but rather tends to memorize the training data. Under-fitting occurs when the network does not follow the data at all [16]. The data is divided into training, validation and test sets. The training data set is used to train the ANN to find the general pattern between its inputs and outputs. The validation data set is used to assess the network and the test data is used to confirm the prediction quality of the developed networks.

The data is segregated in time order. In other words, the data of the earlier period could be used for training the network, the data of the later period be used for validation and the data of the latest period used for testing. This approach may have a recency problem that is the ANNs are only trained using data from early 2004. When forecasting the performance of the indices during 2005, the ANNs are "forced" to utilize the knowledge learnt in early 2004.

Due to the above problem, the data is uniformly randomized and, thereafter, separated into the 3 required sets. As a result, the ANNs are trained using randomly chosen data. A network with very good test data set results may not predict well for future forecasting. However, a network that has been trained with randomly chosen data may predict well for future forecasting even with average test data set results [38].

The output and input data sets of the indices to be forecasted were preconditioned by normalizing the data. Normalizing the data entails manipulating the data sets such that the values within the sets are between 0 and 1. The networks developed were trained utilizing the normalized data sets.

Normalization is accomplished by acquiring the minimum and maximum values within the data sets. The data is normalized by using the following formula:

$$X_{norm} = \frac{X - X_{\min}}{X_{\max} - X_{\min}}, \qquad (1)$$

where

$X_{norm}$ is the normalized value,

$X$ is the actual data,

$X_{\min}$ is the smallest value within the data set,

$X_{\max}$ is the largest value within the data set.

The purpose of normalizing the data sets is to modify the variable levels to a reasonable value. If such a transformation is not employed, the value of the variable could be too large for the network to process, especially when several layers of nodes within the ANN are involved [39]. Normalizing the data sets also reduces the fluctuation and noise within the data [16].

There are a variety of practical reasons that illustrate normalizing the data sets can result in faster training and reduce the chances of obtaining local optima. Some of these reasons include better numerical conditioning (Hessian matrices), better weight initialization values and better weight decay estimates [16].

*2) Optimization of the number of hidden nodes within the neural network.*

MLP and the RBF neural network architectures were utilized in forecasting the next day closing price performance for each of the indices considered. The MLP and RBF neural network architectures are possibly the most extensively employed ANNs in pattern classification [17]. Due to the non-linear capabilities of these networks, they are said to be excellent universal approximators that provide highly accurate solutions. As a result, these networks produce very practical tools for classification and inversion problems [16].

It has been stated that a network with 1 hidden layer, provided with sufficient data, can be used to model any function [40]. As a result, the MLP and RBF neural network architectures employed consisted of only 1 hidden layer.

During this stage of development, the number of inputs has been assumed to be arbitrary. This will be optimized at a later stage of implementation. This stage of development involved optimizing the ANN architecture. As a result, designing the ANN thus entailed selecting the correct number of hidden neurons and the appropriate network architectures that would yield the most accurate results.

The inputs to the networks were kept constant. There were 7 inputs to the networks. These were the previous 7 day closing index prices. The developed ANN had 4 outputs. Each output represented a performance class considered. The largest value, from the 4 neural classifier network outputs, indicates that the network forecasts that the next day closing index price will behave according to the performance class corresponding to that particular output.

The MLP network hidden layer consists of non-linear activation functions. The choice of the activation function is mainly dependant on the application of the network [16]. However, it has been found that the hyperbolic tangent activation function offers a practical advantage of giving rise to faster convergence during training [17]. As a result, this function has been utilized within the MLP networks.

The MLP network output layer also consists of activation functions. There are 3 major forms of the function that should be considered. These are the linear, logistic sigmoidal and softmax activation functions [17]. It has been stated that the appropriate selection of the output-unit activation function for a classification problem is the logistic sigmoidal function [17]. As a result, this function has been employed within the output layer of the MLP network.

The RBF network that has been developed contained a Gaussian activation function within its hidden layer and a linear activation function within its output layer.

The number of hidden neurons or nodes has been optimized by minimizing an error function that mapped the number of hidden nodes to the accuracy of the developed networks. The process has been performed on the validation data set.

Since this is a classification implementation, the accuracy of the networks developed can no longer be calculated utilizing the sum of square error of the difference between the target and the forecasted network output values. Instead, a confusion matrix is employed to identify the number of true and false classifications that are generated by the ANN developed. This is then utilized to calculate the true accuracy of the ANN classifiers, using the following equation:

$$Accuracy = \sqrt{\frac{TP \times TN}{(TP + FN) \times (FP + TN)}}\,, \qquad (2)$$

where

$TP$ is the true positive (1 classified as a 1),

$TN$ is the true negative (0 classified as a 0),

$FN$ is the false negative (1 classified as a 0),

$FP$ is the false positive (0 classified as a 1).

The hidden neurons or intermediate units were optimized by creating various MLP and RBF networks with hidden nodes of 5 to 150. As a result, 292 ANNs were developed. Networks with hidden nodes greater than 150 were not developed due to the predictive capabilities or generalization capabilities reducing as the number of intermediate units increase. More hidden nodes increases the dimensionality of the function being fitted, enabling easier training which results from higher training capacity. However, this detrimentally affects the generalization capabilities of the network. A major consideration when developing a suitable ANN for a financial application is to make a trade-off between convergence and generalization [41]. Utilizing the training data set, these networks were trained. The validation data set was then presented to the networks. Thereafter, the accuracy of the developed networks was calculated for the training and validation data sets. MLP and RBF networks with the number of hidden nodes that resulted in the largest accuracy value, when presented with the validation data set, were analyzed.

Table IV illustrates the MLP and RBF networks that resulted in the largest accuracy value for the validation data set. Networks consisting of these numbers of hidden nodes were developed in the next stage of implementation.

TABLE IV:
RESULTS OF VARIED HIDDEN NODES.

| Network architecture. | Number of hidden nodes. | Accuracy (Validation data set). |
|---|---|---|
| Dow Jones Industrial Average. | | |
| MLP | 18 | 0.56885 |
| MLP | 55 | 0.60007 |
| MLP | 58 | 0.60007 |
| MLP | 64 | 0.60007 |
| MLP | 78 | 0.60007 |
| RBF | 29 | 0.61538 |
| RBF | 55 | 0.55292 |
| RBF | 73 | 0.55292 |
| RBF | 106 | 0.56885 |
| RBF | 116 | 0.58456 |
| JSE All Share. | | |
| MLP | 26 | 0.56885 |
| MLP | 35 | 0.58456 |
| MLP | 58 | 0.58456 |
| MLP | 96 | 0.53674 |
| MLP | 107 | 0.53674 |
| RBF | 5 | 0.55292 |
| RBF | 12 | 0.55292 |
| RBF | 16 | 0.55292 |
| RBF | 18 | 0.53674 |
| RBF | 22 | 0.55292 |
| NASDAQ 100. | | |
| MLP | 51 | 0.55292 |
| MLP | 100 | 0.60007 |
| MLP | 119 | 0.60007 |
| MLP | 135 | 0.56885 |
| MLP | 148 | 0.56885 |
| RBF | 42 | 0.60007 |
| RBF | 64 | 0.60007 |
| RBF | 74 | 0.58456 |
| RBF | 99 | 0.58456 |
| RBF | 131 | 0.60007 |
| Nikkei 225 Stock Average. | | |
| MLP | 60 | 0.55292 |
| MLP | 84 | 0.56885 |
| MLP | 103 | 0.55292 |
| MLP | 118 | 0.55292 |
| MLP | 128 | 0.56885 |
| RBF | 19 | 0.50356 |
| RBF | 64 | 0.53674 |
| RBF | 95 | 0.53674 |
| RBF | 105 | 0.53674 |
| RBF | 138 | 0.5203 |

*3) Optimization of the input time window using polynomial approximation.*

This stage of implementation involved the optimization of the number of inputs that would concede the largest forecasting classification accuracy. As a result, this step of development entailed selecting the correct number of closing prices of the indices for the previous days as inputs to the networks.

The number of inputs has been optimized by minimizing an error function that mapped the number of inputs to the accuracy of the developed networks. The process was performed on the validation and test data sets.

The input time window was optimized by constructing various MLP and RBF networks with the number of closing prices for the previous days, required to predict the desired output, ranging from 5 to 19. These developed networks contained the number of hidden nodes as illustrated in Table IV. The networks also employed the same activation functions mentioned in the previous section.

Utilizing the training data set, these networks are trained. The validation and test data sets are then presented to the ANN. Thereafter, the accuracies for the training, validation and test data sets are calculated. When presented with the validation and test data sets, ANNs that resulted in the largest accuracy were analyzed. Table V illustrates the MLP and RBF networks that resulted in the best accuracy values for the validation and test data.

Moving averages of 5, 6, 7 and 8 days of the closing index prices were also considered as inputs to the networks. The longer the time span of the moving average, the less sensitive it will be to daily price changes [42]. This is the reason for utilizing these moving averages as the ANNs are predicting the next day closing price performance for each index. Moving averages are utilized to emphasize the direction of a trend and reduce price as well as volume fluctuations that may confuse interpretations [42]. As a result, the moving average is employed to reduce the noise within the data.

Moving averages were introduced as inputs to the networks mentioned in Table V. The number of closing prices of the indices for the previous days is kept constant at the values illustrated in Table V. However, the moving average employed varied from a 5 day moving average to an 8 day moving average. It has been determined, from this investigation, that the moving averages introduced did increase the accuracy of the networks. This is valid for the Dow Jones Industrial Average, Nasdaq 100 and the Nikkei 225 Stock Average indices. However, the accuracy of the networks employed to forecast the closing price performance of the JSE All Share index did not improve with the addition of the moving averages as inputs. Table VI contains the results of this investigation.

TABLE V:
RESULTS OF VARIED INPUT DAYS.

| Network architecture. | Input days. | Hidden nodes. | Accuracy (Validation). | Accuracy (Test). |
|---|---|---|---|---|
| Dow Jones Industrial Average. | | | | |
| MLP | 16 | 18 | 0.675 | 0.63052 |
| MLP | 17 | 55 | 0.66032 | 0.61538 |
| MLP | 16 | 58 | 0.6455 | 0.61538 |
| MLP | 15 | 64 | 0.63052 | 0.60007 |
| MLP | 18 | 78 | 0.70396 | 0.60007 |
| RBF | 7 | 29 | 0.61538 | 0.53674 |
| RBF | 16 | 55 | 0.60007 | 0.55292 |
| RBF | 18 | 73 | 0.61538 | 0.58456 |
| RBF | 17 | 106 | 0.60007 | 0.56885 |
| RBF | 18 | 116 | 0.6455 | 0.55292 |
| JSE All Share. | | | | |
| MLP | 13 | 26 | 0.66032 | 0.6455 |
| MLP | 18 | 26 | 0.675 | 0.675 |
| MLP | 18 | 35 | 0.6455 | 0.675 |
| MLP | 18 | 96 | 0.6455 | 0.6455 |
| MLP | 19 | 96 | 0.675 | 0.66032 |
| RBF | 8 | 12 | 0.56885 | 0.58456 |
| RBF | 13 | 16 | 0.58456 | 0.56885 |
| RBF | 9 | 18 | 0.56885 | 0.61538 |
| RBF | 12 | 18 | 0.58456 | 0.56885 |
| RBF | 18 | 22 | 0.56885 | 0.55292 |
| NASDAQ 100. | | | | |
| MLP | 10 | 51 | 0.6455 | 0.68954 |
| MLP | 11 | 100 | 0.61538 | 0.61538 |
| MLP | 18 | 119 | 0.66032 | 0.6455 |
| MLP | 19 | 135 | 0.66032 | 0.66032 |
| MLP | 19 | 148 | 0.66032 | 0.68954 |
| RBF | 19 | 64 | 0.63052 | 0.63052 |
| RBF | 14 | 99 | 0.61538 | 0.61538 |
| RBF | 15 | 99 | 0.6455 | 0.63052 |
| RBF | 19 | 99 | 0.6455 | 0.63052 |
| RBF | 15 | 131 | 0.63052 | 0.63052 |
| Nikkei 225 Stock Average. | | | | |
| MLP | 16 | 60 | 0.63052 | 0.61538 |
| MLP | 19 | 84 | 0.60007 | 0.61538 |
| MLP | 18 | 103 | 0.63052 | 0.66032 |
| MLP | 19 | 118 | 0.61538 | 0.61538 |
| MLP | 19 | 128 | 0.60007 | 0.60007 |
| RBF | 15 | 64 | 0.55292 | 0.56885 |
| RBF | 17 | 95 | 0.55292 | 0.61538 |
| RBF | 14 | 138 | 0.55292 | 0.55292 |
| RBF | 16 | 138 | 0.55292 | 0.58456 |
| RBF | 18 | 138 | 0.58456 | 0.61538 |

TABLE VI:
RESULTS OF VARIED DAY MOVING AVERAGE.

| Network architecture. | Input days. | Day moving average. | Hidden nodes. | Accuracy (Validation). | Accuracy (Test). |
|---|---|---|---|---|---|
| Dow Jones Industrial Average. | | | | | |
| MLP | 16 | 5 | 58 | 0.70396 | 0.70396 |
| MLP | 16 | 6 | 58 | 0.70396 | 0.70396 |
| MLP | 15 | 5 | 64 | 0.70396 | 0.70396 |
| MLP | 15 | 7 | 64 | 0.71826 | 0.71826 |
| MLP | 15 | 8 | 64 | 0.71826 | 0.73245 |
| RBF | 16 | 6 | 55 | 0.61538 | 0.58456 |
| RBF | 18 | 5 | 73 | 0.6455 | 0.55292 |
| RBF | 17 | 5 | 106 | 0.63052 | 0.56885 |
| RBF | 18 | 7 | 116 | 0.675 | 0.55292 |
| RBF | 18 | 8 | 116 | 0.63052 | 0.60007 |
| JSE All Share. | | | | | |
| MLP | 18 | 7 | 26 | 0.58456 | 0.60007 |
| MLP | 18 | 8 | 26 | 0.61538 | 0.60007 |
| MLP | 18 | 7 | 35 | 0.61538 | 0.60007 |
| MLP | 18 | 8 | 35 | 0.66032 | 0.58456 |
| MLP | 18 | 6 | 96 | 0.6455 | 0.58456 |
| RBF | 8 | 5 | 12 | 0.53674 | 0.58456 |
| RBF | 13 | 5 | 16 | 0.5203 | 0.55292 |
| RBF | 9 | 6 | 18 | 0.55292 | 0.5203 |
| RBF | 12 | 5 | 18 | 0.5203 | 0.58456 |
| RBF | 18 | 5 | 22 | 0.56885 | 0.55292 |
| NASDAQ 100. | | | | | |
| MLP | 18 | 6 | 119 | 0.675 | 0.675 |
| MLP | 19 | 5 | 135 | 0.66032 | 0.66032 |
| MLP | 19 | 7 | 135 | 0.675 | 0.66032 |
| MLP | 19 | 5 | 148 | 0.68954 | 0.70396 |
| MLP | 19 | 6 | 148 | 0.66032 | 0.66032 |
| RBF | 19 | 5 | 64 | 0.66032 | 0.60007 |
| RBF | 19 | 6 | 64 | 0.66032 | 0.63052 |
| RBF | 19 | 5 | 99 | 0.6455 | 0.61538 |
| RBF | 15 | 5 | 131 | 0.6455 | 0.60007 |
| RBF | 15 | 6 | 131 | 0.70396 | 0.61538 |
| Nikkei 225 Stock Average. | | | | | |
| MLP | 18 | 6 | 103 | 0.61538 | 0.63052 |
| MLP | 18 | 7 | 103 | 0.63052 | 0.63052 |
| MLP | 19 | 7 | 118 | 0.61538 | 0.675 |
| MLP | 19 | 5 | 128 | 0.63052 | 0.61538 |
| MLP | 19 | 6 | 128 | 0.66032 | 0.61538 |
| RBF | 15 | 5 | 64 | 0.56885 | 0.63052 |
| RBF | 17 | 8 | 95 | 0.53674 | 0.55292 |
| RBF | 14 | 6 | 138 | 0.53674 | 0.60007 |
| RBF | 14 | 8 | 138 | 0.53674 | 0.53674 |
| RBF | 18 | 6 | 138 | 0.53674 | 0.55292 |

*4) Comparison of the various networks developed and the selection of the superior network.*

This stage of development entailed the comparison of the various ANNs that were created. It also involves the selection of the best networks to forecast the various indices considered.

