A Model for Stock Price Prediction Using the Soft Computing Approach

BY

ADEBIYI AYODELE ARIYO
(CU04GP0044)

A THESIS SUBMITTED IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE AWARD OF DOCTOR OF PHILOSOPHY DEGREE (Ph.D) IN MANAGEMENT INFORMATION SYSTEM

OF THE DEPARTMENT OF COMPUTER AND INFORMATION SCIENCES, COLLEGE OF SCIENCE AND TECHNOLOGY COVENANT UNIVERSITY, OTA NIGERIA

MARCH, 2012

CERTIFICATION

This is to certify that this thesis is an original research work undertaken by **Adebiyi Ayodele Ariyo** under the supervisions of Professors C.K Ayo and S.O. Otokiti and that the work has not been submitted for the award of any other degree in this or any other institution.

1. Name: **Prof. Charles Korede Ayo**
   Supervisor

   Signature: ____________________ Date:_____________

2. Name: **Professor Sunday O. Otokiti**
   Co-Supervisor

   Signature: ____________________ Date:_____________

3. Name: **Professor Charles Korede Ayo**
   Head of Department

   Signature: ____________________ Date:_____________

DECLARATION

It is hereby declare that this research was undertaken by Ariyo Ayodele Adebiyi. The thesis is based on his original study in the Department of Computer and Information Sciences, College of Science and Technology, Covenant University, Ota, under the supervision of Prof. C.K. Ayo and Prof. S.O. Otokiti. Ideas and views of this research work are products of the original research undertaken by Ariyo Ayodele Adebiyi and the views of other researchers have been duly expressed and acknowledged.

Professor Charles Korede Ayo
Supervisor

Signature: ____________________ Date:_____________

Professor Sunday O. Otokiti
Co-Supervisor

Signature: ____________________ Date:_____________

DEDICATION

This thesis is dedicated to God Almighty the Alpha and Omega of my life and also to my late grandmother for her unfathomable love and passionate desire that I succeed in life.

ACKNOWLEDGEMENTS

My deep appreciation goes to God Almighty; the Alpha and Omega of my life. I can see vividly His hand upon my life. He is always there for me in all situations. I appreciate greatly His priceless help and the wisdom He granted me throughout my doctoral studies. My profound appreciation goes to the Chancellor, Dr. David O. Oyedepo and the members of the Board of Regents of Covenant University for the Vision and Mission of the University. Also, special thanks to the Management staff of the University: the Vice Chancellor, Prof. Aize Obayan, the Registrar, Mr. J.N. Taiwo, the Deans of the Colleges, Prof. F.K. Hymore and Prof. K. Shoremilekun for their commitment to the pursuit of excellence and sound academic scholarship.

My appreciation goes to my supervisor, Professor Charles Korede Ayo for his untiring efforts, guidance, unconditional supports and immense contributions to the thesis. I equally want to thank him especially for his fatherly disposition to all his postgraduate students, his genuine concern both academically and spiritually. His leadership style is unparalleled and worthy of emulation. He is always interesting in providing opportunities for young academics, particularly those interested in Software Engineering and Intelligent System research. Many thanks goes to my co-supervisor, Prof. Sunday O. Otokiti, the former Head of Department of Business Studies at Covenant University, now the Dean of College of Business and Social Sciences, Landmark University, Omu-Aran for his enormous contributions, support and encouragement throughout the period of this study.

I also want to thank the Dean of College of Science and Technology, Prof. F.K. Hymore and Dean of Postgraduate Studies, Prof. C.O. Awonuga for their encouragement and support throughout the period of this study. I thank every member of staff of Department of Computer and Information Sciences for their support; in particular Dr. N.A. Omoregbe and Dr. J.O. Daramola for their contributions at the early stage of this study.

My appreciation goes to the Librarian for the useful materials obtained in the course of this work and the management of yahoo finance (http:/finance.yahoo.com) and cashcraft (www.cashcraft.com) for the live data used in this study.

I wish to appreciate the following people:. Firstly, my father, Elder Abiodun Adebiyi for his prayers and blessings in the course of this study and my brother and sister, Abayomi Adebiyi and Kehinde Adebiyi for their support and encouragement. Secondly, my in-laws, are wonderful people, in particular my mother-in-law, Mrs. Alice Ebun Owolabi, I am grateful to her for her prayers and support. Also, I appreciate my in-laws Dr. Henry Owolabi, Pastor Femi Owolabi and Pastor & Dr. (Mrs.) Olasehinde for their love and constant encouragement.

Finally, I am indeed grateful for the unflinching support of my beloved wife, Mrs. Marion Olubunmi Adebiyi for her love, encouragement and commitment to the success of this study. To my wonderful children, Ireoluwa Joy Adebiyi and Ayomide Iranwooluwa Joshua Adebiyi, thank you for being part of me and my success.

ABSTRACT

A number of research efforts had been devoted to forecasting stock price based on technical indicators which rely purely on historical stock price data. However, the performances of such technical indicators have not always satisfactory. The fact is, there are other influential factors that can affect the direction of stock market which form the basis of market experts' opinion such as interest rate, inflation rate, foreign exchange rate, business sector, management caliber, investors' confidence, government policy and political effects, among others.

In this study, the effect of using hybrid market indicators such as technical and fundamental parameters as well as experts' opinions for stock price prediction was examined. Values of variables representing these market hybrid indicators were fed into the artificial neural network (ANN) model for stock price prediction.

The empirical results obtained with published stock data show that the proposed model is effective in improving the accuracy of stock price prediction. Also, the performance of the neural network predictive model developed in this study was compared with the conventional Box-Jenkins autoregressive integrated moving average (ARIMA) model which has been widely used for time series forecasting. Our findings revealed that ARIMA models cannot be effectively engaged profitably for stock price prediction. It was also observed that the pattern of ARIMA forecasting models were not satisfactory. The developed stock price predictive model with the ANN-based soft computing approach demonstrated superior performance over the ARIMA models; indeed, the actual and predicted value of the developed stock price predictive model were quite close.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER ONE

INTRODUCTION

1.1 Background of the Study

The ability to accurately predict the future is crucial for decision processes in planning, organizing, scheduling, purchasing, strategy formulation, policy making and supply chains management and so on. Therefore, prediction/forecasting is an area where a lot of research efforts have been carried out in the past. This area is presently still an important and active field of human activity and will continue to be in the future (Zhang, 2004). Stock price prediction has always been a subject of interest for most investors and financial analysts, but clearly, finding the best time to buy or sell has remained a very difficult task for investors because there are other numerous factors that may influence stock prices (Weckman et al., 2008, Chang and Liu, 2008 and Adebiyi et al 2009). Therefore, stock market prediction has remained an important research topic in business. However, stock markets environments are very complicated, dynamic, stochastic and thus difficult to predict (Wei, 2005; Gerasimos et al., 2005; Yang and Wu, 2006; Tsanga et al., 2007; and Tae 2007).

Presently, financial forecasting is regarded as one of the most challenging applications of time series forecasting. Financial time series presents complex behaviour, resulting from a huge number of factors, which could be economic, political, or psychological. They are inherently noisy, non-stationary, and deterministically chaotic.

Data mining technology has found increasing acceptance in business areas that need to analyze large amounts of data in order to discover knowledge which could not be found using traditional methods. Time series data mining is identified as one of the 10 challenging problems in data mining research (Yang and Wu, 2006; Lay-Ki and Sang, 2007). Financial time series forecasting has been a subject of research since 1980s. The objective is to beat financial markets and win much profit. However, due to complexity of financial time series, there is some skepticism about predictability of financial time series. This is reflected in the well-known Efficient Market Hypothesis theory (EMH). According to the EMH theory, the current price is the best prediction for the next day, and buy-hold is the best trading strategy.

However, there are strong evidences which refuse the efficient market hypothesis (Pegal, 2007). Due to dynamic nature and unpredictable environment of stock market domain, predicting the future has always been the desire of mankind because of their inability to deal with uncertain, fuzzy, or insufficient data which fluctuate rapidly in very short periods of time. Artificial neural networks (ANNs) have become a very important method for stock market predictions (Schoeneburg, 1990). In recent years, soft computing techniques such as Artificial Neural Networks, Fuzzy Logic, and Genetic Algorithms have gained popularity for this kind of application. Much research efforts have been made to improve the predictive accuracy and computational efficiency of share values (Xiaodan Wu et al, 2001).

The elasticity and adaptability advantages of the artificial neural network models have attracted the interest of many other researchers apart from Business and Banking efforts. Other interested disciplines include the electrical engineering, robotics and computer engineering, oil and medicine industries. For the last decade, the artificial neural network models have been heavily used in the fields of business, finance and economics for several purposes like time series forecasting and performance measurement (Avci, 2007).

Artificial neural networks (ANNs) and fuzzy logic (FL) are two of the key technologies that have received growing attention in solving real world, nonlinear, time-variant problems. The need to solve highly nonlinear, time-variant problems has been on the increase as many of today's applications have nonlinear and uncertain behaviour which changes with time like stock market (Robert, 1995 and Emdad, 2000).

The several distinguishing features of ANNs make them attractive and widely used for forecasting task in the domain of business, economic, and finance applications. The reasons for this are not far fetched. First, artificial neural networks are data-driven self-adaptive methods in that there are few apriori assumptions made about the models for problems under study. Secondly, artificial neural networks can be generalized after learning from the data presented to them and correctly infer unseen part of the population. Thirdly, ANNs are universal approximators in that it has been shown that a network can approximate any continuous function to any desired accuracy. Finally, ANNs are strong in solving nonlinear problems. Traditional techniques for time series predictions, such as the Box-Jenkins or Autoregressive Integrated Moving Average

(ARIMA) assumed that the time series under study are generated from linear processes which is unrealistic because real-world systems are often nonlinear (Zhang et al., 1998, Khasei et al., 2009, Mehdi and Mehdi, 2010).

ANN theory grew out of artificial intelligence research, or the research in designing machines with cognitive ability. An ANN is a computer program or hardwired machine that is designed to learn in a manner similar to the human brain. Haykin 1994 in his contribution describes ANN as an adaptive machine or more specifically; “a massively parallel distributed processor that has a natural propensity for storing experiential knowledge and making it available for use”. He likened the ANN to the brain in two respects: (a) Knowledge is acquired by the network through a learning process and interneuron connection strengths known as synaptic weights, which can be used to store the knowledge and (b) artificial neural network is able to work parallel with input variables and consequently handle large sets of data swiftly.

The principal strength of the network is its ability to find patterns and irregularities as well as detecting multi-dimensional non-linear connections in data. The latter quality is extremely useful for modeling dynamic systems, e.g. the football league result prediction and stock market behaviour. Apart from that, neural networks are frequently used for pattern recognition tasks and non-linear regression (Nygren, 2004). ANN can be used to predict stock market prices because they are able to learn nonlinear mappings between inputs and outputs. Contrary to the efficient market hypothesis theory, several researchers claim the stock market and other complex systems exhibit chaos. Chaos is a nonlinear deterministic process which only appears randomly because it can not be easily expressed. With neural networks' ability to learn nonlinear and chaotic systems, it may be possible to outperform traditional analysis and other computer-based methods (Ramon, 1997).

Financial forecasting is of considerable practical interest. Due to artificial neural networks ability to mine valuable information from a mass of historical information and be efficiently used in financial areas, the applications of artificial neural networks to financial forecasting have been very popular over the last few years (Widrow et al., 1994; Refenes, 1995; Gately, 1996; Yao et al., 1999; Kate and Gupta, 2000; Abu-Mostafa et al., 2001; and Defu et al., 2005).

In this study artificial neural network which is one of the soft computing paradigms will be used to develop stock price predictive model. However, from literature survey, previous research efforts on stock market prediction had engaged predominantly technical indicators for forecasting of stock prices; the impact of fundamental analysis variables has been largely ignored. The fundamental analysis is based on financial status and performance of the company. The technical analysis is based on the historical financial time series data. In this study, we explore the combination of the technical indicators, fundamental indicators and experts opinion for stock market prediction with the objective of attaining improved stock market prediction.

1.2 STATEMENT OF THE PROBLEM

Generally in stock markets, investors are often faced with difficulties of inability to:

i) determine and predict the stock market behaviour due to the dynamism and unpredictable environment of stock market domain.

ii) take decision on the appropriate stock to buy or sell for better profit due to unpredictable nature of stock markets.

iii) analyze and extract useful knowledge from a vast amount of information in order to make qualitative investment decision.

iv) engage effectively technical trading strategies and buy-hold strategy.

Hence, in this study, an ANN-based predictive model to overcome the problems stated above, and profer solutions to the following research questions, is being proposed.

i) Could our proposed stock price predictive model be effective in improving the accuracy of stock price prediction with the combination of the parameters of technical, fundamental analysis and experts' opinion variables?

ii) Could our proposed stock price predictive model enhance investment decisions of investors?

1.3 OBJECTIVES OF THE RESEARCH

The aim of this research work is to develop an improved predictive model for stock price prediction using the soft computing approach for hybridized market indicators with a view to increase forecasting accuracy for stock prices.