Committees of MLP as well as committees of RBF networks were also developed. It has been stated that ANNs utilized simultaneously, as committees, will provide an average error that is lower than any individual network [16]. As a result, a combination of networks as a classifier should outperform a single network classifier.

The 5 most accurate networks from the input time window optimization and the moving averages investigation were used in the committees for the indices concerned. The outputs of these networks were fed into a voting system. The voting system determined the final output of the committee. If the majority of the ANNs within the committee classified an output into a certain class, the voting system would classify the output of the committee as the class. If 2 of the networks within the committee classified an output into the same class and another 2 of the networks classified their outputs into a different class, the voting system would classify the output of the committee as undecided. Table VII illustrates the results of this investigation.

TABLE VII:
RESULTS OF COMMITTEE ARCHITECTURES.

| Network architecture. | Networks within committee. | | | Accuracy (Validation). | Accuracy (Test). |
|---|---|---|---|---|---|
| | Input days. | Day moving average. | Hidden nodes. | | |
| **Dow Jones Industrial Average.** | | | | | |
| MLP | 15 | 5 | 64 | 0.7209 | 0.7493 |
| | 15 | 7 | 64 | | |
| | 15 | 8 | 64 | | |
| | 16 | 5 | 58 | | |
| | 16 | 6 | 58 | | |
| RBF | 16 | 6 | 55 | 0.6187 | 0.5454 |
| | 17 | 5 | 106 | | |
| | 18 | 5 | 73 | | |
| | 18 | 7 | 116 | | |
| | 18 | 8 | 116 | | |
| **JSE All Share.** | | | | | |
| MLP | 13 | - | 26 | 0.704 | 0.6775 |
| | 18 | - | 26 | | |
| | 18 | - | 35 | | |
| | 18 | - | 96 | | |
| | 19 | - | 96 | | |
| RBF | 8 | - | 12 | 0.522 | 0.6024 |
| | 9 | - | 18 | | |
| | 12 | - | 18 | | |
| | 13 | - | 16 | | |
| | 18 | - | 22 | | |
| **NASDAQ 100.** | | | | | |
| MLP | 18 | 6 | 119 | 0.6826 | 0.6947 |
| | 19 | 5 | 135 | | |
| | 19 | 7 | 135 | | |
| | 19 | 5 | 148 | | |
| | 19 | 6 | 148 | | |
| RBF | 15 | 5 | 131 | 0.7195 | 0.6051 |
| | 15 | 6 | 131 | | |
| | 19 | 5 | 64 | | |
| | 19 | 6 | 64 | | |
| | 19 | 5 | 99 | | |
| **Nikkei 225 Stock Average.** | | | | | |
| MLP | 18 | 6 | 103 | 0.6529 | 0.6402 |
| | 18 | 7 | 103 | | |
| | 19 | 7 | 118 | | |
| | 19 | 5 | 128 | | |
| | 19 | 6 | 128 | | |
| RBF | 14 | 6 | 138 | 0.5801 | 0.5618 |
| | 14 | 8 | 138 | | |
| | 15 | 5 | 64 | | |
| | 17 | 8 | 95 | | |
| | 18 | 6 | 138 | | |

It is evident, from the investigations conducted, that the networks illustrated in Table VIII resulted in the most accurate forecasting classifiers.

TABLE VIII:
NETWORKS SELECTED.

| Network architecture. | Networks within committee. | | | Accuracy (Validation). | Accuracy (Test). |
|---|---|---|---|---|---|
| | Input days. | Day moving average. | Hidden nodes. | | |
| **Dow Jones Industrial Average.** | | | | | |
| MLP | 15 | 5 | 64 | 0.7209 | 0.7493 |
| | 15 | 7 | 64 | | |
| | 15 | 8 | 64 | | |
| | 16 | 5 | 58 | | |
| | 16 | 6 | 58 | | |
| **JSE All Share.** | | | | | |
| MLP | 13 | - | 26 | 0.704 | 0.6775 |
| | 18 | - | 26 | | |
| | 18 | - | 35 | | |
| | 18 | - | 96 | | |
| | 19 | - | 96 | | |
| **Nikkei 225 Stock Average.** | | | | | |
| MLP | 18 | 6 | 103 | 0.6529 | 0.6402 |
| | 18 | 7 | 103 | | |
| | 19 | 7 | 118 | | |
| | 19 | 5 | 128 | | |
| | 19 | 6 | 128 | | |
| **NASDAQ 100.** | | | | | |
| Network architecture. | Input days. | Day moving average. | Hidden nodes. | Accuracy (Validation). | Accuracy (Test). |
| MLP | 19 | 5 | 148 | 0.68954 | 0.70396 |

It has also been determined that the committee of networks does not always result in a more accurate solution. This is true for the forecasting of the closing price performance of the Nasdaq 100 index. Another conclusion that could be drawn is that the MLP network architecture is particularly suited for this application. The difference in accuracy between the 2 network architectures is approximately 10% for the forecasting of the Dow Jones Industrial Average and JSE All Share indices. However, the difference in accuracy between these network topologies is approximately 5% for the forecasting of the Nasdaq 100 and the Nikkei 225 Stock Average indices.

### *B. The Stock Quantity Selection Component.*

The data used to develop the SQSC module has been generated based on the 4 forecasted closing price performance classes illustrated in Table I. The development process was divided into various stages. The following procedure has been pursued in the creation of the various classifiers employed:

1. Selection and processing of data to be used by the classifiers during training, validation and testing.
2. Optimization of the classification threshold of the various classes to be categorized.
3. Optimization of the classifier architectures.
4. Comparison of the various classifiers developed

and the selection of the superior model.

The remainder of this section will elaborate on the various stages of implementation mentioned above.

*1) Selection and processing of data.*

The data utilized in developing and testing the various classifiers has been created by analyzing all the possible combinations of the 4 forecasted closing price performance classes. As a result, the entire data set consisted of 256 unique data records.

In order to present the forecasted closing price performance classes to the classifiers, a binary notation is employed. These inputs are presented to the classifier using 4 inputs. This input representation format is used for all indices considered. A similar binary notation scheme is also utilized to present the investment recommendation outputs. Table IX illustrates the manner in which the inputs and outputs of the classifier are to be interpreted.

As previously mentioned, 3 designs of the SQSC module were considered. The design that employed 4 classifiers utilized the above input output representation format. However, the design that used 16 classifiers only employed the above input representation format. As stated earlier, the classifiers of this design have 1 output that indicates whether or not to execute an investment recommendation.

The design that utilized a single classifier has 16 inputs and 16 outputs. The input representation format for this design is the same as above. However, the first group of 4 inputs corresponds to the forecasted performance of the Dow Jones Industrial Average index. Similarly, the second, third and fourth group of 4 inputs characterizes the forecasted performance of the JSE All Share, Nasdaq 100 and Nikkei 225 Stock Average indices, respectively. The outputs are to be interpreted in a similar manner.

The data is divided into a training, validation and test set. During the implementation of all 3 designs considered, the training data set consisted of all data records where the inputs were classified into 2 of the 4 closing price performance classes. However, the models developed were validated and tested with the remaining possible closing price performance class combinations.

Dividing the data set into 3 portions assists in ensuring that over-fitting as well as under-fitting has been avoided during the development of the ANNs. As mentioned earlier, the training data set is used to train the ANN to find the general pattern between its inputs and outputs. The validation data set is used to assess the network and the test data is employed to confirm the classification quality of the developed model.

The training data set is used to create the cluster centers within the FISs. However, the validation and test data sets are utilized to assess the classification ability of the inference systems.

Due to the binary representation of the inputs and outputs, normalization of the data is not required.

TABLE IX:
CLASSIFIER INPUT AND OUTPUT REPRESENTATION.

Classifier Inputs.

| Input. | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| Large Rise (LR). | 1 | 0 | 0 | 0 |
| Slight Rise (SR). | 0 | 1 | 0 | 0 |
| Slight Drop (SD). | 0 | 0 | 1 | 0 |
| Large Drop (LD). | 0 | 0 | 0 | 1 |

Classifier Outputs.

| Output. | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| Sell stocks in possession at the next day closing price. | 0 | 1 | 0 | 0 |
| Hold current investment position. | 1 | 0 | 0 | 0 |
| Buy stocks to the value of 15 % of available trading capital at the next day closing price. | 0 | 0 | 1 | 0 |
| Buy stocks to the value of 25 % of available trading capital at the next day closing price. | 0 | 0 | 0 | 1 |

*2) Optimization of the classification threshold.*

MLP and RBF neural network architectures were utilized in the classification of investment recommendations.

As mentioned earlier, it has been stated that a network with 1 hidden layer, provided with sufficient data, can be used to model any function [40]. As a result, the ANN architectures employed consisted of only 1 hidden layer.

The hyperbolic tangent activation function has been utilized within the MLP network hidden layer, as this function offers a practical advantage of faster convergence during training [17]. As mentioned previously, it has been stated that the appropriate selection of the output layer activation function for a classification problem is the logistic sigmoidal function [17]. As a result, this function has been employed within the output layer of the MLP network. The RBF networks that have been developed contained a Gaussian activation function within its hidden layer and a linear activation function within its output layer.

The FISs developed utilized subtractive clustering to create the required membership functions and set of fuzzy inference rules. During this stage of implementation, the number of hidden nodes within the ANNs and the cluster radius utilized by the cluster centers within the FISs were assumed to be arbitrary. This will be optimized at a later stage of development. During this stage of development, the number of hidden nodes within the ANNs as well as the cluster radius utilized by the FISs was 10 and 0.5, respectively. This stage of implementation involved the optimization of the interpretation of the classifiers. As a result, this involved the selection of an appropriate classification threshold value that would yield the most accurate results.

The classification threshold has been optimized by minimizing an error function that mapped the classification thresholds to the accuracy of the developed classifiers. The process has been performed on the validation and test data sets.

Since this is a classification implementation, the accuracy of the models can no longer be calculated using the sum of square error of the difference between the target and investment recommendation classifier output. Instead a confusion matrix is utilized to identify the number of true and false classifications that are generated by the models developed. This is then used to calculate the true accuracy of the classifiers, using equation (1).

The classification threshold was optimized by initially creating classifiers utilizing a threshold value of 0.5. This implies that if the classifier outputs a value less than 0.5, the output will be regarded as a 0. Similarly, if the output value is larger than or equal to 0.5, the output will be interpreted as a 1. This threshold value of 0.5 proved to be adequate for the MLP networks as well as the FISs implementations. The threshold value resulted in 100% accurate classifications. This has been demonstrated on the training, validation and test data sets.

However, the RBF network used in the design, which employed a single classifier that has 16 inputs and 16 outputs, did not perform well utilizing this threshold value. As a result, the classification threshold of this model had been varied from 0.1 to 0.5 in iterations of 0.01. For each of the threshold values, a classification hit rate and a classification false alarm rate were calculated using equations (3) and (4).

$$hit\ rate = \frac{TP}{TP + FN}, \tag{3}$$

$$false\ alarm\ rate = \frac{FP}{FP + TN}, \tag{4}$$

where

$TP$ is the true positive (1 classified as a 1),

$TN$ is the true negative (0 classified as a 0),

$FN$ is the false negative (1 classified as a 0),

$FP$ is the false positive (0 classified as a 1).

These were then plotted to generate a Receiver Operating Characteristic (ROC) curve. The area under the curve indicates the classification capability of the network. An area that is close to unity suggests excellent discrimination capabilities and an area that is close to 0.5 shows poor results. Table X illustrates the threshold values that resulted in the largest accuracy value for the validation and test data sets. Fig. 2 illustrates the ROC curves for the Dow Jones Industrial Average index. Similar results were also attained for all indices considered. The threshold value of 0.5 proved to be satisfactory for design 2 and design 3 RBF classifiers.

TABLE X:
RESULTS OF VARIED CLASSIFICATION THRESHOLD FOR RBF CLASSIFIER.

| | Classification thresholds | | | |
|---|---|---|---|---|
| Class. | Dow Jones Industrial Average. | JSE All Share. | NASDAQ 100. | Nikkei 225 Stock Average. |
| Sell stocks in possession at the next day closing price. | 0.19 | 0.23 | 0.20 | 0.11 |
| Hold current investment position. | 0.17 | 0.10 | 0.12 | 0.17 |
| Buy stocks to the value of 15 % of available trading capital at the next day closing price. | 0.19 | 0.16 | 0.13 | 0.17 |
| Buy stocks to the value of 25 % of available trading capital at the next day closing price. | 0.24 | 0.17 | 0.17 | 0.19 |

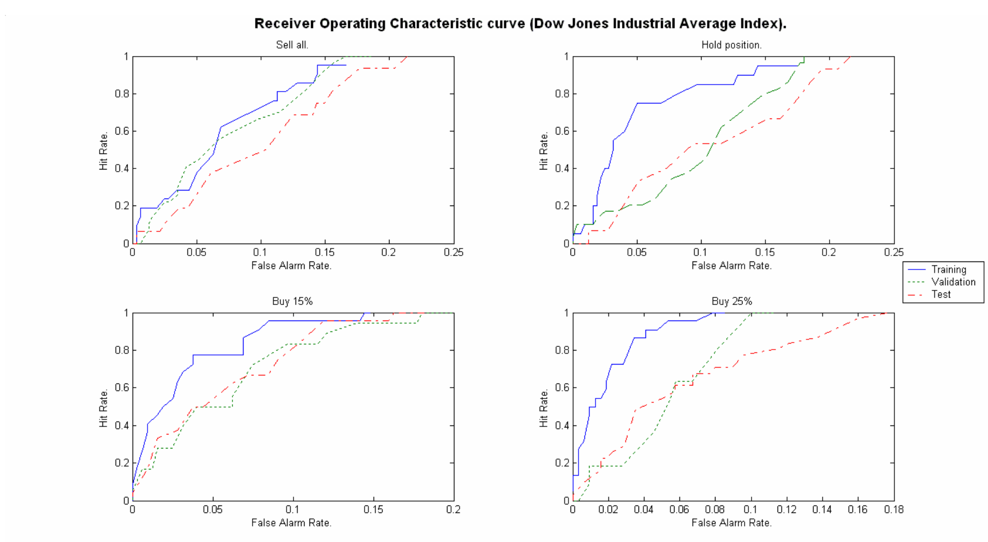


Fig. 2. Results of classification threshold optimization.

3) *Optimization of the classifier architectures.*

This stage of implementation involved the optimization of the ANN and Fuzzy Inference System (FIS) architectures. As a result, this step of development involved the selection of the correct number of hidden neurons that would yield the most accurate results. It also entailed selecting the correct cluster radius that would concede the largest investment recommendation classification accuracy.

The number of hidden neurons or nodes has been optimized by minimizing an error function that mapped the number of hidden nodes to the accuracy of the developed network. The process was performed on the validation and test data sets.

The hidden nodes were optimized by creating various MLP and RBF ANNs with hidden nodes of 1 to 75. As a result, 150 ANNs were developed. These developed networks employed the classification thresholds stated in the previous section. The networks also utilized the same activation functions mentioned in the previous section.

Utilizing the training data set, these networks are trained. The validation and test data are then presented to the ANN. Thereafter, the accuracies for the training, validation and test data sets are calculated. When presented with the validation and test data, ANNs that resulted in the largest accuracy were analyzed.

Fig. 3 illustrates the results of the design that employed 16 classifiers. Similar results were achieved for the other design implementations considered.

The investigation revealed that a MLP and RBF network, used in the design that employed a single classifier, with number of hidden nodes larger than 12 and 52, respectively, yield 100% accurate models for categorizing the investment recommendations appropriately. The investigation also determined that MLP and RBF ANNs, utilized in the design that employed 4 classifiers, with number of hidden nodes greater than 2 and 5, respectively, achieved the same results. Similar results were obtained with MLP and RBF networks that contained more than 1 hidden neuron. These networks were used in the design that employed 16 classifiers.

The cluster radius indicates the range of influence of a cluster. A small cluster radius results in small clusters in the data and, therefore, many fuzzy rules. Large cluster radii yield few large clusters in the data and, hence, fewer fuzzy rules [30].

The cluster radius has been optimized by minimizing an error function that mapped the radius to the accuracy of the developed inference systems. This process was performed on the validation and test data sets.

During this step of implementation, the optimization process entailed the construction of various inference systems with the cluster radius ranging from 0.01 to 1.

The investigation determined that FISs, employed in the design utilizing 4 classifiers, with a cluster radius equal to or greater than 0.01 achieve 100% accuracy in categorizing the investment recommendations appropriately. However, the FIS, employed in the design that used 1 classifier, did not achieve 100% investment recommendation classification accuracies. It has been determined that a cluster radius of 0.11 achieved the most accurate results. The lowest accuracy value attained was 83%. The largest accuracy value was 100%. Fig. 4 illustrates the results of the optimization of the cluster radius investigation.

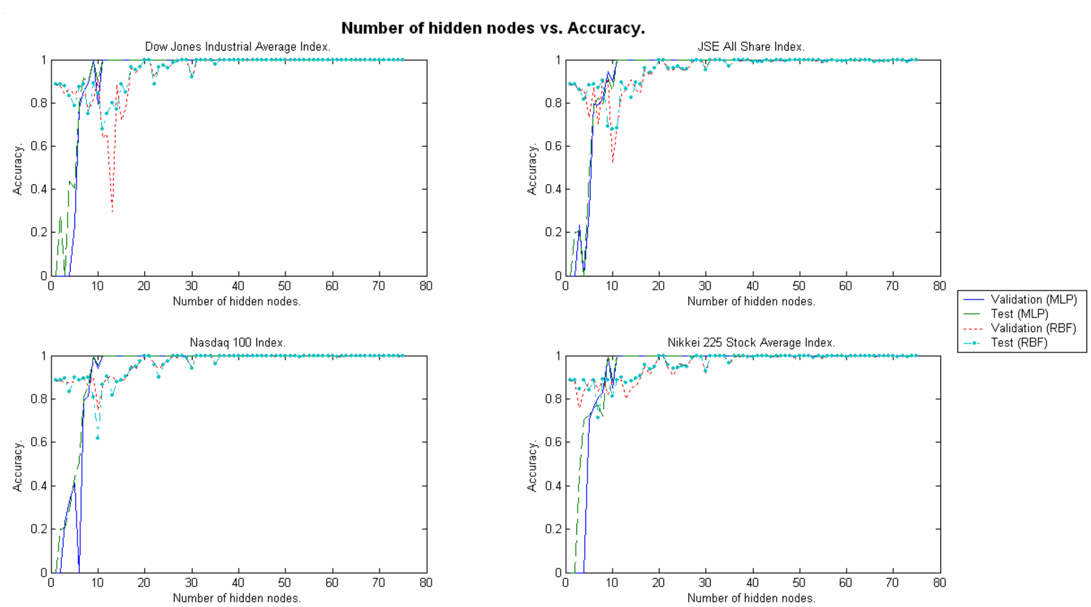


Fig.3. Results of varied number of hidden nodes.

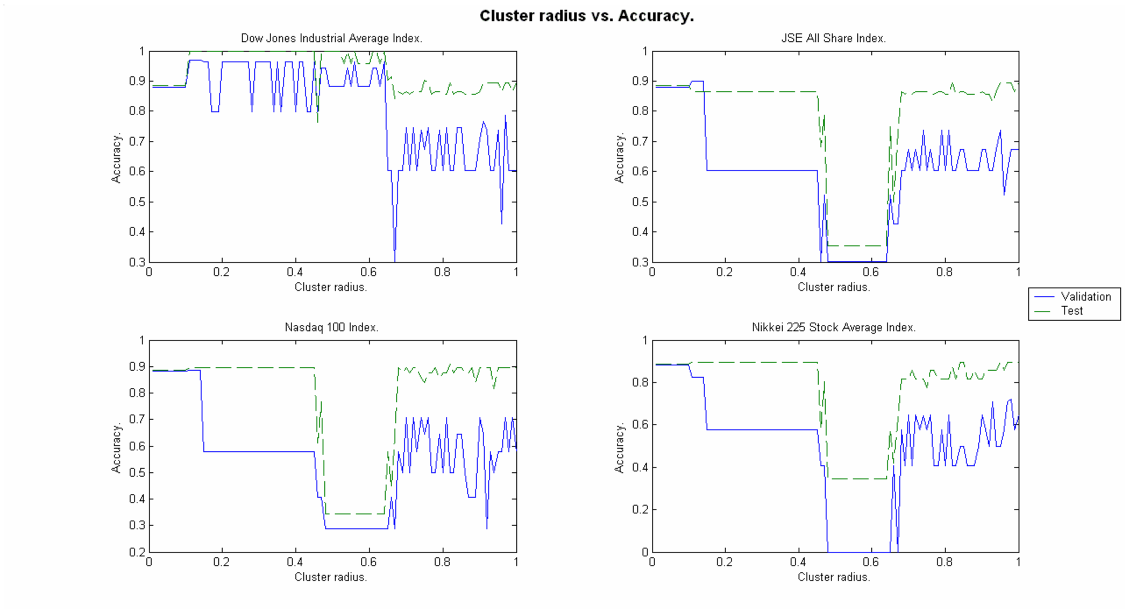


Fig. 4. Results of varied cluster radius.

*4) Comparison of the various designs implemented and the selection of the superior model.*

This stage of implementation entailed the comparison of the various designs that were developed. It also involves the selection of the best design to classify the investment recommendations.

Table XI illustrates the various models that have been created. The above designs have been compared in terms of their complexity as well as scalability. Complexity, in this context, is defined as the number of classifiers employed by the design. Scalability is defined as the ability of the design to accommodate the classification of additional investment recommendations.

It is evident that the design, which employed 1 classifier, has low complexity and low scalability. When additional investment recommendations are to be added to the component, the classifier employed is to be re-trained. However, the design that utilized 4 classifiers has low complexity as there are few classifiers used. This design also has high scalability. It is not required to re-create the existing classifiers, when additional recommendations are added. It is evident that the design, which employed 16 classifiers, has high complexity. In order to add investment recommendations to this design, the existing classifiers do not have to be re-created. As a result, the design has high scalability.

Due to the above analysis, the design that consisted of 4 classifiers is most appropriate for this application. It does not employ many classifiers and the design does not require re-work when additions are to be made.

It is evident from Table XI that both the ANN and FIS implementations of this design perform satisfactorily. As a result, either of the classifier architectures could be used.

TABLE XI:
CLASSIFIERS SELECTED.

| Classifier topology. | Fuzzy rules/ Hidden nodes. | Membership functions. | Accuracy (Training). | Accuracy (Validation). | Accuracy (Test). |
|---|---|---|---|---|---|
| 1 Classifier design. | | | | | |
| MLP | 12 | - | 100 | 100 | 100 |
| RBF | 52 | - | 100 | 100 | 100 |
| FIS | 85 | 1360 | 100 | 83 | 87 |
| 4 Classifier design. | | | | | |
| MLP | 2 | - | 100 | 100 | 100 |
| RBF | 5 | - | 100 | 100 | 100 |
| FIS | 4 | 16 | 100 | 100 | 100 |
| 16 Classifier design. | | | | | |
| MLP | 1 | - | 100 | 100 | 100 |
| RBF | 1 | - | 100 | 100 | 100 |
| FIS | 4 | 16 | 100 | 100 | 100 |

## IV. THE SYSTEM.

In the previous section, the 2 major components of the application were implemented and the superior designs as well as models were selected. As a result, the final system utilizes the classifiers illustrated in Fig. 5. It should be noted that the RBF or the FIS implementation of the SQSC module could also be used in the system.

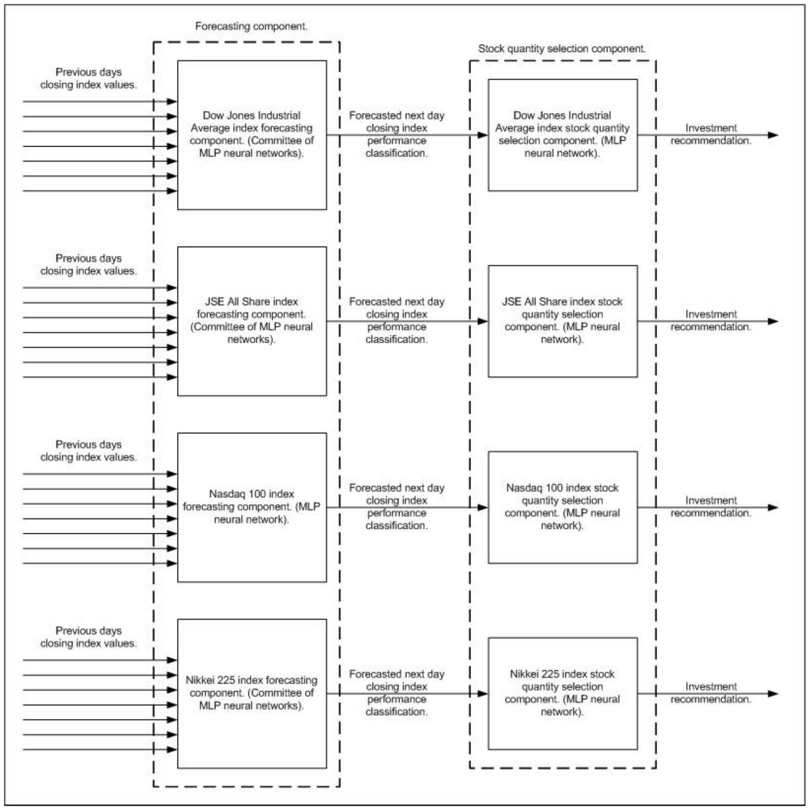


Fig. 5. The forecasting and stock quantity selection system.

Examining Table VIII and XI, it can be concluded that the final application is 72% accurate in the forecasting of the closing price performance as well as in the recommending of the quantity of stocks to purchase and sell for index funds based on the Dow Jones Industrial Average index. Similarly, the application is 68%, 69% and 64% accurate in relation to the JSE All Share, Nasdaq 100 and Nikkei 225 Stock Average indices, respectively.

## V. CONCLUSION.

This project entailed the development of a system that could estimate the next day closing index price performance as well as recommend the actions to be executed based on these estimates. The Dow Jones Industrial Average, the JSE All Share, the Nasdaq 100 and the Nikkei 225 Stock Average indices were considered.

The development methodology utilized in the creation of the FC module involved, initially, varying the number of hidden nodes within the ANNs. This resulted in creating acceptable network architectures. Thereafter, the numbers of closing prices of the indices for the previous days as inputs to the networks were varied. Moving averages were also introduced as inputs to the networks to reduce the noise within the data. Acceptable forecasting classification accuracies were achieved. The best and worst accuracy levels obtained were 72% and 64%, respectively. These accuracy levels were attained for the Dow Jones Industrial Average and the Nikkei 225 Stock Average indices, respectively. As a result, it can be concluded that the univariate approach of the forecasting of indices is relevant and can result in highly accurate solutions.

The accuracy of these performance classifications could be improved by using complex committee of classifiers. It could also be improved by employing a genetic algorithm to create the optimal ANN architecture. The genetic algorithm could also be used to optimize the appropriate number of closing prices of the indices for the previous days as inputs to the networks.

Various designs of the SQSC module were considered. The responsibility of this module entailed the categorization of investment recommendations, based on the forecasted performance of indices, appropriately. Designs that utilized 1, 4 and 16 classifiers were implemented. The development methodology employed in the creation of these designs, initially, involved the selection of appropriate classification thresholds. Thereafter, the number of hidden nodes within the ANNs as well as the cluster radius of the cluster centers within the FISs was varied. This resulted in creating acceptable classifier architectures. Acceptable investment recommendation classification accuracies were achieved.

The designs were compared in terms of complexity as well as scalability. Complexity is concerned with the number of classifiers that are used within the design. Scalability is the ability of the design to accommodate the classification of additional investment recommendations. The design that employed 4 classifiers has low complexity and high scalability. Each of the classifiers utilized in the design consisted of 4 inputs and 4 outputs. This design is most appropriate for the application of concern.

## REFERENCES.


[1] Yahoo Finance, last accessed: 4 April 2006. [Online]. Available: http://finance.yahoo.com.

[2] E. E. Peters, *Chaos and order in the capital markets: A new view of cycles, prices and market volatility, second edition.*, John Wiley & Sons Inc, 1996.

[3] S. Taylor, *Modeling financial time series.*, John Wiley & Sons, 1986.

[4] J. Jarrett, E. Kyper, "Evidence on the seasonality of stock market prices of firms traded on organized markets.", Taylor & Francis, *Applied Economics Letters.*, vol.12, no. 9, pp. 537-543, 2005.

[5] J. S. Ang, R. A. Pohlman, *A note on the price behavior of Far eastern stock.*, J. Int. Business Studies, 1978.

[6] B. H. Solnik, *Note on the validity of the random walk for European stock prices*, J. Finance, 1973.

[7] J.P.C Kleijnen. *Statistical Techniques in Simulation*, New York: Marcel Dekker, 1974.

[8] SH. Chun, YJ. Park, "Dynamic adaptive ensemble case-based reasoning: application to stock market prediction.", *European Systems with Applications.*, vol. 28, issue 3, pp. 435 - 443, 2005.

[9] M. E. Azoff, *Neural network time series forecasting of financial markets.*, John Wiley & Sons Ltd, 1994.

[10] E. Gately, *Neural networks for Financial Forecasting.*, John Wiley & Sons Ltd, 1996.

[11] A. N. Refenes, Y. Bentz, "Neural networks in financial engineering: A study in methodology.", *IEEE Transaction on Neural Networks.*, vol. 8, issue 6, pp. 1222 - 1267, 1997.

[12] D. N. Chorafas, *Chaos theory in the financial markets: Applying fractals, fuzzy logic, genetic algorithms, Swarn simulation & the Monte Carlo method to manage markets.*, Probus Publishing Company, 1994.