Consequently, the objectives of the research are to:

i. develop a predictive model for stock price prediction using hybridized market indicators for enhanced decision making.

ii. compare the predictive performance of the statistical technique of autoregressive integrated moving average (ARIMA) and the proposed ANN-based predictive model.

iii. evaluate the proposed stock price predictive model with performance measures.

1.4 METHODOLOGY

The research methodology used in this study is as follows:

i. The forecasting techniques engaged in this study were ARIMA model and soft computing technique of the artificial neural network with multilayer perceptron (MLP) model trained with backpropagation (BP) algorithm. The procedures involved in using ARIMA model for forecasting include: (1) data collection and examination; (2) testing for stationarity; (3) model identification and estimation; (4) model diagnostics checking; and (5) forecasting and forecast evaluation. Similarly, the steps required for ANN model in developing financial predictive model include: (1) variable selection; (2) data collection; (3) data preprocessing; (4) training, testing, and validation sets; (5) neural network design (number of hidden layers, number of hidden neurons, number of output neurons, transfer functions); (6) evaluation criteria; (7) neural network training (number of training iterations, learning rate and momentum); and (8) implementation.

ii. The stock data used in this study were obtained from New York Stock Exchange (NYSE) and Nigerian Stock Exchange (NSE) respectively. The historical stock data of four different companies, two from each of the stock exchange mentioned were used. The two companies' stock data from NYSE are Dell Inc. and Nokia Inc. relatively from information technology industry sector. The companies from NSE are from banking industry sector which are UBA bank and Zenith bank. The technical data used in this study are raw daily opening price, highest price, lowest price, closing price and volume traded in each day. The fundamental data used consist of price per earning, return on asset and return equity and the expert opinions were obtained through interactions with financial experts and through administration of questionnaires.

iii. Some performance measures: root mean square error, mean square error, and confusion matrix; were used to evaluate the predictive models developed.

iv. The predictive models were implemented using Eviews software for ARIMA model and Matlab for artificial neural network model.

1.5 SIGNIFICANCE OF THE STUDY

The stock price predictive model developed in this study is of immense benefits to the stakeholders such as traders, investors and stock brokers in stock market domain. It can serve as a useful guide to individual investors in making investment decision on which stock to buy or sell. Moreover, it can enable individual investors to increase wealth through profit gained from usage of the predictive model. Furthermore, it will stimulate interest of individuals to invest in stock market indexes thereby making the sector vibrant, robust and healthy.

1.6 MOTIVATION FOR THE STUDY

The motivation for this study stems from the following reasons: firstly, investors in stock market are desirous to make profits from their investment; however, lack of adequate knowledge of the right stock to buy or sell at the right time poses a big challenge. Secondly, to further contradict the hypothesis formulated in stock market known as the Efficient Market Hypothesis, which says there is no way to make profit by predicting the stock market.

1.7 CONTRIBUTION TO KNOWLEDGE

The specific contributions of this research work pertain to stock forecasting modeling in stock market domain both at local and global levels. Firstly, the study provides an improved predictive model for stock price prediction using the soft computing approach with hybrid market indicators that combined the parameters of technical and fundamental analyses as well as experts' opinion.

Secondly, this study was able to resolve and clarify contradictory findings reported in literature on the superiority of statistical techniques of ARIMA model over soft computing technique of ANN model in time series prediction and vice-versa. The findings in this study showed that ANN model outperformed the statistical forecasting techniques, and in particular, ARIMA, which is the most widely used statistical forecasting technique.

1.8 LIMITATION AND SCOPE OF THE STUDY

In the study only one soft computing technique, namely ANN, was used. Furthermore the composition of expert's opinion was limited to five parameters. Also, Eviews and Matlab were used to simulate the proposed model.

1.9 ORGANIZATION OF STUDY

The organization of the thesis is summarized as follows: Chapter one is the introduction which is composed of background information of the study, statement of the problem and research question, aim and objective of the study, research methodology, significance of the study, motivation for the study, limitation and scope of the study. Chapter two presents literature review – soft computing theories, related works with use of soft computing in stock prediction and gaps in literature. Chapter three consists of research methodology used in this study which includes sources of data, details of forecasting techniques employed and performance measures used to evaluate the predictive model. Chapter four presents detailed results of the study and finally, chapter five is composed of summary, conclusion and future research work.

CHAPTER TWO

LITERATURE REVIEW

2.1 INTRODUCTION

This chapter provides a detailed background and understanding of different soft computing (SC) models and statistical techniques that are widely used in time series prediction. Some related works of the applications of stock price prediction in literature were reviewed and discussed.

This study involves the development of an improved predictive model with the SC paradigm in particular ANN model with hybrid market indicators. In this work, we compare our results with that of a popular statistical technique ARIMA model and hence a brief theory of this statistical technique is provided in this chapter following the theories of SC paradigms. Stock market is the application area in this thesis and therefore the extensive review of the literature is performed focusing on many different individual approaches of the SC paradigms applied in stock price prediction. We review some of the key papers that have been highly influential to the application area and report the findings.

2.2 BACKGROUND OF SOFT COMPUTING TECHNIQUES

Soft computing is an important branch of the study in the area of intelligent and knowledge-based system. It differs from conventional (hard) computing in that; it is tolerant of impression, uncertainty, partial truth and approximation. In effect, the role model soft computing is the human mind. Humans can effectively handle incomplete, imprecise, and fuzzy information in making intelligent decisions. Artificial neural networks (ANN), fuzzy logic (FL), evolutionary computing (EC), and probability theory (PT) are the techniques of soft computing for knowledge representation and for mimicking the reasoning and decision-making processes of a human (Karray and De Silva, 2004).

2.2.1 **Artificial Neural Network**

The immense capabilities of the human brain in processing information and making instantaneous decisions, even under very complex circumstances and uncertain environments have inspired researchers in studying and possibly mimicking the computational abilities of the human brain to build computational systems, which can process information in a similar way. Such systems are called artificial neural networks (Karray and De Silva, 2004).

An Artificial Neural Network (ANN) or simply a neural network (NN) is an information processing paradigm that is inspired by the way biological nervous systems, such as the brain, process information. The key element of this paradigm is the novel structure of the information processing system. It is composed of a large number of highly interconnected processing elements (neurons) working in unison to solve specific problems. ANNs, like people, learn by example. An ANN is configured for a specific application, such as pattern recognition or data classification, through a learning process. Learning in biological systems involves adjustments to the synaptic connections that exist between the neurons. This is true of ANNs as well (Stergious and Siganos, 2009).

ANN can also be defined as a biologically inspired computational model which consists of processing elements (called neurons) and connections between them with coefficients (weights) bound to the connections, which constitute the neuronal structure, and training and recall algorithms attached to the structure. Neural networks are called connectionist models because of the main role of the connections in them. The connection weights are the memory of the system (Nikola, 1998).

Advantages and Disadvantages of Neural Networks

Neural networks, with their remarkable ability to derive meaning from complicated or imprecise data, can be used to extract patterns and detect trends that are too complex to be noticed by either humans or other computer techniques. A trained neural network can be thought of as an expert in the category of information it has been given to analyze. This expert can then be used to provide projections given new situations of interest. Other advantages of artificial neural network include (Nikola, 1998):

a. Learning - a network can start with no knowledge and can be trained using a given set of data examples, that is, input-output pairs (a supervised training), or only input data (unsupervised training); through learning, the connection weights change in such a way that the network learns to produce desired outputs for known inputs; learning may require repetition.

b. Generalization - if a new input vector that differs from the known examples is supplied to the network, it produces the best output according to the examples used.
c. Massive potential parallelism - during the processing of data, many neurons "fire" simultaneously.
d. Robustness - if some neurons "go wrong", the whole system may still perform well.
e. Partial match, is what is required in many cases as the already known data do not coincide exactly with the new facts.

These main characteristics of artificial neural networks make them useful for knowledge engineering. Artificial neural networks can be used for building expert systems.

The main drawbacks of artificial neural networks are (Emdad, 2000):

a. Black box nature – the relationship of weight changes with the input-output behaviour during training and use of trained system to generate correct outputs using the weights. Our understanding of the “black box” is incomplete compared to a fuzzy rule based system description.
b. High cost of implementation – it may not provide most cost effective solution, ANN implementation is typically more costly than other technologies. A software solution generally takes a long time to process.
c. Inability to determine structure size – it is difficult, if not impossible to determine the proper size and structure of a neural network to solve a given problem. Also, ANN does not scale well. Manipulating learning parameters for learning and convergence becomes increasingly difficult.

Similarities between Biological Neuron and Artificial Neuron

For years neurobiologists and psychologists have tried to understand how the human brain works. This attempt led to the creation of a cognitive science, also known as artificial intelligence (AI). The human brain is probably the most powerful computer that mankind has ever known. It is the greatest and most complex biological component. The brain consists of approximately 10^{10} neurons that communicate with each other through $\pm 10^{10}$ synapses. The brain is an organ that makes appropriate decisions based on analysis of information received from the environment. The information input is communicated between different neurons each of which sends and receives signals from neighbouring neurons.

The basic element of the nervous system is the neuron as shown in figure 2.1. The neuron is composed of the dendrites (which receives signals from other neurons), cell body or soma (which adds impulses received from different dendrites), and axons which serve as channels through which signals are transmitted to other neurons. Each neuron is connected to others through synaptic junctions, or synapses. Information is transmitted from one neuron to another through the synapses. The synapse can be seen as a point of contact between two neurons.

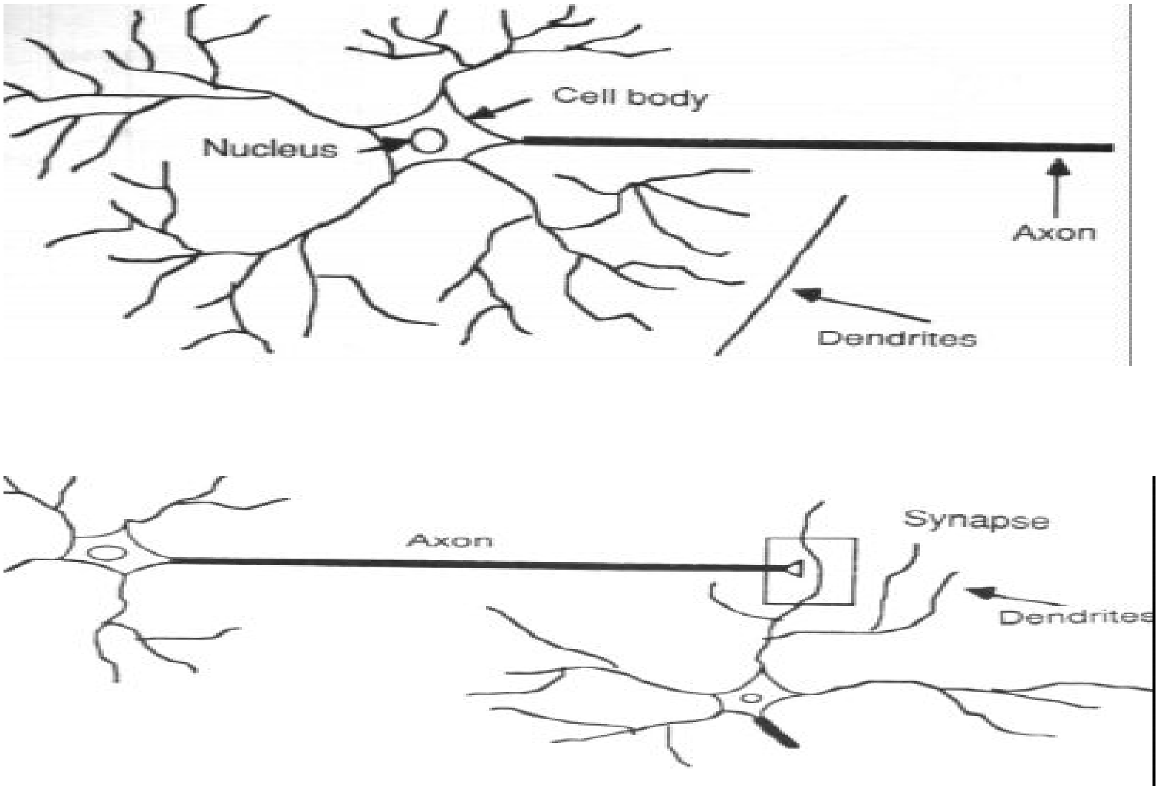


Figure 2.1: Biological Neuron Source: (Stergious and Siganos, 2010)

The transmission of signals from one neuron to another is effectively through the activation of the receiving neuron by means of electrical impulses. The activated neuron send impulses onto others. This process takes place among a large number of neurons, all of which become simultaneously activated. This leads to the creation of thoughts or actions. The brain performs this task on a continuous basis. The exceptional characteristic of the human brain is its ability to learn from the past, which is facilitated by the complex system of sending and receiving electrical impulses among neurons.

Artificial intelligence in the cognitive science endeavours to simulate neuron activity in the brain by building a system that would learn through experience. The construction of a network, known as an artificial neural network simulates brain characteristics. Neural derives from neuron, and artificial from the fact that it is not biological. Unlike the brain, the ANN performs discrete operations, which are made possible with electronic computers' ability to swiftly perform complex operations (Alain, 2002).

A neural network is a computational technique that benefits from techniques similar to ones employed in the human brain. It is designed to mimic the ability of the human brain to process data and information and comprehend patterns. It imitates the structure and operations of the three dimensional lattice of network among brain cells (nodes or neurons, and hence the term neural). The technology is inspired by the architecture of the human brain, which uses many simple processing elements operating in parallel to obtain high computation rates. Similarly, the neural network is composed of many simple processing elements or neurons operating in parallel whose function is determined by network structure, connection strengths, and the processing performed at computing elements or nodes. The learning process of the neural network can be likened to the way a child learns to recognize patterns, shapes and sounds, and discerns among them. For example, the child has to be exposed to a number of examples of a particular type of tree for her to be able to recognize that type of tree later on. In addition, the child has to be exposed to different types of trees for her to be able to differentiate among trees (Shaikh, 2004).

The human brain has the uncanny ability to recognize and comprehend various patterns. The neural network is extremely primitive in this aspect. The network's strength, however, is in its ability to comprehend and discern subtle patterns in a large number of variables at a time without being stifled by detail. It can also carry out multiple operations simultaneously. Not only can it identify patterns in a few variables, it also can detect correlations in hundreds of variables. It is this feature of the network that is particularly suitable in analyzing relationships between a large number of market variables. The networks can learn from experience. They can cope with fuzzy patterns – patterns that are difficult to reduce to precise rules. They can also be retrained and thus can adapt to changing market behavior.

The network holds particular promise for econometric applications. Multilayer feedforward networks with appropriate parameters are capable of approximating a large number of diverse functions arbitrarily well. Even when a data set is noisy or has irrelevant inputs, the networks can learn important features of the data. Inputs that may appear irrelevant may in fact contain useful information. The promise of neural networks lies in their ability to learn patterns in a complex signal.

Much is still unknown about how the brain trains itself to process information, so several theories abound. In the human brain, a typical neuron collects signals from others through a host of fine structures called *dendrites*. The neuron sends out spikes of electrical activity through a long, thin strand known as an *axon*, which splits into thousands of branches. At the end of each branch, a structure called a *synapse* converts the activity from the axon into electrical effects that inhibit or excite activity from the axon into electrical effects that inhibit or excite activity in the connected neurons. When a neuron receives excitatory input that is sufficiently large compared with its inhibitory input, it sends a spike of electrical activity down its axon. Learning occurs by changing the effectiveness of the synapses so that the influence of one neuron on another changes.

Neurons are the building blocks of an ANN. Each neuron accepts many inputs and produces an output that is the result of some processing. Neurons are interconnected by links that are weighted, where the weight represents the strength of the connection. Figure 2.2 gives analogy of a computational neuron to a biological neuron. An artificial neuron's inputs are a man-made version of a biological neuron's dendrites. Synapse activities are represented by weights on the edges, somas are represented as mathematical equation shown in the circle, and the axon is the calculated output of each neuron.

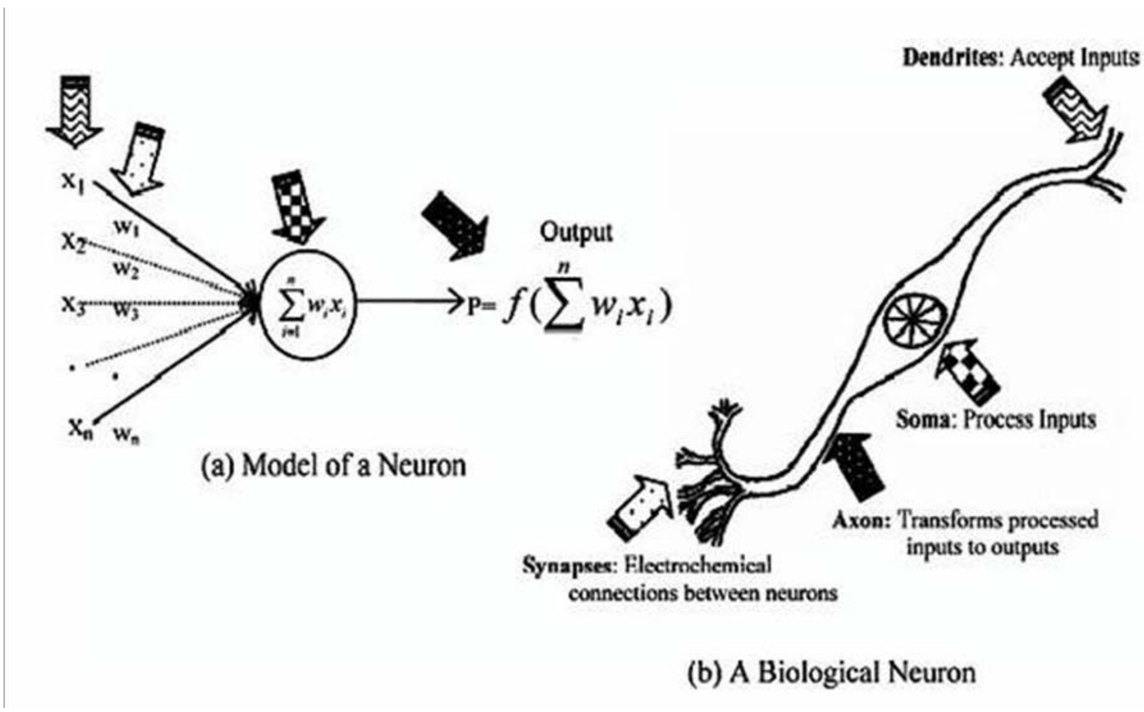


Figure 2.2: Model of a neuron Source: Hassan, 2007

Network Architecture

Neurons form interconnected networks, and thus, the way they are connected is best visualized using network architecture. Depending on the position of neurons in an ANN, there exist three layers: the input layer, the hidden layer and the output layer. The number of hidden layers may range from zero to as many as the complexity of the problem requires. Depending on the problem being solved, researchers have presented various network architectures. Architectures that are most commonly used are: feed-forward fully connected NN, recurrent NN, and time-delay NN.

Feed-forward networks

Feed-forward ANNs (figure 2.3) allow signals to travel one way only; from input to output. There is no feedback (loops) i.e. the output of any layer does not affect that same layer. Feed-forward ANNs tend to be straight forward networks that associate inputs with outputs. They are extensively used in pattern recognition. This type of organisation is also referred to as bottom-up or top-down.

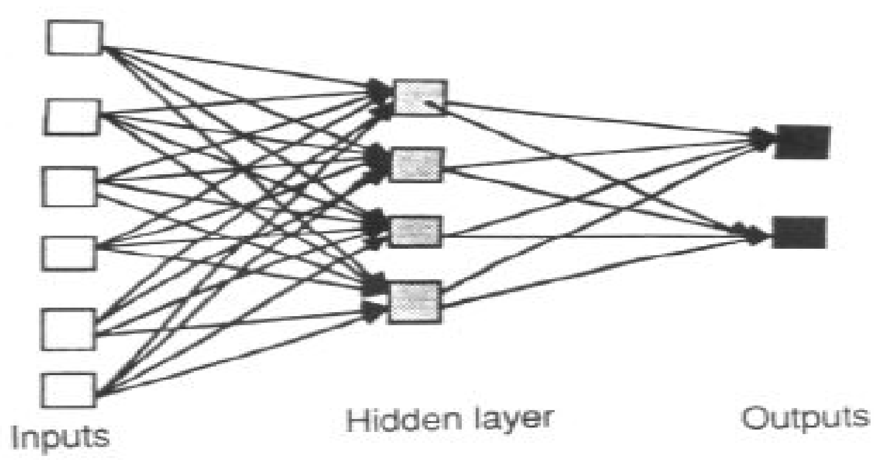


Figure 2.3: An example of a simple feedforward network

Feedback networks

Feedback networks can have signals traveling in both directions by introducing loops in the network. Feedback networks are very powerful and can get extremely complicated. Feedback networks are dynamic; their state is changing continuously until they reach an equilibrium point. They remain at the equilibrium point until the input changes and a new equilibrium needs to be found. Feedback architectures are also referred to as interactive or recurrent, although the latter term is often used to denote feedback connections in single-layer organizations

Network layers

The commonest type of artificial neural network consists of three groups, or layers, of units: a layer of *input* units is connected to a layer of *hidden* units, which is connected to a layer of *output* units (see figure 2.3). The activity of the input units represents the raw information that is fed into the network. The activity of each hidden unit is determined by the activities of the input units and the weights on the connections between the input and the hidden units. The behaviour of the output units depends on the activity of the hidden units and the weights between the hidden and output units.

This simple type of network is interesting because the hidden units are free to construct their own representations of the input. The weights between the input and hidden units determine when each hidden unit is active, and so by modifying these weights, a hidden unit can choose what it represents. The network architecture can be either single-layer or multi-layer architectures.

The Learning Process

ANNs are unique amongst mathematical processing methods in that they can learn from data characteristics, and adapt network parameters according to underlying structures in the training dataset. The process is called learning. All learning methods used for neural networks can be classified into three major categories:

1. Supervised learning: The training examples comprise input vectors x and the desired output vectors y. Training is performed until the neural network learns to associate each input vector x to its corresponding and desired output vector y; for example, a neural network can learn to approximate a function $y = f(x)$ represented by a set of training examples (x, y). It encodes the examples in its internal structure.

2. Unsupervised learning: Only input vectors x are supplied; the neural network learns some internal features of the whole set of all the input vectors presented to it.

3. Reinforcement learning: Sometimes called reward-penalty learning, is a combination of the above two paradigms; it is based on presenting input vector x to a neural network and looking at the output vector calculated by the network. If it is considered good, then a reward is given to the network in the sense that the existing connection weights are increased; otherwise the network is punished, the connection weights, being considered as not appropriate set, decrease. Thus reinforcement learning is learning with a critic, as opposed to learning with a teacher.

Transfer Function

The behaviour of an ANN depends on both the weights and the input-output function (transfer function) that is specified for the units. This function typically falls into one of three categories:

- linear
- threshold (Tanh)
- sigmoid

For **linear units**, the output activity is proportional to the total weighted output. The equation is a = n and the range = range of n. The graphical representation is depicted in figure 2.4 below:

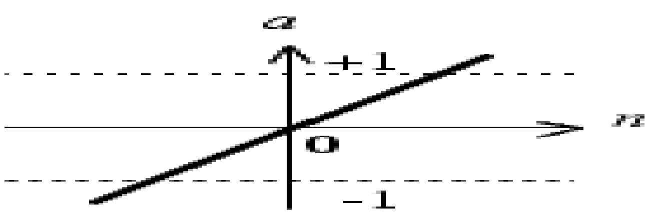


Figure 2.4: A diagram of the linear function

For **threshold units,** the output is set at one of two levels, depending on whether the total input is greater than or less than some threshold value. The equation is a = tanh(n) and the range is [-1, +1]. The graphical representation is depicted in figure 2.5 below:

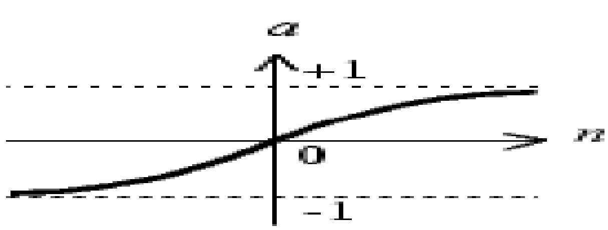


Figure 2.5: A diagram of the Tanh function

For **sigmoid units**, the output varies continuously but not linearly as the input changes. Sigmoid units bear a greater resemblance to real neurons than do linear or threshold units.

The equation is $a = \frac{1}{1+e^{-n}}$ and the range [0, +1]. The graphical representation is depicted in figure 2.6 below:

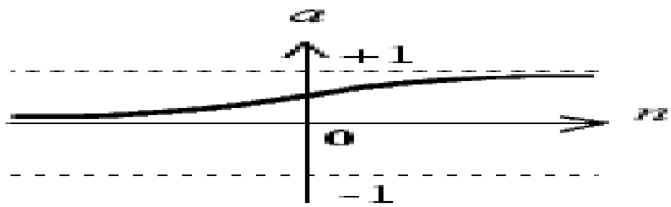


Figure 2.6: A diagram of the sigmoid function

To make a neural network performs some specific task, we must choose how the units are connected to one another and we must set the weights on the connections appropriately. The connections determine whether it is possible for one unit to influence another. The weights specify the strength of the influence.

Learning Algorithms

In order to train a neural network to perform some task, we must adjust the weights of each unit in such a way that the error between the desired output and the actual output is reduced. This process requires that the neural network compute the error derivative of the weights. In other words, it must calculate how the error changes as each weight is increased or decreased slightly. The back propagation algorithm is the most widely used method for determining the error derivative of the weights.