[13] I. L. MacDonald, W. Zucchini, *Hidden Markov and other models for discrete-valued time-series.*, Chapman & Hall, 1997.

[14] W. Leigh, M. Paz, R. Purvis, "An analysis of a hybrid neural network and pattern recognition technique for predicting short-term increases in the NYSE composite index.", *Omega.*, vol. 30, issue 2, pp. 69-76, 2002.

[15] D. V. Zhora, "Data preprocessing for stock market forecasting using random subspace classifier network.", *Proceedings of International Joint Conference on Neural Networks.*, pp. 2549-2554, Montreal, Canada, 2005.

[16] C. M. Bishop, *Neural Networks for Pattern Recognition.*, Oxford University Press, 1995.

[17] I. T. Nabney, *Netlab: Algorithms for Pattern Recognition.*, Springer, 2002.

[18] J. M. Keller, M. Gray and J. Givens, "A fuzzy k-nearest neighbor algorithm.", IEEE Transaction on System, Man and Cybernetics., vol. 15, no. 4, pp. 580-585, 1985.

[19] J. C. Bezdek, *Pattern Recognition with Fuzzy Objective Function Algorithms.*, Plenum, 1981.

[20] H. Ishibuchi, K. Nozaki, N. Yamamoto and H. Tanaka, "Construction of fuzzy classification systems with rectangular fuzzy rules using genetic algorithms.", *Fuzzy Sets and Systems.*, vol. 65, no. 2-3, pp. 237-253, 1994..

[21] H. Ishibuchi, K. Nozaki, N. Yamamoto and H. Tanaka, "Selecting fuzzy if-then rules for classification problems using genetic algorithms.", *IEEE Transactions on Fuzzy Systems.*, vol. 3, no. 3, pp. 260-270, 1995

[22] M. Russo, "FuGeNeSys – A fuzzy genetic neural system for fuzzy modeling.", *IEEE Transaction on Fuzzy Systems.*, vol. 6, no. 3, pp. 373-388, 1998..

[23] F. Masulli, F. Casalino and F. Vannucci, "Bayesian properties and performances of adaptive fuzzy systems in pattern recognition problems.", *Proceedings of the European Conference on Artificial Neural Networks.*, *ICANN-94.*, pp. 189-192, 1994.

[24] D. Nauck and R. Kruse, "A neuro-fuzzy method to learn fuzzy classification rules from data.", *Fuzzy Sets and Systems.*, vol. 89, no. 3, pp. 277-288, 1997.

[25] C-T. Sun and J-S. R. Jang, "A neuro-fuzzy classifier and its applications.", *Proceedings of the IEEE International Conference on Fuzzy System.*, vol. 1, pp. 94-98, 1993.

[26] C-W. Hsu and C-J. Lin, "A comparison of methods for multi-class support vector machines", *IEEE Transactions on Neural Neworks.*, vol. 13, no. 2, pp. 415-425, 2002.

[27] J. H. Friedman, "Another approach to polychotomous classification.", *Department of Statistics, Stanford University.*, Technical report, 1996.

[28] J. Schurmann, *Pattern Classification. A Unified View of Statistical and Neural Principles.*, John Wiley & Sons Inc, 1996.

[29] B. Zadrozny, "Reducing multiclass to binary by coupling probability estimates.", *Neural Information Processing Systems Foundation 14*, pp. 1041-1048, 2002.

[30] S. Chiu, "Fuzzy Model Identification Based on Cluster Estimation.", *Journal of Intelligent and Fuzzy Systems.*, vol. 2, no. 3, pp. 267-278, 1994.

[31] J-S. R. Jang, "ANFIS: Adaptive-Network-Based Fuzzy Inference System.", *IEEE Transaction on System, Man and Cybernetics.*, vol. 23, no. 3, pp. 665-685, 1993.

[32] Investments guide, last accessed: 4 April 2006. [Online]. Available: http://www.investments-guide.com.

[33] P. K. H. Phua, X. Zhu, C. H. Koh, "Forecasting stock index increments using neural networks with trust region methods.", *Proceedings of International Joint Conference on Neural Networks*, vol. 1, pp. 260-265, 2003.

[34] B. Leke and T. Marwala. "Optimization of the stock market input time-window using Bayesian neural networks." *Proceedings of the IEEE International Conference on Service Operations, Logistics and Informatics*, pp. 883-894, Beijing, China, 2005.

[35] A. J. Conway, K. P. Macpherson, J. C. Brown, "Delayed time series predictions with neural networks.", *Neurocomputing.*, vol. 18, issue 1-3, pp. 81-89, 1998.

[36] B. W. Wah, M. Qian, "Constrained formulations and algorithms for stock-price predictions using recurrent FIR neural networks.", *Proceedings of the Eighteenth National Conference on Artificial Intelligence and Fourteenth Conference on Innovative Applications of Artificial Intelligence.*, *AAAI Press.*, pp. 211-216, 2002.

[37] R. Heinkel, A. Kraus, "Measuring event impacts in thinly traded stocks.", *The Journal of Financial and Quantitative Analysis.*, vol. 23, no. 1, pp. 71-88, 1988.

[38] J. Yao, C. L. Tan, HL. 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.

[39] J. Yao, C. L. Tan, "Guidelines for financial forecasting with neural networks.", *Proceedings of International Conference on Neural Information Processing.*, Shanghai, China, pp. 757-761, 2001.

[40] R. Beale, T. Jackson., *Neural Computing-An introduction.*, Taylor & Francis, 1990.

[41] E. B. Baum, D. Haussler., "What size net gives valid generalization?", *Neural Computation*, vol. 1, pp. 81-90, 1989.

[42] J. Cory, "Technical analysis tutorial", *Investopedia.com*, last accessed 4 April 2006. [Online]. Available: http://www.investopedia.com/university/technical/

# Conclusion.

Due to the volatile nature of the world economies, it is crucial that individuals invest their earnings for future necessities. This research proposes an application that could assist investors in making such financial decisions. It is anticipated that this application would be used in conjunction with other financial analysis tools. As a result, the proposed system should be employed to confirm an investment decision.

This research entailed the development of an application that could forecast the next day closing index price performance as well as recommend the actions to be executed based on these estimates. The Dow Jones Industrial Average, the JSE All Share, the Nasdaq 100 and the Nikkei 225 Stock Average indices were considered.

As the application is to be utilized in assisting an investor in making a financial decision, the system is based on a profitable trading strategy. This research focuses on the "Buy low, sell high" trading strategy. The strategy has been implemented as well as compared to the "Buy and hold" trading strategy in terms of profits generated. The "Buy low, sell high" trading strategy entails purchasing certain stocks at a low price and selling these stocks when the price is high. The "Buy and hold" trading strategy, as the name suggests, involves an investor purchasing certain stocks and retaining them for a particular duration. It has been determined that the "Buy low, sell high" trading strategy is most profitable. As a result, the system has been based on this trading strategy.

The developed application consisted of a Forecasting Component (FC) and a Stock Quantity Selection Component (SQSC). As the name suggests, the FC module generated predictions of the next day closing price performance for the indices considered. The Multi-Layer Perceptron (MLP) and Radial Basis Function (RBF) neural network architectures were utilized in the development of this component. The SQSC module produced investment recommendations based on the FC module predictions. Computational intelligent methods such as artificial neural networks (ANNs), Fuzzy Inference Systems (FISs) as well as Adaptive Neuro-Fuzzy Inference Systems (ANFISs) were utilized in the creation of this component.

There are 2 approaches to the forecasting of indices. A univariate model employs the past values of the time series to generate a prediction. A multivariate model utilizes additional information such as market indicators or fundamental factors of companies as inputs. Several research outcomes that employed the univariate approach, have illustrated acceptable results. As a result, the FC module used the univariate approach.

The development methodology utilized in the creation of the FC module involved, initially, varying the number of hidden nodes within the ANNs. This resulted in creating acceptable network architectures. Thereafter, the numbers of closing prices of the indices for the previous days as inputs to the networks were varied. Moving averages were also introduced as inputs to the networks to reduce the noise within the data. The most accurate networks developed were employed in a committee of neural networks. Acceptable forecasting classification accuracies were achieved. It has been determined that the MLP neural network architecture is particularly suited for this application. Accuracies of 72%, 68%, 69% and 64% were obtained for the prediction of closing price performance of the Dow Jones Industrial Average, JSE All Share, Nasdaq 100 and Nikkei 225 Stock Average indices, respectively. The difference in accuracy between the MLP and RBF networks is approximately 10% for the forecasting of the Dow Jones Industrial Average and JSE All Share indices. However, the difference in accuracy between these network topologies is approximately 5% for the forecasting of the Nasdaq 100 and the Nikkei 225 Stock Average indices. As a result, it can be concluded that the univariate approach of the forecasting of indices is relevant and can result in highly accurate solutions.

The accuracy of these performance classifications could be improved by using complex committee of classifiers. It could also be improved by employing a Genetic Algorithm (GA) to create the optimal neural network architecture. The GA could also be used to optimize the appropriate number of closing prices of the indices for the previous days as inputs to the networks.

Various classification designs of the SQSC module were considered. Designs that utilized 1, 4 and 16 classifiers were implemented. The development methodology employed in the creation of these designs, initially, involved the selection of appropriate classification thresholds. Thereafter, the number of hidden nodes within the ANNs as well as the cluster radius of the cluster centers within the FISs had been varied. This resulted in creating acceptable classifier architectures. Investment recommendation classification accuracies of 100% were achieved.

The SQSC module designs were compared in terms of complexity as well as scalability. Complexity is concerned with the number of classifiers that are employed within the design. Scalability is the ability of the design to accommodate the classification of additional investment recommendations. The design that employed 4 classifiers has low complexity and high scalability. Each of the classifiers utilized in the design consisted of 4 inputs and 4 outputs. This design is most appropriate for the application of concern.

The final system consisted of a FC module that employed 3 committees of MLP neural networks to predict the closing price performance of the Dow Jones Industrial Average, JSE All Share and Nikkei 225 Stock Average indices. The FC module within the final system utilized a MLP network to forecast the closing index price performance of the Nasdaq 100 index. The system consisted of a SQSC module that employed 4 MLP neural network investment recommendation classifiers. It should be noted that the RBF or Fuzzy Inference System implementations of the design could also be utilized.

# Appendix A: Paper submitted to 2006 IEEE International Conference on Systems, Man and Cybernetics.

# Forecasting closing price indices using neural networks.

P. B. Patel, *Member, IEEE* and T. Marwala,, *Member, IEEE.*

*Abstract*—**Accurate financial prediction is of great practical interest to both individual and institutional investors. This paper proposes an application, which employs artificial neural networks that could be used to assist investors in making financial decisions. The Multi-layer perceptron as well as Radial Basis Function neural network architectures are implemented as classifiers to forecast the closing index price performance. Categorizes that these networks classify are based on a profitable trading strategy that outperforms the long-term "Buy and hold" trading strategy. The Dow Jones Industrial Average, Johannesburg Stock Exchange All Share, Nasdaq 100 and the Nikkei 225 Stock Average indices are considered. The best and worst forecasting classification accuracies obtained were 72% and 64%, respectively. These accuracy levels were attained for the Dow Jones Industrial Average and the Nikkei 225 Stock Average indices, respectively.**

## I. INTRODUCTION.

Trading in stock market indices has gained exceptional popularity in major financial markets worldwide. Due to the increasing diversity of financial index related instruments as well as the economic growth experienced during the past years, the extent of global investment opportunities for both individual and institutional investors has broadened [1]. As a result, it is of critical importance that applications which assist investors in making financial decisions be developed.

The aim of this research is to develop a prediction application, using computational intelligent methods, which could assist investors in making financial decisions. It is anticipated that this application would be used in conjunction with other financial analysis techniques. As a result, this application should be used to confirm an investment decision.

There are many funds that are highly correlated with indices [1]. As a result, the developed system is concerned with the Dow Jones Industrial Average, Johannesburg Stock Exchange or the JSE Securities Exchange (JSE) All Share, Nasdaq 100 and the Nikkei 225 Stock Average indices. This approach would assist in creating a diversified portfolio, as these indices are concerned with different industry sectors. A diversified portfolio is concerned with, among other aspects, investing in a wide range of stocks, instead of having the majority of trading capital invested in 1 particular share. It is a risk management technique, which ideally results in a lower risk portfolio.

Predicting stock market performance has been a major research area for many years. There are various schools of thought in terms of the ability to profit from the equity markets. Some believe that no investor can obtain above average trading advantages based on the historical and present information. The Random Walk Hypothesis (RWH) states that prices on the stock market wander in a purely random and unpredictable manner [2]. As a result, according to this theory, every price change occurs without any influence from past prices. The Efficient Market Hypothesis (EMH) states that the markets incorporate all available information and prices are adjusted immediately once new information becomes available [2]. If these theories are true, there should not be any advantage in predicting stock performance, as the market would react and compensate for any actions performed due to the predicted information.

These theories have been met with a great deal of opposition. The argument against the EMH is that many investors base their expectations on past prices, past earnings, track record as well as other indicators. Since stock prices are largely influenced by investor expectations, many believe it only makes sense that past prices do affect future prices.

Compelling evidence has also been provided that rejects the RWH [3]. It has been illustrated that stock market price movements, of the United States [4] as well as Japan [5], have conformed only to the weak form of the EMH. There has also been a study of 234 stocks from 8 major European stock markets, which indicated that these stock markets exhibited a slight departure from the RWH [6]. As a result, the above offers encouragement for research into developing market prediction applications.

Traditionally, moving average, exponential smoothing and linear regression statistical methods have been used in the prediction of stock prices [7]. Regression models have been used to identify cycles and trends [7].

Recently, artificial neural networks (ANNs) have been applied to solve problems of predicting future stock indices

Manuscript received March 1, 2006. This work was supported in part by the National Research Foundation.

Prof. T. Marwala, is with the School of Electrical and Information Engineering, University of Witwatersrand, Private Bag 3, 2050, Johannesburg, South Africa. (phone: +27117177217; e-mail: t.marwala@ee.wits.ac.za).

P. B. Patel is a postgraduate student at the School of Electrical and Information Engineering, University of Witwatersrand, Private Bag 3, 2050, Johannesburg, South Africa. (e-mail: p.patel@ee.wits.ac.za).

[9][10][11]. Advanced methods such as genetic algorithms [12], Markov models [13] and fuzzy methods [12] have also been frequently used. ANNs together with pattern recognition techniques for stock market forecasting have also been employed [14]. Research has also been conducted in the prediction of stocks using case-based reasoning [8]. Random subspace classifier networks have also been used to predict the next day stock price return [15].

In this research, the Multi-Layer Perceptron (MLP) and the Radial Basis Function (RBF) neural network architectures have been considered. Detailed explanations on these ANN architectures can be found in [16].

The section to follow examines the forecasting system as well as its implementation methodology. The paper ends with the comparison of the various forecasting classifiers developed and the selection of the superior network. Recommendations for possible improvement in forecasting accuracies are also stated.

## II. THE DEVELOPED SYSTEM.

As the developed system is to be used in assisting an investor in making financial decisions, the application should be based on a profitable trading strategy. There are numerous trading strategies available [17]. This research focuses on the "Buy low, sell high" trading strategy. The strategy has been implemented as well as compared to the "Buy and hold" trading strategy in terms of profits generated.

The "Buy low, sell high" trading strategy entails purchasing certain stocks at a low price and selling these stocks when the price is high. The "Buy and hold" trading strategy, as the name suggests, involves an investor purchasing certain stocks and retaining them for a particular duration.