2.2.2 Fuzzy logic

One way to represent inexact data and knowledge, closer to humanlike thinking, is to use fuzzy rules instead of exact rules when representing knowledge. Fuzzy systems are rule-based expert systems based on fuzzy rules and fuzzy inference. Fuzzy rules represent in a straightforward way commonsense knowledge and skills, or knowledge that is subjective, ambiguous, vague, or contradictory. This knowledge might have come from many different sources. Commonsense knowledge may have been acquired from long-term experience, from the experience of many people, over many years (Nikola, 1998).

A fuzzy logic system consists of three main blocks: fuzzification, inference mechanism, and defuzzification. These components of fuzzy logic system are briefly described below (Branco and Dente, 2000).

2.2.2.1 **Fuzzification**

Fuzzification is a mapping from the observed numerical input space to the fuzzy sets defined in the corresponding universe of discourse. The *fuzzifier* maps a numerical value denoted by $x' = (x_1, x_2, ..., x_m)$ into fuzzy sets represented by membership functions in U. These functions are Gaussian, denoted by $\mu_{A_j^i}(x_j')$ as we expressed in equation (2.1).

$$\mu_{A_j^i}(x_j') = a_j^i \exp\left[-\frac{1}{2}\left(\frac{x_j - b_j^i}{c_j^i}\right)^2\right] \tag{2.1}$$

where $1 \le j \le m$ refers to the variable (*j*) from m considered input variables; $1 \le i \le n_j$ considers the *i* membership function among all n_j membership functions considered for variable (*j*); a_j^i defines the maximum of each Gaussian function, here $a_j^i = 1.0; b_j^i$ is the center of the Gaussian function; and c_j^i defines its shape width.

2.2.2.2 **Inference Mechanism**

Inference mechanism is the fuzzy logic reasoning process that determines the outputs corresponding to the fuzzified inputs.

The fuzzy rule-based is composed by IF-THEN rules like

$$\mathrm{R}^{(l)}: \text{IF } (x_1 \text{ is } A_1^{(l)} \text{ and } x_2 \text{ is } A_2^{(l)} \text{ and} \ldots x_m \text{ is } A_m^{(l)}) \text{ THEN } (y \text{ is } w^{(l)}),$$

where: $\mathrm{R}^{(l)}$ is the *l*th rule with $1 \le l \le c$ determining the total number of rules; $x_1, x_2, ..., x_m$ and y are respectively, the input and output system variable; $A_j^{(l)}$ are the antecedent linguistic terms in rule (*l*) with $1 \le j \le m$ being the number antecedent variables; and $w^{(l)}$ is the rule conclusion for that type of rules, a real number usually called fuzzy singleton. The conclusion, a numerical value can be considered as a pre-defuzzified output that helps to accelerate the inference process. The reasoning process combines all rule contributions $w^{(l)}$ using the centroid defuzzification formula in a weighted form, as indicated in equation (2.2). The equation maps input process states $(x_j^{'})$ to the value resulting from inference function $Y(x^{'})$.

$$Y(x^{'}) = \frac{\sum c_{l=1}\left(\prod_{j=1}^{m} \mu_{A_j^{(l)}}(x_j^{'})\right) w^{(l)}}{\sum_{l=1}^{c}\left(\prod_{j=1}^{m} \mu_{A_j^{(l)}}(x_j^{'})\right)} \tag{2.2}$$

2.2.2.3 **Defuzzification**

Basically, defuzzification maps output fuzzy set defined over an output universe of discourse to crisp outputs. The common defuzzification strategies are briefly described below:

a. The Max Criterion Method

The max criterion method produces the point at which the possibility distribution of the fuzzy output reaches a maximum value.

b. The Mean of Maximum Method

The mean of maximum generates an output which represents the mean value of all local inferred fuzzy outputs whose membership functions reach the maximum. In the case of a discrete universe, the inferred fuzzy output may be expressed as:

$$z_0 = \sum_{j=1}^{l} \frac{w_j}{l} \tag{2.3}$$

where w_j is the support value at which the membership function reaches the maximum value $\mu_z(w_j)$ and *l* is the number of such support values.

c. The Center of Area Method

The center of area generates the center of gravity of the possibility distribution of the inferred fuzzy output. In the case of a discrete universe, this method yields:

$$z_0 = \frac{\sum_{j=1}^{n} \mu_z(w_j) w_j}{\sum_{j=1}^{n} \mu_z(w_j)} \tag{2.4}$$

where *n* is the number of quantization levels of the output.

Advantages of fuzzy logic

a. Fuzzy logic converts complex problems into simpler problems using approximate reasoning.

b. A fuzzy logic description can effectively model the uncertainty and nonlinearity of a system.

c. Fuzzy logic is easy to implement using both software on existing microprocessor or dedicated hardware.

d. Fuzzy logic based solutions are cost effective for a wide range of applications such as home appliances when compared to traditional methods.

Disadvantages of fuzzy logic

a. For complex system, it becomes more difficult to determine the correct set of rules and membership functions to describe the system.

b. The use of fixed geometric shaped membership functions in fuzzy logic limits system knowledge more in the rule base and the membership function base. This results in requiring more system memory and processing time.

c. Fuzzy logic uses heuristics algorithms for defuzzification, rule evaluation, and antecedent processing. Heuristic algorithms can cause problems mainly because it does not guarantee satisfactory solutions that operate under all possible conditions.

d. The generalization of fuzzy logic is poor compared with artificial neural networks. The generalization capability is important in order to handle unforeseen circumstances.

e. Once the rules are determined, they remain fixed in the fuzzy logic controller, which is unable to learn.

f. Conventional fuzzy logic cannot generate rules that will meet a pre-specified accuracy. Accuracy is improved only by trial and error.

g. Conventional fuzzy logic does not incorporate previous state information (very important for pattern recognition, like speech) in the rule base.

2.2.3 **Evolutionary Computing**

Genetic algorithms is a part of evolutionary computing, which is a rapidly growing area of artificial intelligence. Evolutionary computing (EC) represents another tool of soft computing techniques based on the concepts of artificial evolution. Generally speaking, evolution is the process by which life adapts to changing environments. The offspring of an organism must inherit enough of its parent's characteristics to remain viable while introducing some differences which can cope with new problems presented by its surroundings. Naturally, some succeed and others fail. Those surviving have the chance to pass characteristics on to the next generation. A creature's survival depends, to a large extent, on its fitness within its environment, which is in turn determined by its genetic makeup. Researchers have sought to formalize the mechanisms of evolution in order to apply it artificially to very different types of problem. The pursuit of artificial evolution using computers has led to the development of an area commonly known as evolutionary computation or evolutionary algorithms.

Evolutionary computation or evolutionary computing is a broad term that covers a family of adaptive search population-based techniques that can be applied to the optimization of both discrete and continuous mappings. This computational paradigm includes such techniques as evolutionary programming, evolutionary strategies, genetic programming, and genetic algorithms (Karray and De Silva, 2004). A well known instance of an EC is sometimes a Genetic Algorithm (GA). Figure 2.7 illustrate the basic structure of an evolutionary algorithm.

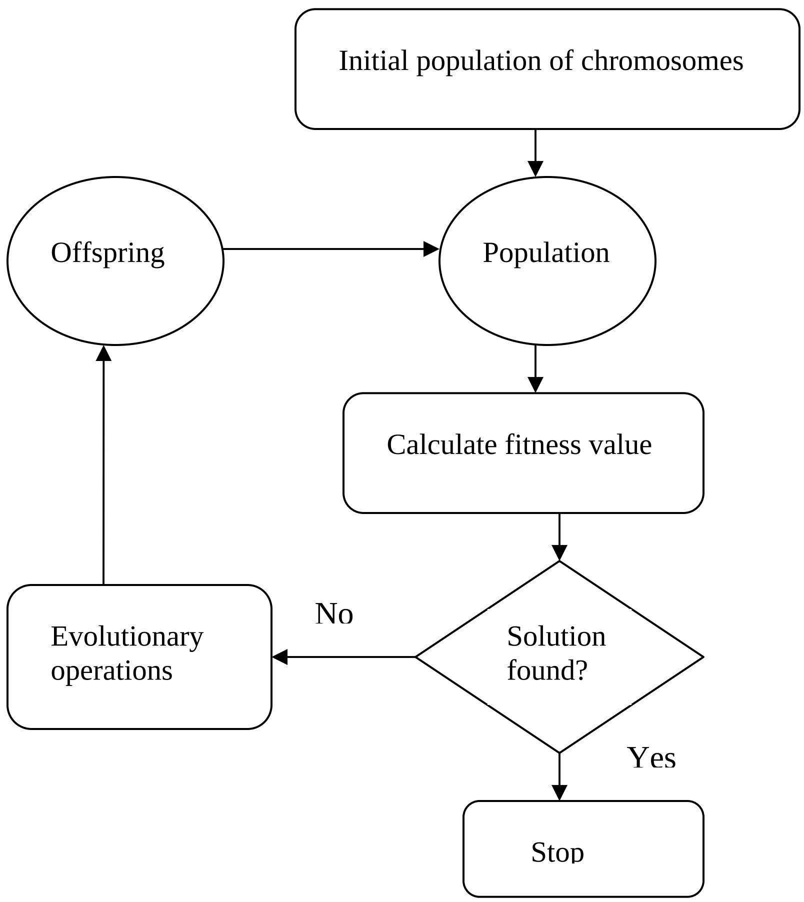


Figure 2.7: Basic structure of an evolutionary algorithm

Genetic Algorithms

Genetic algorithm (GA) is a search heuristic that mimics the process of natural evolution. This heuristic is routinely used to generate useful solutions to optimization and search problems. GA is started with a set of solutions (represented by chromosomes) called population. Solutions from one population are taken and used to form a new population. Solutions which are selected to form new solutions (offspring) are selected according to their fitness - the more suitable they are the more chances they have to reproduce. This is repeated until some condition (for example number of populations or improvement of the best solution) is satisfied (Karray and De Silva, 2004).

Genetic Algorithms Operators

The most important of an evolutionary process pertains to the way a composition of a population changes. In nature, we commonly look at three major forces: *natural selection, mating,* and *mutation.* The equivalents of these forces in artificial evolution are*: selection, crossover,* and *mutation*. Collectively, they form a large group of processes, which act on individuals, set of individuals, populations, and genes, and are known as *genetic operators* (Karray and De Silva, 2004).

- **Selection:** This procedure is applied to select the individuals that participate in the reproduction process to give birth to the next generation. Selection operators usually work on a population, and may serve to remove weaklings, or to select strong individuals for reproduction.
- **Crossover:** Crossover selects genes from parent chromosomes and creates a new offspring. This process involves randomness, and thus most crossover operators randomly select a set of genes from each parent to form a child's genotype.
- **Mutation:** While the crossover operation generates new combinations of genes, and therefore new combinations of traits, mutation can introduce completely new alleles into a population. It has been widely recognized that mutation is the

operator that creates completely new solutions while crossover and selection serve to explore variant of existing solutions while eliminating bad ones.

Basic Steps of Genetic Algorithm

1. **[Start]** Generate random population of *n* chromosomes (suitable solutions for the problem)
2. **[Fitness]** Evaluate the fitness *f(x)* of each chromosome *x* in the population
3. **[New population]** Create a new population by repeating following steps until the new population is complete
   i. **[Selection]** Select two parent chromosomes from a population according to their fitness (the better fitness, the bigger chance to be selected)
   ii. **[Crossover]** With a crossover probability cross over the parents to form a new offspring (children). If no crossover was performed, offspring is an exact copy of parents.
   iii. **[Mutation]** With a mutation probability mutate new offspring at each locus (position in chromosome).
   iv. **[Accepting]** Place new offspring in a new population
4. **[Replace]** Use new generated population for a further run of algorithm
5. **[Test]** If the end condition is satisfied, **stop**, and return the best solution in current population
6. **[Loop]** Go to step **2**

The figure 2.8 below depicts the schematic representation of a genetic algorithm.