The strategies were implemented for the period 5 January 2004 to 16 September 2004 using the available data of the indices concerned. It has been assumed that the investment instrument price is equal to the index closing value divided by hundred. This provided a realistic index fund price. It has also been assumed that the investor had a limited trading capital of 100 000 available. This also assisted in simulating a real world scenario. Furthermore, the initial limited trading capital is divided into 4 equal portions, which were used to purchase units in the indices concerned. Thereafter, the trading strategies began.

The method used to implement the "Buy low, sell high" trading strategy involved classifying the change in index or delta into certain categorizes. Delta is defined as the difference between the closing index value for the next day and the closing index value for the previous day. Thereafter, depending on the classification, the strategy would recommend the investor to purchase stocks, hold the current investment position or sell stocks. Table I illustrates these classes as well as the corresponding strategy recommendation.

The percentage thresholds below were determined through experimentation. It has been determined that the percentage threshold combination of 0.8% and -0.20% of the closing value for the previous day is most profitable for the period concerned.

These percentage threshold values were further verified by extending the trading strategy implementation to include 2005 data. The strategies were implemented for the extended period of 5 January 2004 to 31 May 2005.

As a result, the system has been based on the "Buy low, sell high" trading strategy. It employs the percentage threshold combination of 0.8% and -0.2% of the closing index value for the previous day.

The forecasting of the indices can be classified as either multivariate or univariate models. A univariate model utilizes the past values of the time series to generate a prediction [18]. The disadvantage of this approach is that it does not consider the environmental effects and the interactions among different factors other than the outputs. A multivariate model employs additional information such as market indicators, technique indicators or fundamental factors of companies as inputs [18]. The disadvantage of a multivariate model is the difficulty involved in the selection of inputs. There are many factors leading to price fluctuations that cannot be captured precisely or may be too numerous and difficult to be modeled.

Several research outcomes, using the univariate approach, have illustrated acceptable results [18][19] . Networks with root mean square (RMS) errors between 0.0251 and 0.3318 were reported. As a result, this application utilizes the closing prices of the indices, represented by a univariate time series, as inputs.

## III. IMPLEMENTATION METHODOLOGY.

The data used to develop the forecasting application can be obtained from the internet [1]. The development process was divided into various stages. The following procedure has been pursued in the creation of the various ANN architectures employed:

1. Selection and processing of data to be used by ANNs during training, validation and testing.
2. Optimization of the number of hidden neurons or nodes within the ANN.
3. Optimization of the input time window using polynomial approximation.
4. Comparison of the various networks developed and the selection of the superior network.

The remainder of this section will elaborate on the various stages of the procedure stated above.

### *A. Selection and processing of data.*

The data utilized in developing the ANNs included closing price indices values from 5 January 2004 to 31 May 2005.

Before the data sets were analyzed, basic preprocessing of the information was considered. In the case of days with no trading, the missing data maybe required to be manually inserted. It has been stated that there are 3 techniques which could be employed to contend with days with no trading [20].

These are:

1. Ignore the days with no trading and use the data for trading days only.
2. Assign a 0 value for the days with no trading.
3. Build a model that can approximate the value for the days with no trading.

TABLE I
THE "BUY LOW, SELL HIGH" TRADING STRATEGY CATEGORIZES

| Class | Requirement | Strategy recommendation |
|---|---|---|
| Large Rise (LR). | Delta > Positive threshold percentage of previous day closing price. | If LR is forecasted for the next day, sell stocks in possession at the next day closing price. |
| Slight Rise (SR). | 0 < Delta <= Positive threshold percentage of previous day closing price. | If SR is forecasted for the next day, hold current investment position. |
| Slight Drop (SD). | Negative threshold percentage of previous day closing price <= Delta <= 0. | If SD is forecasted for the next day, buy stocks to the value of 15 % of available trading capital at the next day closing price. |
| Large Drop (LD). | Delta < Negative threshold percentage of previous day closing price. | If LD is forecasted for the next day, buy stocks to the value of 25 % of available trading capital at the next day closing price. |

A model was not created to determine the value for the days with no trading because it was feared that the values calculated may contribute significantly to the final error of the networks. Initial experiments were conducted, utilizing techniques 1 and 2 above, and it was found that, technique 1 resulted in lower error values. As a result, technique 1, from the above list, has been employed.

It should also be noted that the data sets utilized during the development of the ANNs consisted of closing values for the indices for the days when trading occurred on all the 4 stock exchanges considered. The closing values for the indices for the days when trading did not occur on all 4 stock exchanges of concern, but occurred on some stock markets considered, were omitted from the above data sets. This assisted in creating a realistic trading scenario.

In order to ensure that over-fitting and under-fitting were avoided, the data is divided into 3 sets. Over-fitting occurs when the network does not generalize but rather tends to memorize the training data. Under-fitting occurs when the network does not follow the data at all [16]. The data is divided into training, validation and test sets. The training data set is used to train the ANN to find the general pattern between its inputs and outputs. The validation data set is used to assess the network and the test data is used to confirm the prediction quality of the developed networks.

The data is segregated in time order. In other words, the data of the earlier period could be used for training the network, the data of the later period be used for validation and the data of the latest period used for testing. This approach may have a recency problem that is the ANNs are only trained using data from early 2004. When forecasting the performance of the indices during 2005, the ANNs are "forced" to utilize the knowledge learnt in early 2004.

Due to the above problem, the data is uniformly randomized and, thereafter, separated into the 3 required sets. As a result, the ANN are trained using randomly chosen data. A network with very good test data set results may not predict well for future forecasting. However, a network that has been trained with randomly chosen data may predict well for future forecasting even with average test data set results [21].

The output and input data sets of the indices to be forecasted were preconditioned by normalizing the data. Normalizing the data entails manipulating the data sets such that the values within the sets are between 0 and 1. The networks developed were trained utilizing the normalized data sets.

Normalization is accomplished by acquiring the minimum and maximum values within the data sets. The data is normalized by using the following formula:

$$X_{norm} = \frac{X - X_{\min}}{X_{\max} - X_{\min}}, \qquad (1)$$

where

$X_{norm}$ is the normalized value,

$X$ is the actual data,

$X_{\min}$ is the smallest value within the data set,

$X_{\max}$ is the largest value within the data set.

The purpose of normalizing the data sets is to modify the variable levels to a reasonable value. If such a transformation is not employed, the value of the variable could be too large for the network to process, especially when several layers of nodes within the ANN are involved [22]. Normalizing the data sets also reduces the fluctuation and noise within the data [16].

There are a variety of practical reasons that illustrate normalizing the data sets can result in faster training and reduce the chances of obtaining local optima. Some of these reasons include better numerical conditioning (Hessian matrices), better weight initialization values and better weight decay estimates [16].

*B. Optimization of the number of hidden nodes within the neural network.*

MLP and the RBF neural network architectures were utilized in forecasting the next day closing price performance for each of the indices considered. The MLP and RBF neural network architectures are possibly the most extensively employed ANNs in pattern classification [23]. Due to the non-linear capabilities of these networks, they are said to be excellent universal approximators that provide highly accurate solutions. As a result, these networks produce very practical tools for classification and inversion problems [16].

It has been stated that a network with 1 hidden layer, provided with sufficient data, can be used to model any function [24]. As a result, the MLP and RBF neural network architectures employed consisted of only 1 hidden layer.

During this stage of development, the number of inputs has been assumed to be arbitrary. This will be optimized at a later stage of implementation. This stage of development involved optimizing the ANN architecture. As a result, designing the ANN thus entailed selecting the correct number of hidden neurons and the appropriate network architectures that would yield the most accurate results.

The inputs to the networks were kept constant. There were 7 inputs to the networks. These were the previous 7 day closing index prices. The developed ANN had 4 outputs. Each output represented a performance class considered. The largest value, from the 4 neural classifier network outputs, indicates that the network forecasts that the next day closing index price will behave according to the performance class corresponding to that particular output.

The MLP network hidden layer consists of non-linear activation functions. The choice of the activation function is mainly dependant on the application of the network [16]. However, it has been found that the hyperbolic tangent activation function offers a practical advantage of giving rise to faster convergence during training [23]. As a result, this function has been utilized within the MLP networks.

The MLP network output layer also consists of activation functions. There are 3 major forms of the function that should be considered. These are the linear, logistic sigmoidal and softmax activation functions [23]. It has been stated that the appropriate selection of the output-unit activation function for a classification problem is the logistic sigmoidal function [23]. As a result, this function has been employed within the output layer of the MLP network.

The RBF network that has been developed contained a Gaussian activation function within its hidden layer and a linear activation function within its output layer.

The number of hidden neurons or nodes has been optimized by minimizing an error function that mapped the number of hidden nodes to the accuracy of the developed networks. The process has been performed on the validation data set.

Since this is a classification implementation, the accuracy of the networks developed can no longer be calculated utilizing the sum of square error of the difference between the target and the forecasted network output values. Instead, a confusion matrix is employed to identify the number of true and false classifications that are generated by the ANN developed. This is then utilized to calculate the true accuracy of the ANN classifiers, using the following equation:

$$Accuracy = \sqrt{\frac{TP \times TN}{(TP + FN) \times (FP + TN)}}, \qquad (2)$$

where

$TP$ is the true positive (1 classified as a 1),

$TN$ is the true negative (0 classified as a 0),

$FN$ is the false negative (1 classified as a 0),

$FP$ is the false positive (0 classified as a 1).

The hidden neurons or intermediate units were optimized by creating various MLP and RBF ANNs with hidden nodes of 5 to 150. As a result, 292 ANNs were developed. Networks with hidden nodes greater than 150 were not developed due to the predictive capabilities or generalization capabilities reducing as the number of intermediate units increase. More hidden nodes increases the dimensionality of the function being fitted, enabling easier training which results from higher training capacity. However, this detrimentally affects the generalization capabilities of the network. A major consideration when developing a suitable ANN for a financial application is to make a trade-off between convergence and generalization [25].

Utilizing the training data set, these networks were trained. The validation data set was then presented to the networks. Thereafter, the accuracy of the developed networks was calculated for the training and validation data sets. MLP and RBF networks with the number of hidden nodes that resulted in the largest accuracy value, when presented with the validation data set, were analyzed.

Table II illustrates the MLP and RBF networks that resulted in the largest accuracy value for the validation data set. Networks consisting of these numbers of hidden nodes were developed in the next stage of implementation.

### *C. Optimization of the input time window using polynomial approximation.*

This stage of implementation involved the optimization of the number of inputs that would concede the largest forecasting classification accuracy. As a result, this step of development entailed selecting the correct number of closing prices of the indices for the previous days as inputs to the networks.

The number of inputs has been optimized by minimizing an error function that mapped the number of inputs to the accuracy of the developed networks. The process was performed on the validation and test data sets.

The input time window was optimized by constructing various MLP and RBF networks with the number of closing prices for the previous days, required to predict the desired output, ranging from 5 to 19. These developed networks contained the number of hidden nodes as illustrated in Table II. The networks also employed the same activation functions mentioned in the previous section.

TABLE II
RESULTS OF VARIED HIDDEN NODES.

| Network architecture. | Number of hidden nodes. |
|---|---|
| Dow Jones Industrial Average. | |
| MLP | 18, 55, 58, 64, 78. |
| RBF | 29, 55, 73, 106, 116. |
| JSE All Share. | |
| MLP | 26, 35, 58, 96, 107. |
| RBF | 5, 12, 16, 18, 22. |
| NASDAQ 100. | |
| MLP | 51, 100, 119, 135, 148. |
| RBF | 42, 64, 74, 99, 131. |
| Nikkei 225 Stock Average. | |
| MLP | 60, 84, 103, 118, 128. |
| RBF | 19, 64, 95, 105, 138. |

Utilizing the training data set, these networks are trained. The validation and test data sets are then presented to the ANN. Thereafter, the accuracies for the training, validation and test data sets are calculated. When presented with the validation and test data sets, ANNs that resulted in the largest accuracy were analyzed. Table III below illustrates the MLP and RBF networks that resulted in the best accuracy values for the validation and test data.

Moving averages of 5, 6, 7 and 8 days of the closing index prices were also considered as inputs to the networks. The longer the time span of the moving average, the less sensitive it will be to daily price changes [26]. This is the reason for utilizing these moving averages as the ANNs are predicting the next day closing price performance for each index. Moving averages are utilized to emphasize the direction of a trend and reduce price as well as volume fluctuations that may confuse interpretations [26]. As a result, the moving average is employed to reduce the noise within the data.

Moving averages were introduced as inputs to the networks mentioned in Table III. The number of closing prices of the indices for the previous days is kept constant at the values illustrated in Table III. However, the moving average employed varied from a 5 day moving average to an 8 day moving average. It has been determined, from this investigation, that the moving averages introduced did increase the accuracy of the networks. This is valid for the Dow Jones Industrial Average, Nasdaq 100 and the Nikkei 225 Stock Average indices. However, the accuracy of the networks employed to forecast the closing price performance of the JSE All Share index did not improve with the addition of the moving averages as inputs.

### D. *Comparison of the various networks developed and the selection of the superior network.*

This stage of development entailed the comparison of the various ANNs that were created. It also involves the selection of the best networks to forecast the various indices considered.

The 5 most accurate networks from the input time window optimization and the moving averages investigation were used in the committees for the indices concern. The outputs of these networks were fed into a voting system. The voting system determined the final output of the committee. If the majority of the ANNs within the committee classified an output into a certain class, the voting system would classify the output of the committee as the class. If 2 of the networks within the committee classified an output into the same class and another 2 of the networks classified their outputs into a different class, the voting system would classify the output of the committee as undecided.

It is evident, from the investigations conducted, that the networks illustrated in Table IV resulted in the most accurate forecasting classifiers.

It has also been determined that the committee of networks does not always result in a more accurate solution. This is true for the forecasting of the closing price performance of the Nasdaq 100 index. Another conclusion that could be drawn is that the MLP network architecture is particularly suited for this application. The difference in accuracy between the 2 network architectures is approximately 10% for the forecasting of the Dow Jones Industrial Average and JSE All Share indices. However, the difference in accuracy between these network topologies is approximately 5% for the forecasting of the Nasdaq 100 and the Nikkei 225 Stock Average indices.

TABLE III
RESULTS OF VARIED INPUT DAYS.

| Network architecture. | Input days. | Hidden nodes. | Accuracy (Validation). | Accuracy (Test). |
|---|---|---|---|---|
| Dow Jones Industrial Average. | | | | |
| MLP | 16 | 58 | 0.6455 | 0.61538 |
| MLP | 15 | 64 | 0.63052 | 0.60007 |
| RBF | 18 | 73 | 0.61538 | 0.58456 |
| RBF | 17 | 106 | 0.60007 | 0.56885 |
| JSE All Share. | | | | |
| MLP | 13 | 26 | 0.66032 | 0.6455 |
| MLP | 18 | 26 | 0.675 | 0.675 |
| MLP | 18 | 35 | 0.6455 | 0.675 |
| MLP | 18 | 96 | 0.6455 | 0.6455 |
| MLP | 19 | 96 | 0.675 | 0.66032 |
| RBF | 8 | 12 | 0.56885 | 0.58456 |
| RBF | 9 | 18 | 0.56885 | 0.61538 |
| NASDAQ 100. | | | | |
| MLP | 19 | 135 | 0.66032 | 0.66032 |
| MLP | 19 | 148 | 0.66032 | 0.68954 |
| RBF | 19 | 64 | 0.63052 | 0.63052 |
| RBF | 15 | 99 | 0.6455 | 0.63052 |
| Nikkei 225 Stock Average. | | | | |
| MLP | 18 | 103 | 0.63052 | 0.66032 |
| MLP | 19 | 118 | 0.61538 | 0.61538 |
| MLP | 19 | 128 | 0.60007 | 0.60007 |
| RBF | 17 | 95 | 0.55292 | 0.61538 |
| RBF | 18 | 138 | 0.58456 | 0.61538 |

TABLE IV
NETWORKS SELECTED.

| Network architecture. | Networks within committee. Input days. | Day moving average. | Hidden nodes. | Accuracy (Validation). | Accuracy (Test). |
|---|---|---|---|---|---|
| Dow Jones Industrial Average. | | | | | |
| MLP | 15 | 5 | 64 | 0.7209 | 0.7493 |
| | 15 | 7 | 64 | | |
| | 15 | 8 | 64 | | |
| | 16 | 5 | 58 | | |
| | 16 | 6 | 58 | | |
| JSE All Share. | | | | | |
| MLP | 13 | - | 26 | 0.704 | 0.6775 |
| | 18 | - | 26 | | |
| | 18 | - | 35 | | |
| | 18 | - | 96 | | |
| | 19 | - | 96 | | |
| Nikkei 225 Stock Average. | | | | | |
| MLP | 18 | 6 | 103 | 0.6529 | 0.6402 |
| | 18 | 7 | 103 | | |
| | 19 | 7 | 118 | | |
| | 19 | 5 | 128 | | |
| | 19 | 6 | 128 | | |
| NASDAQ 100. | | | | | |
| Network architecture. | Input days. | Day moving average. | Hidden nodes. | Accuracy (Validation). | Accuracy (Test). |
| MLP | 19 | 5 | 148 | 0.68954 | 0.70396 |

## IV. Conclusion

This research entailed the development of a system that could estimate the next day closing index price performance. The Dow Jones Industrial Average, the JSE All Share, the Nasdaq 100 and the Nikkei 225 Stock Average indices were considered.

The development methodology utilized involved, initially, varying the number of hidden nodes within the ANNs. This resulted in creating an acceptable network architecture. Thereafter, the numbers of closing prices of the indices for the previous days as inputs to the networks were varied. Moving averages were also introduced as inputs to the networks to reduce the noise within the data. Acceptable forecasting classification accuracies were achieved. The best and worst accuracy levels obtained were 72% and 64%, respectively. These accuracy levels were attained for the Dow Jones Industrial Average and the Nikkei 225 Stock Average indices, respectively. As a result, it can be concluded that the univariate approach of the forecasting of indices is relevant and can result in highly accurate solutions.

The accuracy of these performance classifications could be improved by using complex committee of classifiers. It could also be improved by employing a genetic algorithm to create the optimal ANN architecture. The genetic algorithm could also be used to optimize the appropriate number of closing prices of the indices for the previous days as inputs to the networks.

## References


[1] Yahoo Finance, last accessed: 20 February 2006. [Online]. Available: http://finance.yahoo.com.

[2] E. E. Peters, *Chaos and order in the capital markets: A new view of cycles, prices and market volatility, second edition.*, John Wiley & Sons Inc, 1996.

[3] S. Taylor, *Modeling financial time series.*, John Wiley & Sons, 1986.

[4] J. Jarrett, E. Kyper, "Evidence on the seasonality of stock market prices of firms traded on organized markets.", Taylor & Francis, *Applied Economics Letters.*, vol.12, number 9, pp. 537-543, 2005.

[5] J. S. Ang, R. A. Pohlman, *A note on the price behavior of Far eastern stock.*, J. Int. Business Studies, 1978.

[6] B. H. Solnik, *Note on the validity of the random walk for European stock prices*, J. Finance, 1973.

[7] J.P.C Kleijnen. *Statistical Techniques in Simulation*, New York: Marcel Dekker, 1974.

[8] SH. Chun, YJ. Park, "Dynamic adaptive ensemble case-based reasoning: application to stock market prediction.", *European Systems with Applications.*, vol. 28, issue 3, pp. 435 - 443, 2005.

[9] M. E. Azoff, *Neural network time series forecasting of financial markets.*, John Wiley & Sons Ltd, 1994.

[10] E. Gately, *Neural networks for Financial Forecasting.*, John Wiley & Sons Ltd, 1996.

[11] A. N. Refenes, Y. Bentz, "Neural networks in financial engineering: A study in methodology.", *IEEE Transaction on Neural Networks.*, vol. 8, issue 6, pp. 1222 - 1267, 1997.

[12] D. N. Chorafas, *Chaos theory in the financial markets: Applying fractals, fuzzy logic, genetic algorithms, Swarn simulation & the Monte Carlo method to manage markets.*, Probus Publishing Company, 1994.

[13] I. L. MacDonald, W. Zucchini, *Hidden Markov and other models for discrete-valued time-series.*, Chapman & Hall, 1997.

[14] W. Leigh, M. Paz, R. Purvis, "An analysis of a hybrid neural network and pattern recognition technique for predicting short-term increases in the NYSE composite index.", *Omega.*, vol. 30, issue 2, pp. 69-76, 2002.

[15] D. V. Zhora, "Data preprocessing for stock market forecasting using random subspace classifier network.", *Proceedings of International Joint Conference on Neural Networks.*, pp. 2549-2554, Montreal, Canada, 2005.

[16] C. M. Bishop, *Neural Networks for Pattern Recognition.*, Oxford University Press, 1995.

[17] Investments guide, last accessed: 20 February 2006. [Online]. Available: http://www.investments-guide.com.

[18] P. K. H. Phua, X. Zhu, C. H. Koh, "Forecasting stock index increments using neural networks with trust region methods.", *Proceedings of International Joint Conference on Neural Networks*, vol. 1, pp. 260-265, 2003.

[19] B. Leke and T. Marwala. "Optimization of the stock market input time-window using Bayesian neural networks." *Proceedings of the IEEE International Conference on Service Operations, Logistics and Informatics*, pp. 883-894, Beijing, China, 2005.

[20] R. Heinkel, A. Kraus, "Measuring event impacts in thinly traded stocks.", *The Journal of Financial and Quantitative Analysis.*, vol. 23, no. 1, pp. 71-88, 1988.

[21] J. Yao, C. L. Tan, HL. 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.

[22] J. Yao, C. L. Tan, "Guidelines for financial forecasting with neural networks.", *Proceedings of International Conference on Neural Information Processing.*, Shanghai, China, pp. 757-761, 2001.

[23] I. T. Nabney, *Netlab: Algorithms for Pattern Recognition.*, Springer, 2002.

[24] R. Beale, T. Jackson., *Neural Computing-An introduction.*, Taylor & Francis, 1990.

[25] E. B. Baum, D. Haussler., "What size net gives valid generalization?", *Neural Computation*, vol. 1, pp. 81-90, 1989.

[26] J. Cory, "Technical analysis tutorial", *Investopedia.com*, last accessed 20 February 2006. [Online]. Available: http://www.investopedia.com/university/technical/

# Appendix B: Paper submitted to 2006 International Conference on Neural Information Processing.

## Neural networks, Fuzzy Inference Systems and Adaptive-Neuro Fuzzy Inference systems for financial decision making.

Pretesh B. Patel and Tshilidzi Marwala

Pretesh Patel is a postgraduate student at the School of Electrical and Information Engineering, University of Witwatersrand, Private Bag 3, 2050, Johannesburg, South Africa.
p.patel@ee.wits.ac.za
Prof. Tshilidzi. Marwala, is with the School of Electrical and Information Engineering, University of Witwatersrand, Private Bag 3, 2050, Johannesburg, South Africa.
t.marwala@ee.wits.ac.za
This work was supported in part by the National Research Foundation.

**Abstract.** This paper is employing pattern classification methods for assisting investors in making financial decisions. Specifically, the problem entails the categorization of investment recommendations. Based on the forecasted performance of certain indices, the Stock Quantity Selection Component is to recommend the investor to purchase stocks, hold the current investment position or sell stocks in possession. Three designs of the component were implemented and compared in terms of their complexity as well as scalability. Designs that utilized 1, 4 and 16 classifiers, respectively, were developed. These designs were implemented using Artificial Neural Networks, Fuzzy Inference Systems as well as Adaptive Neuro-Fuzzy Inference Systems. The design that employed 4 classifiers achieved low complexity and high scalability. As a result, this design is most appropriate for the application of concern.

## 1. Introduction

Pattern recognition could be defined as the study of the ability of machines to observe the environment, learn to differentiate between patterns of interest from their backgrounds and formulate reliable as well as sensible decisions about the categories of the patterns [1]. This is a complex task that is an innate ability for humans. However, to develop a system to solve such problems poses formidable research challenges.

This research focuses on a pattern classification problem utilized within an application that could assist individual as well as institutional investors in making financial decisions. It is anticipated that this application would be used in conjunction with other financial analysis methodologies. As a result, such an application should be employed to confirm an investment decision.

Pattern classification is the process of assigning an input pattern to one of a predefined set of classes. It consists of developing a functional relationship between the input features and the target classes. Accurately estimating such a relationship is vital to the success of a classifier.

Specifically, the quantity of stocks or shares to be purchased based on the forecasted performance of certain indices is the pattern classification problem. The Dow Jones Industrial Average, Johannesburg Stock Exchange or the JSE Securities Exchange (JSE) All Share, Nasdaq 100 and Nikkei 225 Stock Average indices are considered. However, the computational intelligent techniques as well as their implementation methodology utilized in this research could be adapted for decision making systems in other industry sectors.

The classification of data into various classes has been an important research area for many years. Artificial neural networks (ANNs) have been applied to pattern classification [2]. Research has also been conducted on fuzzy classification. This resulted in many algorithms, such as fuzzy K-nearest neighbour [3] and fuzzy c-means [4], being applied to decision making systems. Fuzzy systems constructed using genetic algorithms have been utilized [5][6][7]. Fuzzy neural networks have also been employed in pattern classification applications [8][9][10]. Support Vector Machines have been applied to multi-category classification problems [11]. These classification tasks have also been implemented by combining multiple simpler specialized classifiers [12][13][14].

In this research, artificial neural network (ANN) architectures, Fuzzy Inference Systems (FISs) as well as Adaptive Neuro-Fuzzy Inference Systems (ANFISs) have been considered. Specifically, the Multi-Layer Perceptron (MLP) and the Radial Basis Function (RBF) neural network architectures have been considered. FISs developed employed subtractive clustering to generate the required membership functions and set of fuzzy inference rules. Information on these computational intelligent techniques can be found in [15], [16] and [17], respectively.

The next section briefly examines the application of concern. Thereafter, the implementation methodology is described. The paper concludes with the comparison of the various models developed and the selection of the superior classifiers.

## 2. The developed system.

The developed system is to be used in assisting an investor in making financial decisions. As a result, the system should be based on a profitable trading strategy. There are numerous trading strategies available [18]. This research focuses on the “Buy low, sell high” trading strategy. The strategy has been implemented as well as compared to the “Buy and hold” trading strategy in terms of profits generated.

The “Buy low, sell high” trading strategy entails purchasing certain stocks at a low price and selling these stocks when the price is high. The “Buy and hold” trading strategy, as the name suggests, involves an investor purchasing certain stocks and retaining them for a particular duration.

The method used to implement the “Buy low, sell high” trading strategy involved classifying the change in index or delta into certain categorizes. Delta is defined as the difference between the closing index value for the next day and the closing index value for the previous day. This functionality has been implemented within the

Forecasting Component (FC). Depending on the classification of this component, the strategy would recommend the investor to purchase stocks, hold the current investment position or sell stocks. This responsibility can be found in the Stock Quantity Selection Component (SQSC). Table 1 illustrates the forecasted classes as well as the corresponding investment recommendation.

It has been determined that the "Buy low, sell high" trading strategy, with the percentage threshold combination of 0.8% and -0.20% of the closing value for the previous day, is most profitable. As a result, the system has been based on this trading strategy. Further information on the comparison of the 2 trading strategies considered can be found in [19].

Pattern classification problems can be grouped as either dichotomous or polychotomous problems. Dichotomous classification can be interpreted as 2-class classification problems, whereas polychotomous classification involves problems with more than 2 classes to be categorized.

The SQSC module is the center of this research. Based on the forecasted performance of the closing price of the index, the component is to recommend the investor to purchase stocks, hold the current investment position or sell stocks. It is evident that this is a polychotomous classification problem as there are more than 2 classes. Further information on the FC module can be found in [19].

Various classifier designs of the SQSC module were considered. Each of these designs were developed using both ANNs as well as fuzzy logic techniques. The first design employed 1 classifier. This classifier consisted of 16 inputs and 16 outputs. The inputs to the model are the forecasted performance of the closing price of the indices considered. The outputs of the classifier are the investment recommendations for the indices.

The second design involved 4 classifiers. Each classifier has 4 inputs and 4 outputs. Each classifier is used to generate an investment recommendation for an index considered. The input to a classifier is the forecasted performance of the closing price of an index. The output of a classifier is the investment recommendation for the index.

The third and final design considered utilized 16 classifiers. Each classifier has 4 inputs and 1 output. Each classifier is employed to categorize whether or not to execute an investment recommendation. The input to a classifier is the same as design 2 above. The outputs of the classifiers are fed into an interpretation function that generates the final investment recommendations for the indices. This design has been implemented to investigate the method of utilizing simpler classifiers to generate a multi-category classifier.

**Table 1.** The "Buy low, sell high" trading strategy categorizes.

| Class | Requirement | Investment recommendation |
|---|---|---|
| Large Rise (LR) | Delta > Positive threshold percentage of previous day closing price. | If LR is forecasted for the next day, sell stocks in possession. |
| Slight Rise (SR) | 0 < Delta <= Positive threshold percentage of previous day closing price. | If SR is forecasted for the next day, hold current investment position. |
| Slight Drop (SD) | Negative threshold percentage of previous day closing price <= Delta <= 0. | If SD is forecasted for the next day, buy stocks to the value of 15 % of available trading capital. |
| Large Drop (LD) | Delta < Negative threshold percentage of previous day closing price. | If LD is forecasted for the next day, buy stocks to the value of 25 % of available trading capital. |

## 3. Implementation Methodology.

The data used to develop the SQSC module has been generated based on the 4 forecasted closing price performance classes illustrated in Table 1. The development process was divided into various stages. The following procedure has been pursued in the creation of the various classifiers employed:

1. Selection and processing of data to be used by the classifiers during training, validation and testing.
2. Optimization of the classification threshold of the various classes to be categorized.
3. Optimization of the classifier architectures.
4. Comparison of the various classifiers developed and the selection of the superior model.

The remainder of this section will elaborate on the various stages of implementation mentioned above.

### 3.1. Selection and processing of data.

The data utilized in developing and testing the various classifiers has been created by analyzing all the possible combinations of the 4 forecasted closing price performance classes. As a result, the entire data set consisted of 256 unique data records.

In order to present the forecasted closing price performance classes to the classifiers, a binary notation is employed. These inputs are presented to the classifier using 4 inputs. This input representation format is used for all indices considered. A similar binary notation scheme is also utilized to present the investment recommendation outputs. Table 2 illustrates the manner in which the inputs and outputs of the component are to be interpreted.

As previously mentioned, design 1 has 16 inputs and 16 outputs. The input representation format is the same as above. However, the first group of 4 inputs corresponds to the forecasted performance of the Dow Jones Industrial Average index. Similarly, the second, third and fourth group of 4 inputs characterizes the forecasted performance of the JSE All Share, Nasdaq 100 and Nikkei 225 Stock Average indices, respectively. The outputs are to be interpreted in a similar manner.