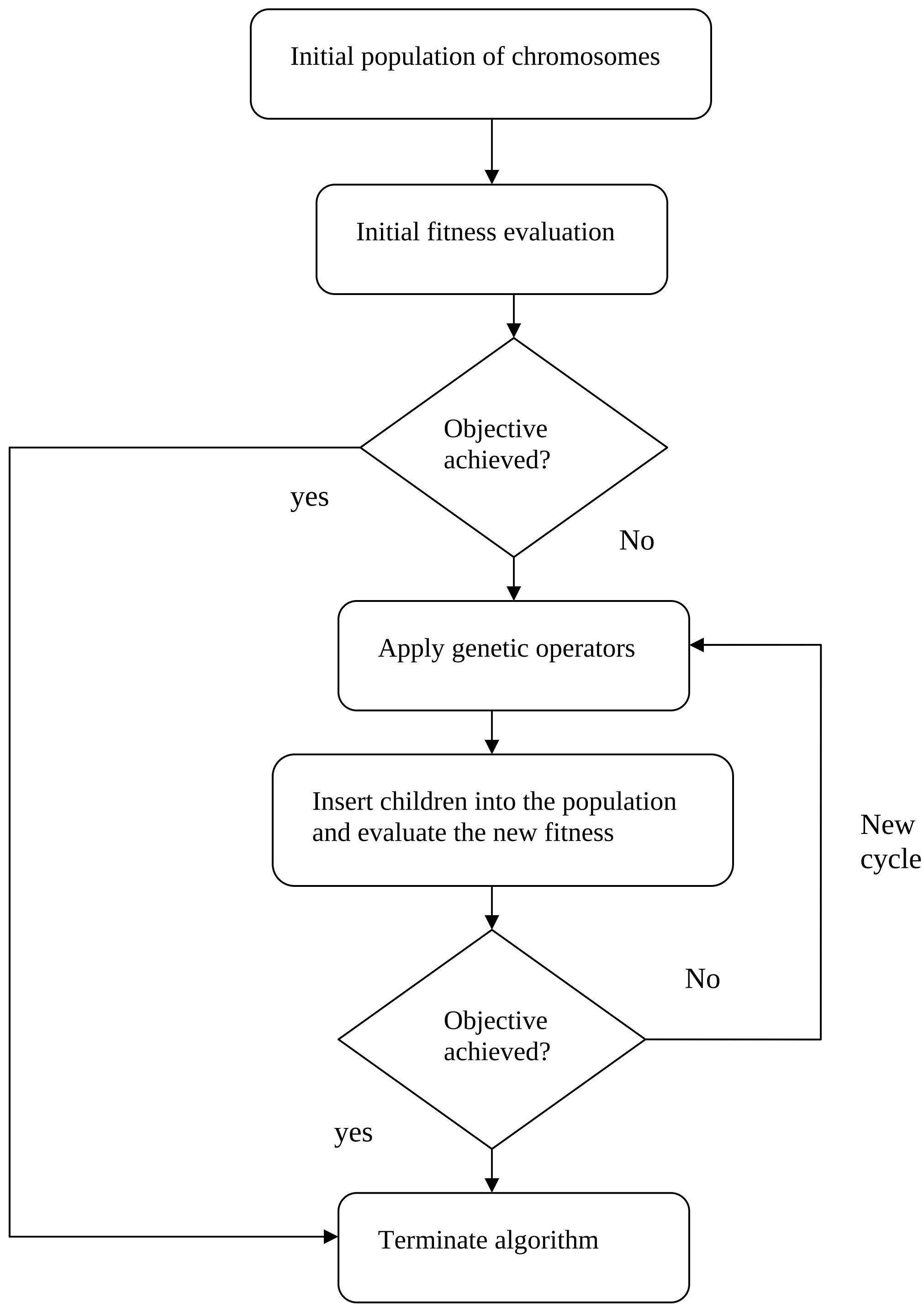


Figure 2.8: Schematic representation of a genetic algorithm (Karray and De Silva, 2004)

2.3 **Statistical Techniques**

The two commonly used domain to forecast time series data are statistical and soft computing domain. The well known SC techniques have already been described in Section 2.2. We describe the most widely used statistical method for time series forecasting in the following paragraphs.

Autoregressive Integrated Moving Average (ARIMA) Model

The ARIMA model is also known as Box-Jenkins model or methodology used in analysis and forecasting. It is widely regarded to be the most efficient forecasting technique, and is used extensively - especially for univariate time series. ARIMA methods for forecasting time series are essentially agnostic. Unlike other methods they do not assume knowledge of any underlying economic model or structural relationships. It is assumed that past values of the series plus previous error terms contain information for the purposes of forecasting.

The main advantage of ARIMA forecasting is that it requires data of the time series in question only. First, this feature is advantageous if one is forecasting a large number of time series. Second, this avoids a problem that occurs sometimes with multivariate models. For example, consider a model including wages, prices and money. It is possible that a consistent money series is only available for a shorter period of time than the other two series, restricting the time period over which the model can be estimated. Third, with multivariate models, timeliness of data can be a problem. If one constructs a large structural model containing variables which are only published with a long lag, such as wage data, then forecasts using this model are conditional forecasts based on forecasts of the unavailable observations, adding an additional source of forecast uncertainty.

Some disadvantages of ARIMA forecasting are that:

- Some of the traditional model identification techniques are subjective and the reliability of the chosen model can depend on the skill and experience of the forecaster (although this criticism often applies to other modelling approaches as well).
- It is not embedded within any underlying theoretical model or structural relationships. The economic significance of the chosen model is therefore not clear. Furthermore, it is not possible to run policy simulations with ARIMA models, unlike with structural models.
- ARIMA models are essentially 'backward looking'. As such, they are generally poor at predicting turning points, unless the turning point represents a return to a long-run equilibrium.

However, ARIMA models have proven themselves to be relatively robust especially when generating short-run forecasts. ARIMA models frequently outperform more sophisticated structural models in terms of short-run forecasting ability (Meyler et al., 1998)

The first three steps in the ARIMA analysis consist of identification, estimation and diagnosis. The first step is *identification* in which autocorrelation functions (ACFs) and partial autocorrelation functions (PACFs) are examined to see which of the potential patterns are present in the data. Also to make the data stationary usually by differencing the data and then analyzing the autocorrelations and partial autocorrelations of the stationary data. Also, when time series is long, there are also tendencies for measures to vary periodically, called seasonality, periodicity, or cyclic in time series data. These patterns are also identified by ACFs and PACFs and accounted for in the model. Time series analysis is more appropriate for data with autocorrelation than, say, multiple regression, for two reasons. The first is that there is explicit violation of the assumption of independence of errors. The errors are correlated due to the pattern over time in the data. Type I error rate is substantially increased if regression is used when there is no autocorrelation. The second is that the patterns may either obscure or spuriously enhance the effect of an intervention unless accounted for in the model. The second step in modeling the series is *estimation* in which the estimated size of a lingering autoregressive or moving average effect is tested against null hypothesis that it is zero. In other words, determining the parameters of the model. This is similar to estimating the parameters in regression analysis.

The third step is *diagnosis*, in which residual scores are examined to determine if there are still patterns in the data that are not accounted for. Residual scores are the differences between the scores predicted by the model and the actual scores for the series. If all patterns are accounted for in the model, the residual are random. In many application of time series, identifying and modeling the patterns in the data is sufficient to produce an equation which is then used to predict the future of the process. This is called *forecasting*, the goal of many applications of time series in the economic and business arena (Tabachnick and Fidell, 2001).

In contrast to other techniques, Box-Jenkins is a procedure which uses a variable's past behavior to select the best forecasting model from a general class of models. It assumes that any time series pattern can be represented by one of three categories of models. These categories include:

• Autoregressive models: forecasts of a variable based on linear function of its past values

• Moving Average models: forecasts based on linear combination of past errors

• Autoregressive Moving Average models: combination of the previous two categories.

Identification of ARIMA (*p, d, q*) Models

The ARIMA (autoregressive, integrated, moving average) model of a time series is defined by three terms (*p, d, q*). Identification of a series is the process of finding integer, usually very small (e.g., 0, 1, or 2), values of *p*, *d*, and *q* that model the patterns in the data. When the value is 0, the element is not needed in the model. The middle element, *d*, is investigated before *p* and *q*. The goal is to determine if the process is stationary and if not, to make it stationary before determining the values of *p* and *q*. A stationary process has a constant mean and variance over the time period of the study (Tabachnick and Fidell, 2001).

Trend Components, *d*: Making the Process Stationary

Differencing the series is the simplest way to make a nonstationary mean stationary. The number of times you have to difference the time series data to make the process stationary determines the value of d. if $d = 0$, the model is already stationary and has no trend. When the series is differenced once, $d = 1$, and linear trend is removed. When the difference is then differenced, d = 2 and both linear and quadratic trend are removed. For nonstationary series, d values of 1 or 2 are usually adequate to make the mean stationary (Tabachnick and Fidell, 2001).

Auto-Regressive (AR) Components

The auto-regressive components represent the memory of the process for preceding observations. The value of p is the number of auto-regressive components in an ARIMA (p, d, q) model. The value of p is 0 if there is no relationship between adjacent observations. When the value of p is 1, there is a relationship between observations at lag 1 and the correlation coefficient ϕ_1 is the magnitude of the relationship. When the value of p is 2, there is a relationship between observations at lag 2 and the correlation coefficient ϕ_2 is the magnitude of the relationship. Thus p is the number of correlations you need to model the relationship. A pth-order autoregressive model; AR(p) has the general form as follows:

$$Y_t = \phi_0 + \phi_1 Y_{t-1} + \phi_2 Y_{t-2} + ... + \phi_p Y_{t-p} + \varepsilon_t \qquad (2.5)$$

where

Y_t = Response (dependent) variable at time t

Y_{t-1}, Y_{t-2}, . . ., Y_{t-p} = Response variable at time lags $t-1$, $t-2$, . . ., $t - p$, respectively

$\phi_0, \phi_1, \phi_2, ..., \phi_p$.= Coefficients to be estimated.

ε_t = Error term at time t

Moving Average (MA) Components

The moving average components represent the memory of the process for preceding random shocks. The value q indicates the number of moving average components in the

ARIMA(*p, d, q*). When q is 0, there are no moving average components. When q is 1, there is a relationship between the current score and the random shock at lag 1 and the correlation coefficient θ_1 represents the magnitude of the relationship. When q is 2, there is a relationship between the current score and the random shock at lag 2, and the correlation coefficient θ_2 represents the magnitude of the relationship. A qth-order moving average has the general form as follows:

$$Y_t = \mu + \varepsilon_t - \theta_1 \varepsilon_{t-1} - \theta_2 \varepsilon_{t-2} - \ldots - \theta_q \varepsilon_{t-q} \tag{2.6}$$

where

Y_t = Response (dependent) variable at time t

μ = Constant mean of the process

$\theta_1, \theta_2, \ldots, \theta_q$ = Coefficients to be estimated

ε_t = Error term at time t

$\varepsilon_{t-1}, \varepsilon_{t-2}, \ldots, \varepsilon_{t-p}$ = Errors in previous time periods that are incorporated in the response $Y_{t.}$

Autoregressive Moving Average (ARMA) Model

The autoregressive and moving average specifications can be combined to form an ARMA (*p, q*) specification. ARMA (*p, q*) has the general form as follows (Tabachnick and Fidell, 2001).

$$Y_t = \phi_0 + \phi_1 Y_{t-1} + \phi_2 Y_{t-2} + \ldots + \phi_p Y_{t-p} + \varepsilon_t - \theta_1 \varepsilon_{t-1} - \theta_2 \varepsilon_{t-2} - \ldots - \theta_q \varepsilon_{t-q} \tag{2.7}$$

Autocorrelation Functions (ACFs) and Partial Autocorrelation Functions (PACFs)

Models are identified through patterns in their autocorrelation functions **(**ACFs) and partial autocorrelation functions **(**PACFs). Both autocorrelations and partial autocorrelation are computed for sequential lags in the series. The first lag has an autocorrelation between Y_{t-1} and Y_t, the second lag has both an autocorrelation and partial autocorrelation between Y_{t-2} and Y_t, and so on. ACFs and PACFs are the functions across all the lags. The equation for autocorrelation is as follows (Tabachnick and Fidell, 2001).

$$r_k = \frac{\frac{1}{N-k}\sum_{t=1}^{N-k}(Y_T - \overline{Y})(Y_{t-k} - \overline{Y})}{\frac{1}{N-1}\sum_{t=1}^{N}(Y_t - \vec{Y})^2} \tag{2.8}$$

where, N is the number of observations in the whole series, k is the lag. $\overline{Y}$ is the mean of the whole series and the denominator is the variance of the whole series.

The standard error of an autocorrelation is based on the squared autocorrelations from all previous lags. At lag 1, there are no previous autocorrelations, so r_0^2 is set to 0.

$$SE_{rk} = \sqrt{\frac{1 + 2\sum_{l=0}^{k-1} r_l^2}{N}} \tag{2.9}$$

However, the standard error for a partial autocorrelation is simple and the same at all lags:

$$SE_{pr} = \frac{1}{\sqrt{N}} \tag{2.10}$$

ARIMA FORECASTING IN PRACTICE

The figure 2.9 graphically illustrates the ARIMA forecasting procedure. However, this process is not a simple sequential one, but can involve iterative loops depending on results obtained at the diagnostic and forecasting stages. The first step is to collect and examine graphically and statistically the data to be forecast. The second step is to test whether the data are stationary or if differencing is required. Once the data are rendered stationary one should seek to identify and estimate the correct ARMA model. Two alternative approaches to model identification are considered - the Box-Jenkins methodology and penalty function criteria.

It is important that any identified model be subjected to a battery of diagnostic checks (usually based on checking the residuals) and sensitivity analysis. For example, the estimated parameters should be relatively robust with respect to the time frame chosen. Should the diagnostic checks indicate problems with the identified model one should return to the model identification stage. Once a model or selection of models has been chosen, the models should then be used to forecast the time series, preferably using out-of-sample data to evaluate the forecasting performance of the model.

One common pitfall of ARIMA modelling is to overfit the model at the identification stage, which maximises the in-sample explanatory performance of the model but may lead to poor out-of-sample predictive power relative to a more parsimonious model. Thus, if a model with a large number of AR and MA lags yields poor forecasting performance, it may be optimal to return to the model identification stage and consider a more parsimonious model (Shaikh, 2004).