**Table 2.** Classifier input and output representation

| Classifier inputs | | | | | Classifier outputs | | | | |
|---|---|---|---|---|---|---|---|---|---|
| Input | 1 | 2 | 3 | 4 | Output | 1 | 2 | 3 | 4 |
| LR | 1 | 0 | 0 | 0 | Sell stocks in possession | 0 | 1 | 0 | 0 |
| SR | 0 | 1 | 0 | 0 | Hold current position | 1 | 0 | 0 | 0 |
| SD | 0 | 0 | 1 | 0 | Buy stocks to the value of 15 % of available trading capital | 0 | 0 | 1 | 0 |
| LD | 0 | 0 | 0 | 1 | Buy stocks to the value of 25 % of available trading capital | 0 | 0 | 0 | 1 |

The data is divided into a training, validation and test set. During the implementation of all 3 designs considered, the training data set consisted of all data records where the inputs were classified into 2 of the 4 closing price performance classes. However, the models developed were validated and tested with the remaining possible closing price performance class combinations.

Dividing the data set into 3 portions assists in ensuring that over-fitting as well as under-fitting has been avoided during the development of the ANNs. Over-fitting occurs when the network does not generalize but rather tends to memorize the training data. Under-fitting occurs when the network does not follow the data at all [15]. The training data set is used to train the ANN to find the general pattern between its inputs and outputs. The validation data set is used to assess the network and the test data is employed to confirm the classification quality of the developed model.

The training data set is used to create the cluster centers within the FISs. However, the validation and test data sets are utilized to assess the classification ability of the inference systems.

Due to the binary representation of the inputs and outputs, normalization of the data is not required. Normalizing the data entails manipulating the data sets such that the values within the sets are between 0 and 1.This results in faster training and it also reduces the chances of obtaining local optima [15].

### 3.2. Optimization of the classification threshold.

MLP and RBF neural network architectures were utilized in the classification of investment recommendations. The MLP and RBF neural network architectures are possibly the most extensively employed ANNs in pattern classification [2]. Due to the non-linear capabilities of these networks, they are said to be excellent universal approximators that provide highly accurate solutions. As a result, these networks produce very practical tools for classification and inversion problems [15].

It has been stated that a network with 1 hidden layer, provided with sufficient data, can be used to model any function [20]. As a result, the ANN architectures employed consisted of only 1 hidden layer.

The MLP network hidden layer consists of non-linear activation functions. The choice of the activation function is largely dependent on the application of the model [15]. However, it has been found that the hyperbolic tangent activation function offers a practical advantage of faster convergence during training [2]. As a result, this function has been employed within the MLP network.

The MLP network output layer also contains activation functions. There are 3 major forms of the function that should be considered. These are the linear, logistic sigmoidal and softmax activation functions [2]. It has been stated that the appropriate selection of the output layer activation function for a classification problem is the logistic sigmoidal function [2]. As a result, this function has been employed within the output layer of the MLP network. The RBF networks that have been developed contained a Gaussian activation function within its hidden layer and a linear activation function within its output layer.

As previously mentioned, the FISs developed utilized subtractive clustering to create the required membership functions and set of fuzzy inference rules. During this stage of implementation, the number of hidden nodes within the ANNs and the cluster radius utilized by the cluster centers within the FISs were assumed to be arbitrary. This will be optimized at a later stage of development. During this stage of development, the number of hidden nodes within the ANNs as well as the cluster radius utilized by the FISs was 10 and 0.5, respectively. This stage of implementation involved the optimization of the interpretation of the classifiers. As a result, this involved the selection of an appropriate classification threshold value that would yield the most accurate results.

The classification threshold has been optimized by minimizing an error function that mapped the classification thresholds to the accuracy of the developed classifiers. The process has been performed on the validation data set.

Since this is a classification implementation, the accuracy of the models can no longer be calculated using the sum of square error of the difference between the target and investment recommendation classifier output. Instead a confusion matrix is utilized to identify the number of true and false classifications that are generated by the models developed. This is then used to calculate the true accuracy of the classifiers, using the following equation:

$$Accuracy = \sqrt{\frac{TP \times TN}{(TP + FN) \times (FP + TN)}} \qquad (\mathbf{1})$$

where

$TP$ is the true positive (1 classified as a 1),

$TN$ is the true negative (0 classified as a 0),

$FN$ is the false negative (1 classified as a 0),

$FP$ is the false positive (0 classified as a 1).

The classification threshold was optimized by initially creating classifiers utilizing a threshold value of 0.5. This implies that if the classifier outputs a value less than 0.5, the output will be regarded as a 0. Similarly, if the output value is larger than or equal to 0.5, the output will be interpreted as a 1. This threshold value of 0.5 proved to be adequate for the MLP networks as well as the FISs implementations. The threshold value resulted in 100% accurate classifications. This has been demonstrated on the training as well as validation data sets. However, the RBF classifier employed in design 1 did not perform well utilizing this threshold value. As a result, the classification threshold of this model had been varied from 0.1 to 0.5 in iterations of 0.01. Table 3 illustrates the threshold values that resulted in the largest accuracy value for the validation data set. The threshold value of 0.5 proved to be satisfactory for design 2 and design 3 RBF classifiers.

**Table 3.** Results of varied classification threshold for design 1 RBF classifier.

| | Classification thresholds. | | | |
|---|---|---|---|---|
| Class. | Dow Jones Industrial Average index. | JSE All Share index. | Nasdaq 100 index. | Nikkei 225 Stock Average index. |
| Sell stocks in possession | 0.19 | 0.23 | 0.20 | 0.11 |
| Hold current position | 0.17 | 0.10 | 0.12 | 0.17 |
| Buy stocks to the value of 15 % of available trading capital | 0.19 | 0.16 | 0.13 | 0.17 |
| Buy stocks to the value of 25 % of available trading capital | 0.24 | 0.17 | 0.17 | 0.19 |

### 3.3. Optimization of the classifier architectures.

This stage of implementation involved the optimization of the ANN and Fuzzy Inference System (FIS) architectures. As a result, this step of development involved the selection of the correct number of hidden neurons that would yield the most accurate results. It also entailed selecting the correct cluster radius that would concede the largest investment recommendation classification accuracy.

The number of hidden neurons or nodes has been optimized by minimizing an error function that mapped the number of hidden nodes to the accuracy of the developed network. The process was performed on the validation and test data sets.

The hidden nodes were optimized by creating various MLP and RBF ANNs with hidden nodes of 1 to 75. As a result, 150 ANNs were developed. These developed networks employed the classification thresholds stated in the previous section. The networks also utilized the same activation functions mentioned in the previous section.

Utilizing the training data set, these networks are trained. The validation and test data are then presented to the ANN. Thereafter, the accuracies for the training, validation and test data sets are calculated. When presented with the validation and test data, ANNs that resulted in the largest accuracy were analyzed.

Fig. 1 illustrates the sell stocks in possession at the next day closing price investment recommendation results of design 1. Similar results were achieved for the other design implementations as well as investment recommendations. Similar results were also obtained for the other indices considered.

The investigation revealed that a design 1 MLP and RBF network with number of hidden nodes larger than 12 and 52, respectively, yield 100% accurate models for categorizing the investment recommendations appropriately. The investigation also determined that design 2 MLP and RBF ANNs with number of hidden nodes greater than 2 and 5, respectively, achieved the same results. Similar results were obtained with design 3 MLP and RBF networks that contained more than 1 hidden neuron.

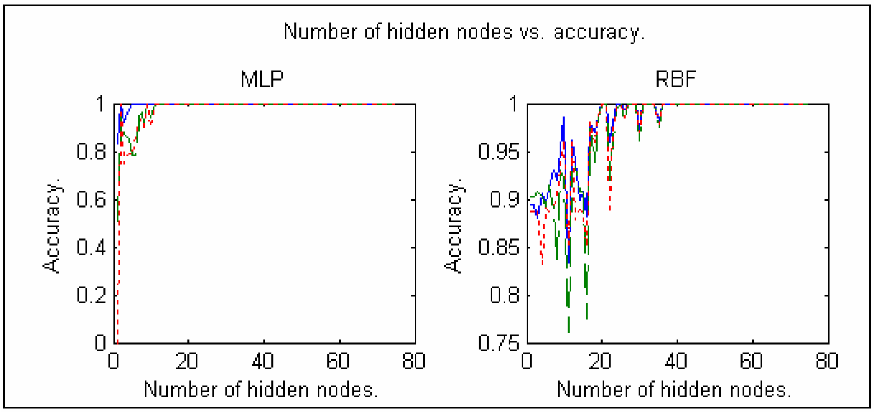


**Fig. 1.** This figure illustrates the results of sell stocks in possession at the next day closing price investment recommendation for design 1. The number of hidden nodes was varied and the corresponding accuracy values achieved were noted. The solid, dashed and dotted line represent the training, validation and test data sets, respectively

The cluster radius indicates the range of influence of a cluster. A small cluster radius results in small clusters in the data and, therefore, many fuzzy rules. Large cluster radii yield few large clusters in the data and, hence, fewer fuzzy rules [16].

The cluster radius has been optimized by minimizing an error function that mapped the radius to the accuracy of the developed inference systems. This process was performed on the validation and test data sets.

During this step of implementation, the optimization process entailed the construction of various inference systems with the cluster radius ranging from 0.01 to 1.

The investigation determined that design 2 FISs with a cluster radius equal to or greater than 0.01 achieve 100% accuracy in categorizing the investment recommendations appropriately. However, the design 1 FIS did not achieve 100% investment recommendation classification accuracies. It has been determined that a cluster radius of 0.11 achieved the most accurate results. The lowest accuracy value attained was 83%. The largest accuracy value was 100%.

### 3.4. Comparison of the various designs implemented and the selection of the superior model.

This stage of implementation entailed the comparison of the various designs that were developed. It also involves the selection of the best design to classifier the investment recommendations.

**Table 4.** This table illustrates the various models that were created. Accuarcies are presented as percentages

| Design | Classifier topology | Hidden nodes | Fuzzy rules | Membership functions | Accuracy (Training) | Accuracy (Validation) | Accuracy (Test) |
|---|---|---|---|---|---|---|---|
| 1 | MLP | 12 | - | - | 100 | 100 | 100 |
| 1 | RBF | 52 | - | - | 100 | 100 | 100 |
| 1 | FIS | - | 85 | 1360 | 100 | 83 | 87 |
| 2 | MLP | 2 | - | - | 100 | 100 | 100 |
| 2 | RBF | 5 | - | - | 100 | 100 | 100 |
| 2 | FIS | - | 4 | 16 | 100 | 100 | 100 |
| 3 | MLP | 1 | - | - | 100 | 100 | 100 |
| 3 | RBF | 1 | - | - | 100 | 100 | 100 |
| 3 | ANFIS | - | 4 | 16 | 100 | 100 | 100 |

Table 4 above, illustrates the various models that have been created. The above designs have been compared in terms of their complexity as well as scalability. Complexity, in this context, is defined as the number of classifiers employed by the design. Scalability is defined as the ability of the design to accommodate the classification of additional investment recommendations.

It is evident that design 1 has low complexity and low scalability. When additional investment recommendations are to be added to the component, the classifier employed is to be re-trained. However, design 2 has low complexity as there are only 4 classifiers utilized. The design also has high scalability. It is not required to re-create the existing classifiers, when additional recommendations are added. Table 4 indicates that design 3 has high complexity. The design contains 16 classifiers. In order to add investment recommendations to the component, the existing classifiers do not have to be re-created. As a result, the design has high scalability.

Due to the above analysis, design 2 is most appropriate for this application. It does not employ many classifiers and the design does not require re-work when additions are to be made.

It is evident from Table 4 that both the ANN and FIS implementations of design 2 perform satisfactorily. As a result, either of the classifier architectures could be used.

## 4. Conclusion.

This research involved the development of a component that could categorize investment recommendations, based on the forecasted performance of indices, appropriately. The Dow Jones Industrial Average, JSE All Share, Nasdaq 100 and Nikkei 225 Stock Average indices were considered.

Various designs of the component were considered. Designs that utilized 1, 4 and 16 classifiers were implemented. The development methodology employed in the creation of these designs, initially, involved the selection of appropriate classification thresholds. Thereafter, the number of hidden nodes within the ANNs as well as the cluster radius of the cluster centers within the FISs was varied. This resulted in creating acceptable classifier architectures. Acceptable investment recommendation classification accuracies were achieved.

The designs were compared in terms of complexity as well as scalability. Complexity is concerned with the number of classifiers that are used within the design. Scalability is the ability of the design to accommodate the classification of additional investment recommendations. Design 2 has low complexity and high scalability. This design consisted of 4 classifiers. Each classifier has 4 inputs and 4 outputs. This design is most appropriate for the application of concern.

## References


[1] A. K. Jain, R. P. W. Duin and J. Mao, "Statistical pattern recognition: A review.", *IEEE Transactions on Pattern Analysis and Machine Intelligence.*, vol. 22, no. 1, pp. 4-37, 2000.

[2] I. T. Nabney, *Netlab: Algorithms for Pattern Recognition.*, Springer, 2002.

[3] J. M. Keller, M. Gray and J. Givens, "A fuzzy k-nearest neighbor algorithm.", *IEEE Transaction on System, Man and Cybernetics.*, vol. 15, no. 4, pp. 580-585, 1985.

[4] J. C. Bezdek, *Pattern Recognition with Fuzzy Objective Function Algorithms.*, Plenum, 1981.

[5] H. Ishibuchi, K. Nozaki, N. Yamamoto and H. Tanaka, "Construction of fuzzy classification systems with rectangular fuzzy rules using genetic algorithms.", *Fuzzy Sets and Systems.*, vol. 65, no. 2-3, pp. 237-253, 1994.

[6] H. Ishibuchi, K. Nozaki, N. Yamamoto and H. Tanaka, "Selecting fuzzy if-then rules for classification problems using genetic algorithms.", *IEEE Transactions on Fuzzy Systems.*, vol. 3, no. 3, pp. 260-270, 1995.

[7] M. Russo, "FuGeNeSys – A fuzzy genetic neural system for fuzzy modeling.", *IEEE Transaction on Fuzzy Systems.*, vol. 6, no. 3, pp. 373-388, 1998.

[8] F. Masulli, F. Casalino and F. Vannucci, "Bayesian properties and performances of adaptive fuzzy systems in pattern recognition problems.", *Proceedings of the European Conference on Artificial Neural Networks., ICANN-94.*, pp. 189-192, 1994.

[9] D. Nauck and R. Kruse, "A neuro-fuzzy method to learn fuzzy classification rules from data.", *Fuzzy Sets and Systems.*, vol. 89, no. 3, pp. 277-288, 1997.

[10] C-T. Sun and J-S. R. Jang, "A neuro-fuzzy classifier and its applications.", *Proceedings of the IEEE International Conference on Fuzzy System.*, vol. 1, pp. 94-98, 1993.

[11] C-W. Hsu and C-J. Lin, "A comparison of methods for multi-class support vector machines", *IEEE Transactions on Neural Neworks.*, vol. 13, no. 2, pp. 415-425, 2002.

[12] J. H. Friedman, "Another approach to polychotomous classification.", *Department of Statistics, Stanford University.*, Technical report, 1996.

[13] J. Schurmann, *Pattern Classification. A Unified View of Statistical and Neural Principles.*, John Wiley & Sons Inc, 1996.

[14] B. Zadrozny, "Reducing multiclass to binary by coupling probability estimates.", *Neural Information Processing Systems Foundation 14*, pp. 1041-1048, 2002.

[15] C. M. Bishop, *Neural Networks for Pattern Recognition.*, Oxford University Press, 1995.

[16] S. Chiu, "Fuzzy Model Identification Based on Cluster Estimation.", *Journal of Intelligent and Fuzzy Systems.*, vol. 2, no. 3, pp. 267-278, 1994.

[17] J-S. R. Jang, "ANFIS: Adaptive-Network-Based Fuzzy Inference System.", *IEEE Transaction on System, Man and Cybernetics.*, vol. 23, no. 3, pp. 665-685, 1993.

[18] Investments guide, last accessed: 25 March 2006. [Online]. Available: http://www.investments-guide.com

[19] P. B. Patel and T. Marwala, "Forecasting closing price indices using neural networks.", *IEEE System, Man and Cybernetics conference.*, submitted, 2006.

[20] R. Beale, T. Jackson, *Neural Computing-An introduction.*, Taylor & Francis, 1990.

# Appendix C: Neural Networks.

Due to the difficulty and complexity of statistical techniques as well as the high level of proficiency required to utilize such methods, there has been a significant increase in the usage of Artificial Neural Networks (ANNs). This increase has also been attributed to the fact that ANNs can be applied to virtually every field in industry. For example, ANNs can be utilized in medical diagnosis, machine fault diagnosis, fingerprint recognition as well as financial creditworthiness evaluation applications. These networks can also be employed in product line development to control the quality of products manufactured. Artificial Neural Network (ANN) research has gathered enormous momentum in recent years. As a result, this field of study has been introduced in many universities.

ANNs were introduced based on the understanding of neurology. They have been motivated by the fact that scientists are challenged to effectively utilize machines on tasks currently solved by humans. Although there is no universally accepted definition of an ANN, it can be considered as an exceptionally robust data-modeling tool that consists of a network of interconnected simple processors or units, which individually operate on local data and together these units capture as well as numerically represent the intricate input output relationships of complex systems [1][2]. These networks are data-mining techniques that have been inspired by the desire to develop artificial systems capable of performing 'intelligent' computations similar to those performed within the human brain. An ANN acquires its knowledge through repeated presentations of data. It 'learns' by adjusting the weights of the network connections, which is similar to adjusting the synaptic weights within the inter-neuron connections within the human brain [1]. Thereafter the network will exhibit some capability for generalization in obtaining rather accurate outputs when presented with new unseen data. An advantage of ANNs is their ability to represent both linear as well as non-linear relationships. As a result, these networks are able to approximate any computable function to arbitrary precision [3].

There exists a great diversity of ANN architectures. However, the most common, and often used in practical applications, is that of a feed-forward structured neural network [3][4]. It has been stated that these ANN architectures with a single hidden layer, provided with sufficient data, can be used to model any function [5]. However, a situation where utilizing 2 or more hidden layers may prove necessary or worthwhile could exist. This is dependent on the primary purpose of the ANN within the application.

Among the family of feed-forward structured networks, the Multi-Layer Perceptron (MLP) and the Radial Basis Function (RBF) neural network architectures are possibly the most extensively employed ANNs in pattern classification [4]. MLP networks employ correlation-based algorithms, whereas RBF networks utilize distance-based algorithms [6]. Both networks function in a supervised manner. Due to the non-linear capabilities of these networks, they are said to be excellent universal approximators that provide highly accurate solutions [3]. As a result, these networks produce very practical tools for classification and inversion problems.

The MLP network evolved from the combination of many simple components. The most fundamental of these is the mathematical model of the neuron. In 1943 McCullock and Pitts proposed this neural model, which then formed the basis for formal calculus of brain activity [7]. In 1958 Rosenblatt introduced the Perceptron model. This was an elementary visual system that could be taught to recognize a limited class of patterns [7]. It was this model that then formed the foundation upon which most forms of artificial intelligence was born [8]. A perceptron can be considered as a device that computes the weighted sum of its inputs. It then propagates this sum through an activation function to produce the output. This activation function can be linear or nonlinear [7]. However, a network of linear perceptrons was found to have serious computational limitations [7]. These limitations were overcome by adding layers of nonlinear perceptrons that resulted in the MLP neural network.

The RBF networks have become a popular alternative to the MLP network approach [2]. RBF networks are inspired from traditional statistical classification techniques [2]. These are based on Cover's theorem on the separability of patterns. This theorem states that nonlinearly separable patterns can be separated linearly if the pattern is cast nonlinearly into a higher dimensional space. Therefore, the RBF network converts the pattern to a higher dimension after which it classifies the pattern linearly [2].

Fig. 1 illustrates a feed forward structured network. As mentioned above, the MLP and RBF networks are feed forward structured network architectures whereby each unit receives inputs only from lower layer units. Feed forward structured networks do not have connections between units in the same layer. These networks usually comprises of input, hidden and output layers, all of which are interconnected with respect to the hidden layer.

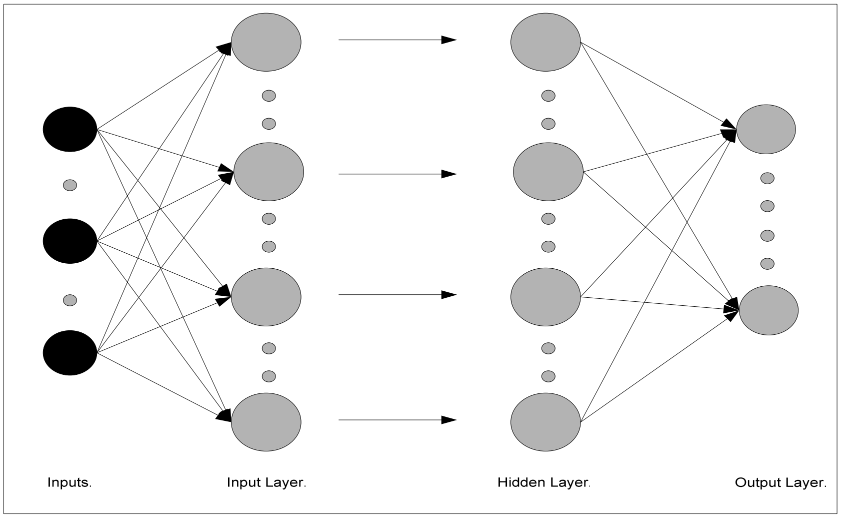


Fig. 1. Feed-forward neural network topology.

The hidden and output layers contain activation functions. The choice of the hidden-unit activation function for the MLP network is mainly dependent on the application of the network [3]. However, it has been determined that the hyperbolic tangent activation function offers a practical advantage of giving rise to faster convergence during training [4]. There are 3 major forms of the MLP network output-unit activation function. These are the linear, logistic sigmoidal and softmax activation functions [4]. It has been stated that the appropriate selection of the MLP network output-unit activation function for a pattern classification problem is the logistic sigmoidal function [4].

The RBF network hidden layer, utilizing a set of basis functions, performs a nonlinear mapping from the input space into a higher dimensional space in which the patterns become linearly separable. In order to accomplish this, the RBF network employs a Gaussian hidden-unit activation function. The output layer usually implements a linear weighted sum of the hidden layer outputs [2]. As a result, a linear activation function is utilized within the RBF network output layer.

The training of these networks is accomplished through backpropagation and a complex nonlinear hidden as well as output weights optimization. At iterations, the error of the network is assessed by forward propagating the inputs through the network and the derivative of this error is calculated with respect to each weight within the network.

The error function generally used in ANN computation is the squared difference between the actual and desired outputs. Optimization techniques, such as the scaled conjugate gradient method, are then used to minimize the error function by altering the weights, initially in the output layer and then the hidden layer. Essentially, the error is backpropagated from the output of the network, through the output weights and to the hidden weights [3]. Detailed explanations on these ANN architectures can be found in [3].

During the development of the above networks, over-fitting as well as under-fitting should be avoided. This can be accomplished by dividing the data into 3 sets. Over-fitting occurs when the network does not generalize but rather tends to memorize the training data. Under-fitting occurs when the network does not follow the data at all [3]. The data is divided into training, validation and test sets. The training data set is used to train the ANN to find the general pattern between its inputs and outputs. The validation data set is used to assess the network and the test data is used to confirm the prediction quality of the developed networks.

## *References.*


[1] "What is a Neural Network?", *Neuro Solution Technologies.*, Last accessed: 4 April 2006. [Online]. Available: http://www.neurosolutions.com/products/ns/whatisNN.html.

[2] S. Haykin, *Neural Networks: A Comprehensive Foundation*., Pearson Education India, 1998.

[3] C. M. Bishop, *Neural Networks for Pattern Recognition.*, Oxford University Press, 1995.

[4] I. T. Nabney, *Netlab: Algorithms for Pattern Recognition.*, Springer, 2002.

[5] R. Beale, T. Jackson., *Neural Computing-An introduction*., Taylor & Francis, 1990.

[6] L.M. Reyneri, M. Sgarbi., "Performance of Weighted Radial Basis Function Classifiers.", *Proceedings of European Symposium on Artificial Neural Networks.*, Italy, pp. 19-25, 1997.

[7] M-S. Chen., "Analysis and Design of the Multi-layer Perceptron using Polynomial Basis Functions.", Ph. D. Dissertation., University of Texas, 1991.

[8] W.Y. Huang and R.P. Lippmann., "Neural net and traditional classifier.", *Neural Information Processing Systems.*, American Institute of Physics, New York, pp. 387-396, 1988.

# Appendix D: Fuzzy Inference Systems.

Fuzzy logic was originally developed by Dr. Lotfi Zadeh. He published his seminal work on fuzzy set in 1965. In 1973 he proposed his theory of fuzzy logic [1].

Instead of classifying membership as either true or false as in a classical logic system, in fuzzy set theory, which is the foundation of Fuzzy Inference Systems (FISs), an input can belong to one or more fuzzy sets with a degree of membership [2]. The degree of membership is defined by fuzzy membership functions. Fuzzy logic also allows conclusions to be reached from inputs with a gradation of truth. Membership can be viewed as a representation of the "possibility" of association with the particular set [2].

One of the major advantages of fuzzy logic is its ability to be developed on top of the experience of experts within an industry [3]. In order to accomplish this, it uses heuristic rules to describe the available expert knowledge. These fuzzy inference rules are expressed in the form "IF A THEN B", where A is the premise and B is the consequence. The actions of the rules are executed or "fired" when the degree of membership of the inputs exceed certain threshold values. The threshold values define the minimum required membership of the inputs that an expert would expect for the particular rule to be executed and are generally defined by subjective criteria. Conflicting rules are allowed to fire jointly [2].

FISs are processes that utilize fuzzy logic to formulate a mapping from a given input to an output [4]. The mapping then provides a foundation from which decisions can be made. FISs have been successfully applied in fields such as automatic control, data classification, decision analysis and expert systems. Due to its multidisciplinary nature, these systems are associated with many names, such as fuzzy-rule-based systems, fuzzy expert systems, fuzzy modeling, fuzzy logic controllers as well as fuzzy systems [4].

A Fuzzy Inference System (FIS) involves Fuzzification, Inference and Defuzzification processes [4]. The Fuzzification process is a mapping from the observed input to the fuzzy sets defined in the corresponding universe. Inference process is a decision making logic that utilizes the fuzzy inference rules to determine fuzzy outputs corresponding to fuzzified inputs. Defuzzification produces non-fuzzy outputs [4].

There are 2 popular types of FISs. These are the Mamdani-type and Sugeno-type inference systems [4]. Mamdani-type inference system is the most commonly employed fuzzy methodology. It was proposed by Ebrahim Mamdani in 1975 [4]. The proposed methodology was based on a paper by Lofti Zadeh in 1973 on fuzzy algorithms for complex systems and decision processes. In the Mamdani-type inference system the fuzzy sets from the consequent of each rule are combined through the aggregation operator and the resulting fuzzy set is defuzzified to yield the output of the system [4]. The Sugeno-type or Takagi-Sugeno-Kang method of inference was introduced in 1985 [4]. In this type of FIS, the consequent of each rule is a linear combination of the inputs. The output is a weighted linear combination of the consequents. This inference methodology is similar to the Mamdani-type process in many respects. The initial Fuzzification and Inference processes of the inference techniques are exactly the same. These inference systems vary in the manner their outputs are determined. The Sugeno output membership functions are either linear or constant [4]. Mamdani-type inference systems are widely accepted, intuitive and well-suited to human input. However, Sugeno method of inference is computationally efficient; performs well with linear, optimization as well as adaptive techniques and is well-suited to mathematical analysis [4]. As a result, the FISs developed in this research utilized Sugeno-type inference systems.

Clustering of numerical data establishes the foundation of many classification and system modeling applications. The objective of clustering is to locate natural groupings in a set of given inputs such that similar inputs are congregated in the same class [4]. Utilizing data clustering to obtain fuzzy inference rules provides an advantage in that the resultant rules are more tailored to the data than a FIS generated without clustering [4].

There are 2 popular data clustering techniques. These are the fuzzy c-means and subtractive data clustering techniques [4]. The fuzzy c-means technique, introduced by Jim Bezdek in 1981, entails each data point belonging to a cluster to some degree that is specified by a membership grade [4]. This data clustering technique provides a method that illustrates the ability to group data points that populate a multidimensional space into a specific number of unique clusters. Fuzzy c-means technique requires 2 predefined clusters that are intended to indicate the mean location of each cluster [4]. Every data point is assigned a membership grade for each cluster. Due to the cluster centers and the membership grades for each data point being updated iteratively, the technique moves the cluster centers to the correct locations within the data set. This iteration involves minimizing a function that represents the distance from any given data point to a cluster center weighted by the membership grade of that data point [4].

Subtractive data clustering technique is a modified form of the Mountain Method for cluster estimation [5]. In this method, each data point is considered as a potential cluster center and defines a measure of the potential of a data point [6]. The measure of potential for a given point is a function of its distances to all other data points. A point with many neighbouring points will have a high potential value. After the potential of every data point has been computed, the point with the largest potential value is selected as the first cluster center. Thereafter, in order to determine the next cluster and its center, all the data points in the vicinity of the first cluster center, which is determined by a radius of influence or cluster radius, is removed. This process is iterated until all the input data are within a cluster radius of a cluster center [4]. Specifying a small cluster radius will usually yield many clusters in the data. However, specifying a large cluster radius will result in few cluster centers in the data [4]. The Sugeno -type inference systems developed in this research, employed subtractive clustering.

## *References.*


[1] "Lotfi Zadeh." Wikipedia. Wikipedia, 2005. *Answers.com* GuruNet Corp. Last accessed: 04 April 2006. [Online]. http://www.answers.com/topic/lotfi-zadeh.

[2] N. K. Kasabov, *Foundations of Neural Networks, Fuzzy Systems, and Knowledge Engineering.*, MIT Press, 1996.

[3] R. E. Uhrig, L H. Tsoukala, *Fuzzy and Neural Approaches in Engineering.*, John Wiley & Sons Ltd, 1997.

[4] The Mathworks, Inc., *Fuzzy Logic Toolbox User's Guide Version 2.*, The Mathworks, Inc, 1995.

[5] J. Yen, L. Wang, "Constructing optimal fuzzy models using statistical information criteria.", *Journal of Intelligent and Fuzzy Systems.*, vol. 7, pp. 185-201, 1999.

[6] S. Chiu, "Fuzzy Model Identification Based on Cluster Estimation.", *Journal of Intelligent and Fuzzy Systems.*, vol. 2, no. 3, pp. 267-278, 1994.

# Appendix E: Source Code.