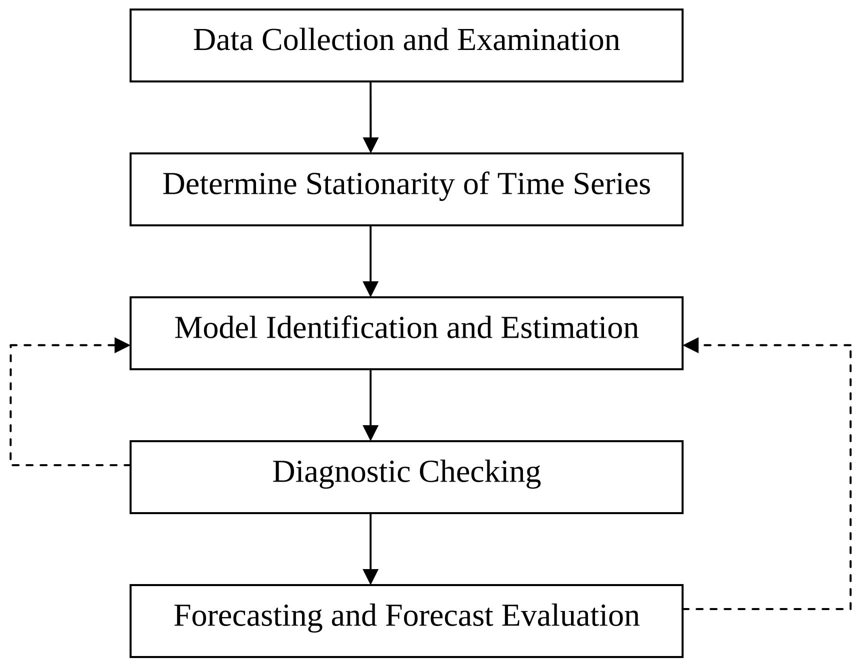


Figure 2.9: ARIMA Forecasting Procedure

2.4 Related Works of the Application of ANN in Stock Price Prediction

A review of previous studies on stock price forecasting shows that the use of artificial neural networks (ANNs) is the most commonly used non-linear technique for Stock market prediction over the last two decades. What follows are reviews of literatures of related works of the application of ANNs to stock price prediction.

With reference to Avci (2007); White (1988) published the first significant study on the application of the neural network models for stock market forecasting. Following White's study, several research efforts were carried out to examine the forecasting effectiveness of the neural network models in stock markets. Among the earlier studies, Kimoto, *et al*. (1990), Kamijo and Tanikawa (1990) can be mentioned. However, in another contribution, Yoda (1994) investigated the predictive capacity of the neural network models for the Tokyo Stock Exchange. Wong et al. (1992) used the neural network models to forecast various US stock returns, while Kryzanowski et al. (1993) used the neural network models to select the stock from the Canadian companies. Furthermore, Jang and Lai (1994) investigated the effectiveness of neural network models in an emerging country case by an application to Taiwan stock exchange weighted price index.

Tsang et al., (2007) in their article stated that the study of stock prediction can be broadly divided into two schools of thought. One focuses on computer experiments in virtual/artificial markets. This is often the case when researchers model the complex movements in the market economics. The other school focuses on stock prediction based on real-life financial data. Other papers on the stock market forecasting applications of neural network models examined the neural network applications from different aspects.

With reference to Avci (2007), some studies considered the effects of modeling preferences on one type of neural network models. Other studies examined variation of effects of; architecture (Brownstone, 1996), training algorithms (Sun *et al*., 2005), and input variables (Kohara *et al*., 1997; Phua *et al*., 2001; Stansel and Eakins, 2004; and Lam, 2004) on neural network models' forecast performances. On the other hand, some other studies were devoted to investigating the forecast performance differences among different neural network models (Kim and Chung, 1998; Saad, *et al*., 1998). Other than the modeling issues, several studies evaluated the profitability of neural network models in stock markets. Among these studies, Perez-Rodriguez *et al*. (2005), reported that the technical trading strategy guided by feedforward neural network model was superior to buy-and-hold strategy. In contrast, Chandra and Reeb (1999) found out that the neural network models produce significantly lower returns than the buy and hold strategy.

Besides the studies that compare the neural network models with buy-and-hold strategy, another group of the studies dealt exclusively with comparing the forecast performance of neural network models with other linear and/or non-linear statistical models. Although, some contrary views exist in literature on the superiority of neural network models on alternative forecasting techniques, the neural network models exhibit promising results for future studies (Avci, 2007).

With reference to Kyoung-jae and Lee (2004) some researchers investigated the issue of predicting the stock index futures market. Choi et al., (1995) and Trippi and DeSieno (1992) predicted the daily direction of change in the S&P 500 index futures using ANN. Trippi and DeSieno (1992) combined the outputs of individual networks using logical (Boolean) operators to produce a set of composite rules. They suggested that their best composite synthesized rule set system achieved a higher gain than previous research. Choi et al. (1995) compared their approach with previous study and suggested that they earned a higher annualized gain than the previous study. However, the annualized gain may not be an appropriate measure for prediction performance because it varies according to the fee for trade and the trading strategy. Duke and Long (1993) predicted German government daily bond futures using backpropagation (BP) neural networks. They reported that the 53.94% of the patterns are accurately predicted through the moving simulation method. Most of the above studies simply applied ANN to stock market prediction

Phua et al., (2003) presents a study of using artificial neural networks in predicting stock index increments. The data of five major stock indices, DAX, DJIA, FTSE-100, HSI and NASDAQ, are applied to test their network model. Computational results from five different financial markets show that the trust region based neural network model obtained better results compared with the results obtained by other neural network models. In particular, their findings showed that the model was able to forecast the sign of the index increments with an average success rate above 60% in all the five stock markets and the best prediction result in their application reaches the accuracy rate of 74%. Chen et al., (2003) in their study, attempted to model and predict the direction of return on market index of the Taiwan Stock Exchange, one of the fastest growing financial exchanges in developing Asian countries. Their motivation is based on the notion that trading strategies guided by forecasts of the direction of price movement may be more effective and lead to higher profits. The probabilistic neural network (PNN) was used to forecast the direction of index return after it is trained by historical data. Empirical results show that the PNN-based investment strategies obtain higher returns than other investment strategies examined in this study.

Kunhuang and Yu (2006) used backpropagation neural network because of its nonlinear structures to forecast fuzzy time series, the study findings showed that that ANN has better forecast ability than time series model. Zhu et al., (2007) also used technical indicators with ANN and their findings revealed that ANN can forecast stock index increment and trading volume will lead to modest improvements in stock index performance. Tsanga et al., (2007) used ANN to created trading alert system and their findings showed that ANN can effectively guide investors when to buy or sell stocks. Avci (2007) also used ANN to forecasting daily and sessional returns of the Ise-100 Index and his finding demonstrated that ANN can be used effectively to forecast daily and sessional returns of the Ise-100 Index.

Other works that had applied ANN models with technical indexes to stock price predictions with varying findings (Kimoto et al., 1990; Kamijo and Tanigawa, 1990; Yao and Poh, 1995; Aiken and Bsat 1999; Sfetsos 2002; Kyoung-Jae and Lee 2004; Stansel and Eakins, 2004; Chen et al., 2005; Huang et al., 2005; Lipinsk 2005; De Leone et al., 2006; Roh, 2007; Giordano et al., 2007; Jain and Kumar, 2007; Kyungjoo et al., 2007; Al-Qahari 2008; Bruce and Garin 2009; Mitra 2009, George and Kimon, 2009; Mohammed 2010; Esmaeil et al. 2010; Tiffany and Kun-Huang, 2010).

Although encouraging results have been reported in which ANN-based systems outperformed widely used, well-established statistical method. Many inconsistent reports have been undermining the robustness of these findings. Among the reasons of these discrepancies are well known problems that characterize ANNs, in particular (1) different network type such as multilayer perceptrons, radial basis functions networks, or RBF and self organizing maps, among others can lead to different results when trained and tested on the same database. This is mainly due to the different classes of decision boundaries that different ANN types prefer; (2) for a given network type, ANNs are sensitive to the choice of topology and size for a given data set; (3) ANN are prone to over fitting, unless great care is taken in choosing the size and connectivity of the network (Massimiliano et al., 2004).

Recent research tends to hybridize several AI techniques with the intention to improve the forecasting accuracy, the combination of forecasting approaches has been proposed by many researchers (Nikolopoulos and Fellrath, 1994; Hiemstra, 1995; Kohara et al., 1997; Chi, 1998, Massimiliano et al., 2004; Raymond 2004; Kyoung-jae 2006; Rohit and Kumkum 2008, Khashei et al., 2008). From their studies, they indicated that the integrated forecasting techniques outperformed the individual forecast. Nikolopoulos and Fellrath (1994) developed a hybrid expert system for investment advising. In their study, genetic algorithms were used to train and configure the architecture of investor's neural network component. Hiemstra (1995) proposed fuzzy expert systems to predict stock market returns. He suggested that ANN and fuzzy logic could capture the complexities of functional mapping because they do not require the specification of the function to approximate. Some researchers tend to include novel factors for the learning process. Kohara et al., (1997) incorporated prior knowledge to improve the performance of stock market prediction. Prior knowledge in their study included non-numerical factors such as political and international events. They made use of prior knowledge of stock price predictions and newspaper information on domestic and foreign events.

Kyoung-jae (2006) proposed a genetic algorithm (GA) approach to instance selection in artificial neural networks (ANNs) for financial data mining. ANN has preeminent learning ability, but often exhibit inconsistent and unpredictable performance for noisy data. In addition, it may not be possible to train ANN or the training task cannot be effectively carried out without data reduction when the amount of data is so large. The GA optimizes simultaneously the connection weights between layers and a selection task for relevant instances. The globally evolved weights mitigate the well-known limitations of gradient descent algorithm. In addition, genetically selected instances shorten the learning time and enhance prediction performance. The study applied the proposed model to stock market analysis. Experimental results show that the GA approach is a promising method for instance selection in ANN.

Rohit and Kumkum (2008) proposed a hybrid machine learning system based on Genetic Algorithm (GA) and Support Vector Machines (SVM) for stock market prediction. A variety of indicators from the technical analysis field of study are used as input features. Also make use of the correlation between stock prices of different companies to forecast the price of a stock, making use of technical indicators of highly correlated stocks, not only the stock to be predicted. The genetic algorithm is used to select the set of most informative input features from among all the technical indicators. The results show that the hybrid GA-SVM system outperforms the stand alone SVM system. Valenzuela et al.

(2008) proposed a new procedure to predict time series using paradigms such as fuzzy systems, neural networks and evolutionary algorithms. Their goal is to obtain an expert system based on paradigms of artificial intelligence, so that the linear model can be identified automatically, without the need of human expert participation. The obtained linear model will be combined with ANN, making up an hybrid system that could outperform the forecasting results. In the work of Khashei et al. (2008), ARIMA models are integrated with ANN and Fuzzy logic in order to overcome the linear and data limitations of ARIMA models, thus obtaining more accurate results.

Khan et al., (2008) compared the performance of backpropagation neural network and genetic-based backpropagation neural network to predict the daily stock price. The results of their study showed that the genetic algorithm based backpropagation neural network predict stock price more accurately as compared to backpropagation neural network. Tseng et al., (2006) investigated whether a hybrid approach combining different stock prediction approaches together can dramatically outperform the single approach. The experiment showed that by combining the single algorithm considerately, a better performance can be received. Wang (2007) proposed hybrid ANN (ANN and GARCH) with technical indicators and his findings showed that ANN combined with other techniques has good forecast ability as good as ARIMA model. Kim and Shin (2007) used hybrid ANN (ANN and Genetic Algorithm) with technical indicators and their findings showed that hybrid model has better forecast ability than single model and ANN has ability to forecast stock market. Yan (2008) used hybrid ANN (ANN and Grey Theory) with technical indicators and his findings showed that hybrid model has better forecasting performance to stock price prediction. Khashei et al., (2009) used hybrid ANN (ANN and Fuzzy Logic) with technical indicators and their finding revealed that hybrid models exhibit effectively improved forecasting accuracy of stock price prediction.

Other research works that demonstrated prevalent use of technical indicators (such as opening price, highest price, lowest price, closing price and trading volume) to forecast stock prices (Zhongxing and Liting 1993; Wang and Leu 1996; Nishina and Hagiwara 1997; Pantazopoulos et al., 1998; Hui et al., 2000; Barnes et al., 2000; Leigh et al 2000; Ayob et al., 2001; Bautista 2001; Phua et al., 2001; Zhang et al., 2002; Rech 2002; Ajith et al. 2003; Perez-Rodriguez et al., 2004; Zhang et al., 2004; Chen et al., 2005; Chun and Park 2005; Doesken et al., 2005, Pan et al., 2005; Pai and Lin 2005; Atsalakis and Valavanis 2006).

However, O'Connor and Maddem (2006) used fundamental indicators with ANN and their findings revealed that ANN has forecast ability in stock market because it has better return than overall stock market. Other research works that engaged the use of fundamental indicators to forecast stock prices (Atiya et al 1997; Quah and Srinivasan 1999; Raposo et al., 2002).

From the above literature review, technical indicators with ANN model had been widely used, while there are only few cases of the use of fundamental indicators. This research work contrasts previous approaches by combining technical indicators, fundamental indicators and experts' opinion to improve stock price prediction using ANN model. The technical analysis variables are the core stock market indices such as current stock price, opening price, closing price, volume, highest price and lowest price etc. Fundamental indicators are the company performance indices such as price per annual earning, return on asset, return on common equity, book value, financial status of the company, etc. while the experts opinion are other influential factors such as interest rate, inflation rate, foreign exchange rate, business sector, management caliber, investors confidence, government policy and political factors among others. Hence, the novelty of our approach in this research work stems from the use of hybrid parameters for improving stock market prediction.

CHAPTER THREE

DESIGN AND DEVELOPMENT OF THE STOCK PRICE PREDICTIVE MODEL

3.1 INTRODUCTION

This chapter presents in detail the approach engaged in developing stock price predictive models in line with the research questions and objectives of the study. The procedures and processes of the two forecasting techniques used in this study are explicitly explained. The forecasting techniques are statistical technique and soft computing technique. The statistical technique employed in this regard is the Auto Regressive Integrated Moving Average (ARIMA) model. It is widely regarded as the most efficient forecasting technique, and is used extensively especially for time series forecasting. The soft computing used in this study is artificial neural networks (ANNs). ANN tools are noted in literature to have increasingly gained popularity due to their inherent capabilities to approximate any nonlinear function to a high degree of accuracy. ANNs are less sensitive to error term assumptions and they can tolerate noise, and chaotic components.

The historical stock data of four different companies that were used in this study were sourced from the London Stock Exchange (LSE) and the Nigeria Stock Exchange (NSE) respectively. The two companies' stock data from LSE are Dell Corporation and Nokia Corporation relatively from information technology industry sector. The Companies from NSE are from banking industry sector which are UBA bank and Zenith bank. The performance measure/evaluation of the models developed was done with confusion matrix, root mean square error (RMSE), mean square error (MSE) and mean absolute percentage error (MAPE) in order to determine the accuracy of the output results of each model developed. The tools for implementation are also discussed which are Matlab and Eviews software.

3.2 THE STEPS IN BUILDING ARIMA MODEL

3.2.1 Step One - Data Collection and Examination

A lengthy time series of data is required for univariate time series forecasting. It is usually recommended that at least 50 observations be available. Using either Box-Jenkins or objective penalty function methods can be problematic if too few observations are available. Unfortunately, even if a long time series is available, it is possible that the series contains a structural break which may necessitate only examining a sub-section of the entire data series, or alternatively using intervention analysis or dummy variables. Thus, there may be some conflict between the need for sufficient degrees of freedom for statistical robustness and having a shorter data sample to avoid structural breaks.

Graphically examining the data is important. They should be examined in levels, logs, differences and seasonal differences. The series should be plotted against time to assess whether any structural breaks, outliers or data errors occur. If so, one may need to consider use of intervention or dummy variables. This step may also reveal whether there is a significant seasonal pattern in the time series.

Another way to examine the properties of a time series is to plot its autocorrelogram. The autocorrelogram plots the autocorrelation between differing lag lengths of the time series.

Plotting the autocorrelogram is a useful aid for determining the stationarity of a time series, and is also an important input into Box-Jenkins model identification. If a time series is stationary then its autocorrelogram should decay quite rapidly from its initial value of unity at zero lag. If the time series is nonstationary then the autocorrelogram will only die out gradually over time (Meyler et al., 1998).

3.2.2 **Step Two - Testing for Stationarity**

The time series under consideration must be stationary before one can attempt to identify a suitable ARMA model. Determine whether the series is stationary or not by considering the graph of ACF. If a graph of ACF of the time series values either cuts off fairly quickly or dies down fairly quickly, then the time series values should be considered stationary. If a graph of ACF dies down extremely slowly, then the time series values should be considered non-stationary. If the series is not stationary, it can often be converted to a stationary series by differencing. That is, the original series is replaced by a series of differences. An ARMA model is then specified for the differenced series. Differencing is done until a plot of the data indicates the series varies about a fixed level, and the graph of ACF either cuts off fairly quickly or dies down fairly quickly.

3.2.3 **Step Three - Model Identification and Estimation**

Having determined the correct order of differencing required to render the series stationary, the next step is to find an appropriate ARMA form to model the stationary series. There are two main approaches to identification of ARMA models in the literature. The traditional method utilizes the Box-Jenkins procedure, in which an iterative process of model identification, model estimation and model evaluation is followed. The Box-Jenkins procedure is a quasi-formal approach with model identification relying on subjective assessment of plots of autocorrelograms and partial autocorrelograms of the series. Objective measures of model suitability, in particular the penalty function criteria, have been used by some authors instead of the traditional Box-Jenkins procedure. For example, the use of objective penalty function criteria (Gómez and Maravall, 1998). However, these objective measures are not without problems either.

3.2.3.1 **Box-Jenkins Methodology**

The Box-Jenkins methodology essentially involves examining plots of the sample autocorrelogram, partial autocorrelogram and inverse autocorrelogram and inferring from patterns observed in these functions the correct form of ARMA model to select. The Box-Jenkins methodology is not only about model identification but is, in fact, an iterative approach incorporating model estimation and diagnostic checking in addition to model identification. Theoretically Box-Jenkins model identification is relatively easy if one has a pure AR or a pure MA process. However, in the case of mixed ARMA models (especially of high order) it can be difficult to interpret sample ACFs and PACFs, and Box-Jenkins identification becomes a highly subjective exercise depending on the skill and experience of the forecaster. Random noise in time series, especially price data, makes Box-Jenkins model identification even more problematic.

3.2.3.2 **Objective Model Identification**

Because of the highly subjective nature of the Box-Jenkins methodology, time series analysts have sought alternative objective methods for identifying ARMA models. Penalty function statistics, such as Akaike Information Criterion (AIC) or Final Prediction Error (FPE) Criterion, Schwarz Criterion (SC) or Bayesian Information Criterion (BIC) and Hannan Quinn Criterion (HQC), have been used to assist time series analysts in reconciling the need to minimise errors with the conflicting desire for model parsimony. These statistics take the form of minimising the sum of the residual sum of squares plus a penalty term which incorporates the number of estimated parameter coefficients to factor in model parsimony (Meyler et al., 1998). These statistics take the form

$$BIC = \log\left(\frac{rss}{n}\right) + \left(\log(n) * \frac{k}{n}\right), \qquad (3.1)$$

$$HQC = \log\left(\frac{rss}{n}\right) + \left(2 * \log(\log(n)) * \frac{k}{n}\right), and \qquad (3.2)$$

$$AIC = \log\left(\frac{rss}{n}\right) + \left(2 * \frac{k}{n}\right) \qquad (3.3)$$

where,

k = number of coefficients estimated (1 + p + q + P + Q)

rss = residual sum of squares

n = number of observations.

Assuming there is a true ARMA model for the time series, the BIC and HQC have the best theoretical properties. The BIC is strongly consistent whereas AIC will usually result in an overparameterised model; that is a model with too many AR or MA terms (Mills

1993). Indeed, it is easy to verify that for n greater than seven the BIC imposes a greater penalty for additional parameters than does the AIC. Gómez and Maravall (1998) also favour the BIC over the AIC.

Thus, in practice, using the objective model selection criteria involves estimating a range of models and the one with the lowest information criterion is selected. This can create a number of difficulties. First, it can be computationally expensive using the penalty function criterion. Estimating all possible models encompassed by a (3,0,3)(2,0,2) model involves estimating 144 different models. Therefore the choice of maximum order is very important to avoid expensive computational requirements. Second, the different objective model selection criteria can suggest different models. That is the ranking order based on the BIC will usually not be the same as under the AIC. Third, even if one utilizes only one measure (e.g., BIC), the difference between the BIC statistics for different models is sometimes only marginal. In summary, the main advantages and disadvantages of objective penalty function criteria are as follows (Meyler et al., 1998):

Advantages of Objective Penalty Function Criteria

- Objective measure with no subjective interpretation.
- Results are readily reproducible and verifiable.
- BIC and HQC are asymptotically consistent.

Disadvantages of Objective Penalty Function Criteria

- There is need to calculate a wide range of models. This can be computationally expensive.

- There are no theoretical guidelines for choosing the maximum order of ARIMA model to consider.
- Sometimes there is little to chose between competing models.

3.2.4 **Step Four - Model Diagnostics Checking**

The fourth step will be the formal assessment of each of the time series models. In this step, the model must be checked for adequacy by considering the properties of the residuals whether the residuals from an ARIMA model must have normal distribution and should be random. An overall check of model adequacy is provided by the Ljung-Box Q statistic. The test statistic Q is:

$$Q_m = n(n+2)\sum_{k=1}^{m}\frac{r_k^2(e)}{n-k} \approx x_{m-r}^2\,; \qquad (3.4)$$

where

$r_k(e)$ = the residual autocorrelation at lag k

n = the number of residuals

m = the number of time lags includes in the test

If the *p-value* associated with the Q statistic is small (*p-value* $<\alpha$), the model is considered inadequate. The analyst should consider a new or modified model and continue the analysis until a satisfactory model has been determined.

3.2.5 **Step Five - Forecasting and Forecast Evaluation**

Forecasts for one period or several periods into the future with the parameters of best model selected, using ARIMA models for forecasting is relatively straightforward. For example, consider a non-seasonal (1, 0, 1) model. The estimated model is given by

$$Y_t = \phi_1 Y_t + \varepsilon_t + \theta_1 \varepsilon_t \tag{3.5}$$

Then the forecast value one period ahead on all information up to time, *t*, is simply given by

$$Y^F_{t+1/t} = \phi_1 Y_t + \theta_1 \varepsilon_t \tag{3.6}$$

as $E_t(\varepsilon_{t+1})$ equal to zero $\forall i > 0$

Similarly,

$$Y^F_{t+2/t} = \phi_1 Y^F_{t+1/t} \tag{3.7}$$

as $E_t(\varepsilon_{t+1})$ and $E_t(\varepsilon_{t+2})$ equal to zero and replace $Y^F_{t+1/t}$ with the value given in equation 3.7 above

$$Y^F_{t+3/t} = \phi_1 Y^F_{t+2/t} \tag{3.8}$$

and so on.

The forecast evaluation can be achieved by the statistics such as mean error (ME), mean absolute error (MAE), mean squared error (MSE) and root mean squared error (RMSE). Denoting the forecast error as $\varepsilon_t = Y_t - Y^F_t$ (i.e., the difference between the actual value of the series and the forecast value), then

$$ME = \frac{1}{F}\sum_{i=1}^{F} \varepsilon_t \tag{3.9}$$

$$MAE = \frac{1}{F}\sum_{i=1}^{F} |\varepsilon_t| \tag{3.10}$$

$$MSE = \frac{1}{F}\sum_{i=1}^{F} (\varepsilon_t)^2 \tag{3.11}$$

$$RMSE = \sqrt{\frac{1}{F}\sum_{i=1}^{F} (\varepsilon_t)}^{\,2} \tag{3.12}$$

where,

F equals the number of out-of-sample observations retained for forecast evaluation allowing for the forecast step (Meyler et al., 1998).

3.3 MODEL DEVELOPMENT AND FORECASTING- ARIMA MODEL

As earlier mentioned, the data used in this research work were historical daily stock prices. The stock data consists of four different prices of the index, namely: open price, low price, high price and close price respectively. The open price is the opening price of the index at the start of a trading day, the low price represents the minimum price of the index during the trading day, the high price represents the maximum price of the index during the trading day, and closing price indicates the price of the index when the market closes. In this research the closing price is chosen to represent the price of the index to be modeled and predicted. The choice is because the closing price reflects all the activities of the index in a day. By following the steps in the ARIMA model-building, also illustrated in figure 3.1, we can obtain the following results in the subsection below:

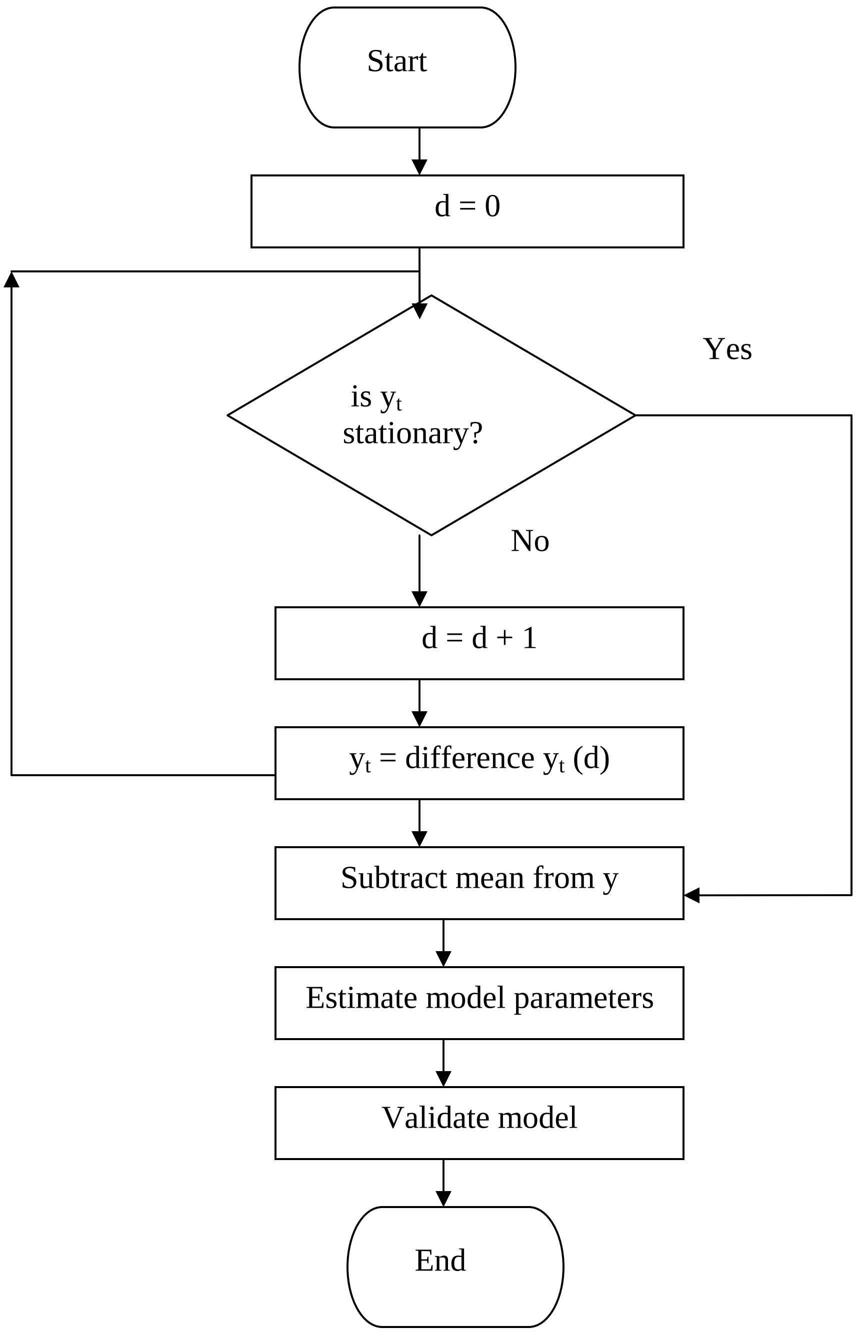


Figure 3.1: Flowchart for building ARIMA model

3.3.1 ARIMA (*p, d, q*) Model Development for Stock Price of Dell Incorporation

The Dell Inc. stock data used in this study covers the period from 17th August, 1988 to 25th February, 2011 having a total number of 5680 observations. Figure 3.2 depicts the original pattern of the time series of the index in order to get a general idea whether the time series is stationary or not. From the graph below the time series have random walk pattern. The series varies randomly and there is no global trend or seasonality pattern observed.

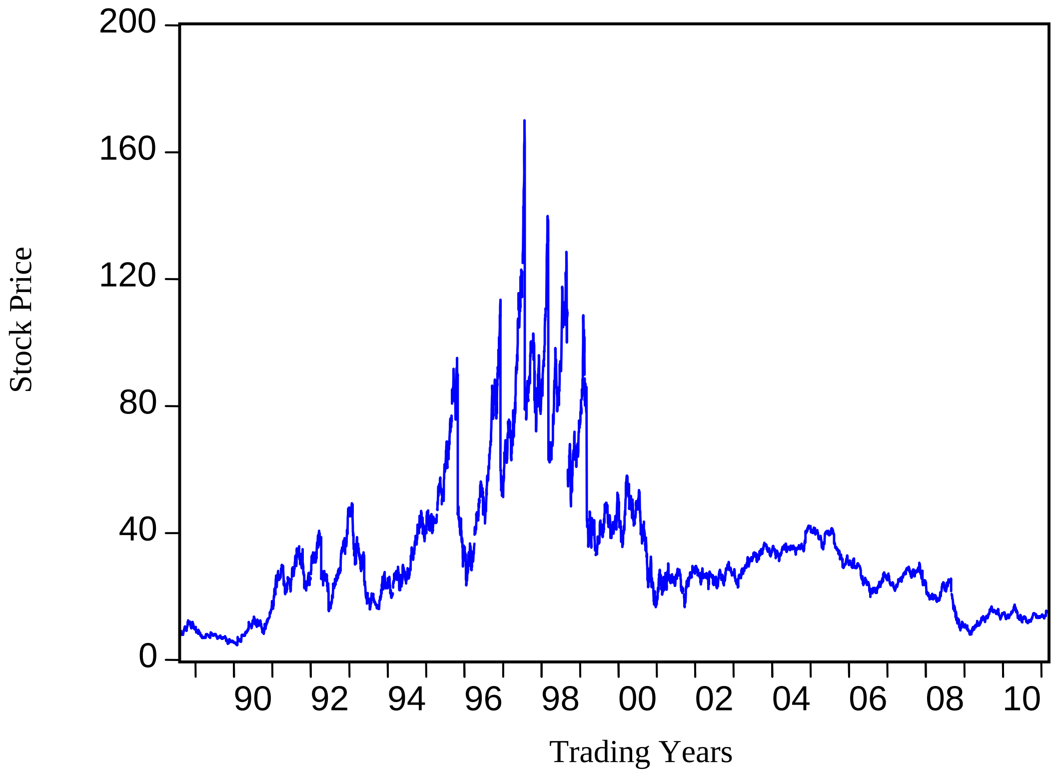
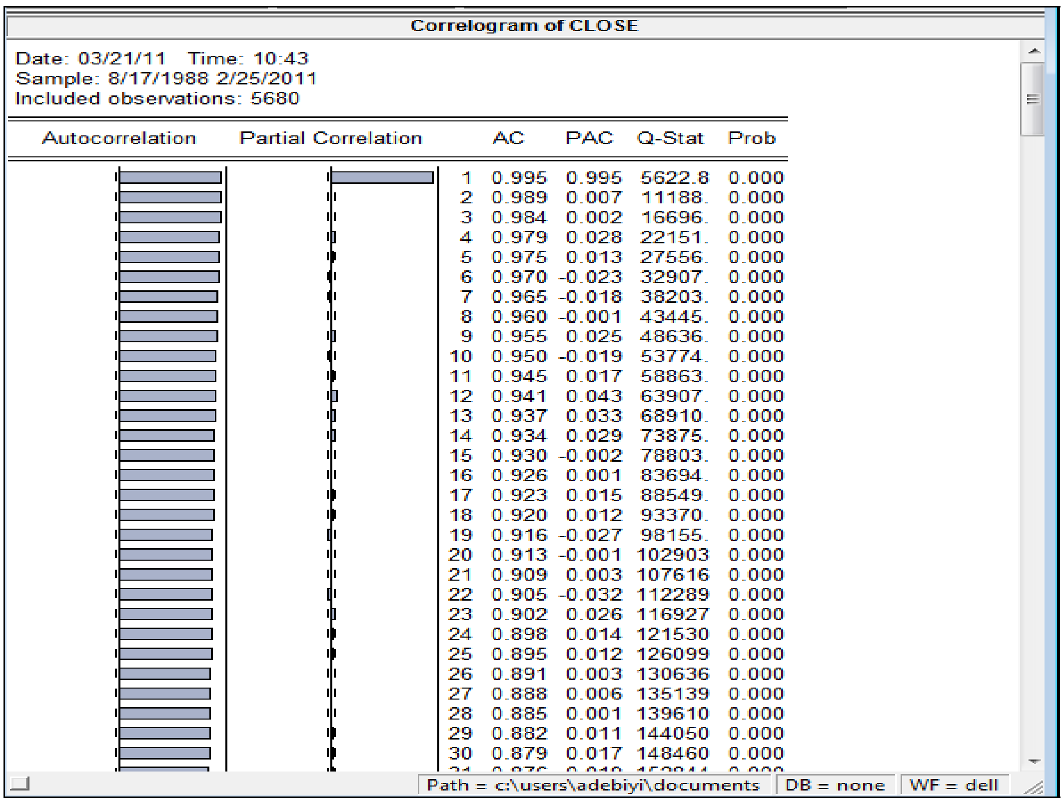


Figure 3.2: Graphical representation of the Dell stock closing price index

With a correlogram we can determine whether a particular series is stationary or nonstationary. If a time series is stationary then its autocorrelation function (ACF) should decay quite rapidly from its initial value of unity at zero lag. If the time series is nonstationary then the ACF will only die out gradually over time. Figure 3.3 is the correlogram of the time series of Dell stock index. From the graph of the correlogram, the autocorrelation function dies down extremely slowly which simply means that the time series is nonstationary. Differencing the time series is the easiest way to make a nonstationary mean stationary. The number of times you have to difference the time series to make the process stationary determines the value of difference (*d*). Figure 3.4 and figure 3.5 is the line graph and correlogram of the series after first differencing, where it shows the correlogram up to 30 lags of the series.


Correlogram of CLOSE

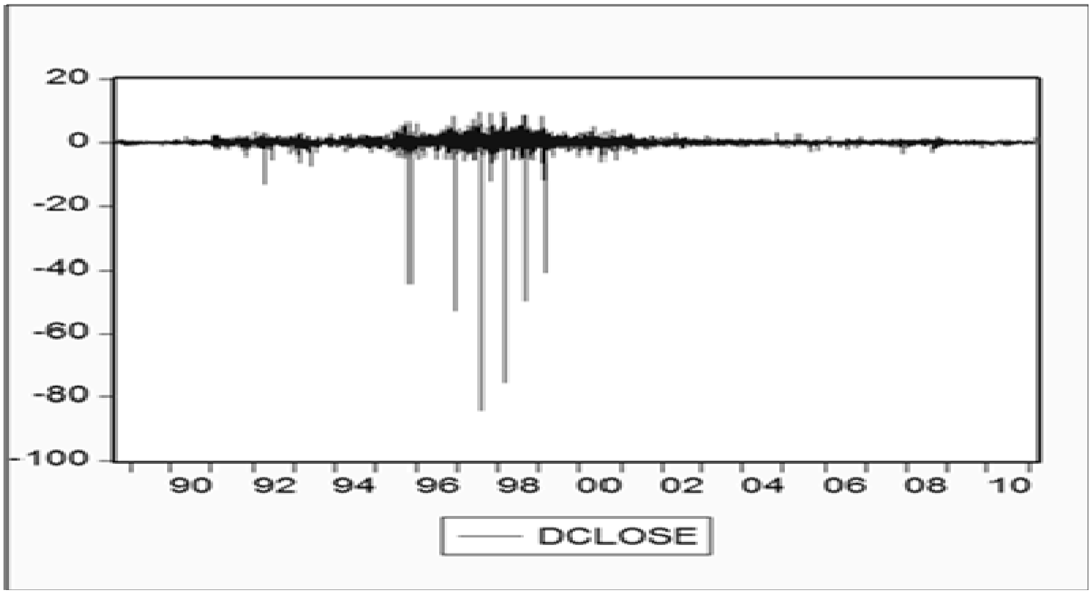
Date: 03/21/11 Time: 10:43
Sample: 8/17/1988 2/25/2011
Included observations: 5680

Autocorrelation	Partial Correlation		AC	PAC	Q-Stat	Prob
		1	0.995	0.995	5622.8	0.000
		2	0.989	0.007	11188.	0.000
		3	0.984	0.002	16696.	0.000
		4	0.979	0.028	22151.	0.000
		5	0.975	0.013	27556.	0.000
		6	0.970	-0.023	32907.	0.000
		7	0.965	-0.018	38203.	0.000
		8	0.960	-0.001	43445.	0.000
		9	0.955	0.025	48636.	0.000
		10	0.950	-0.019	53774.	0.000
		11	0.945	0.017	58863.	0.000
		12	0.941	0.043	63907.	0.000
		13	0.937	0.033	68910.	0.000
		14	0.934	0.029	73875.	0.000
		15	0.930	-0.002	78803.	0.000
		16	0.926	0.001	83694.	0.000
		17	0.923	0.015	88549.	0.000
		18	0.920	0.012	93370.	0.000
		19	0.916	-0.027	98155.	0.000
		20	0.913	-0.001	102903	0.000
		21	0.909	0.003	107616	0.000
		22	0.905	-0.032	112289	0.000
		23	0.902	0.026	116927	0.000
		24	0.898	0.014	121530	0.000
		25	0.895	0.012	126099	0.000
		26	0.891	0.003	130636	0.000
		27	0.888	0.006	135139	0.000
		28	0.885	0.001	139610	0.000
		29	0.882	0.011	144050	0.000
		30	0.879	0.017	148460	0.000

Path = c:\users\adebiyi\documents | DB = none | WF = dell

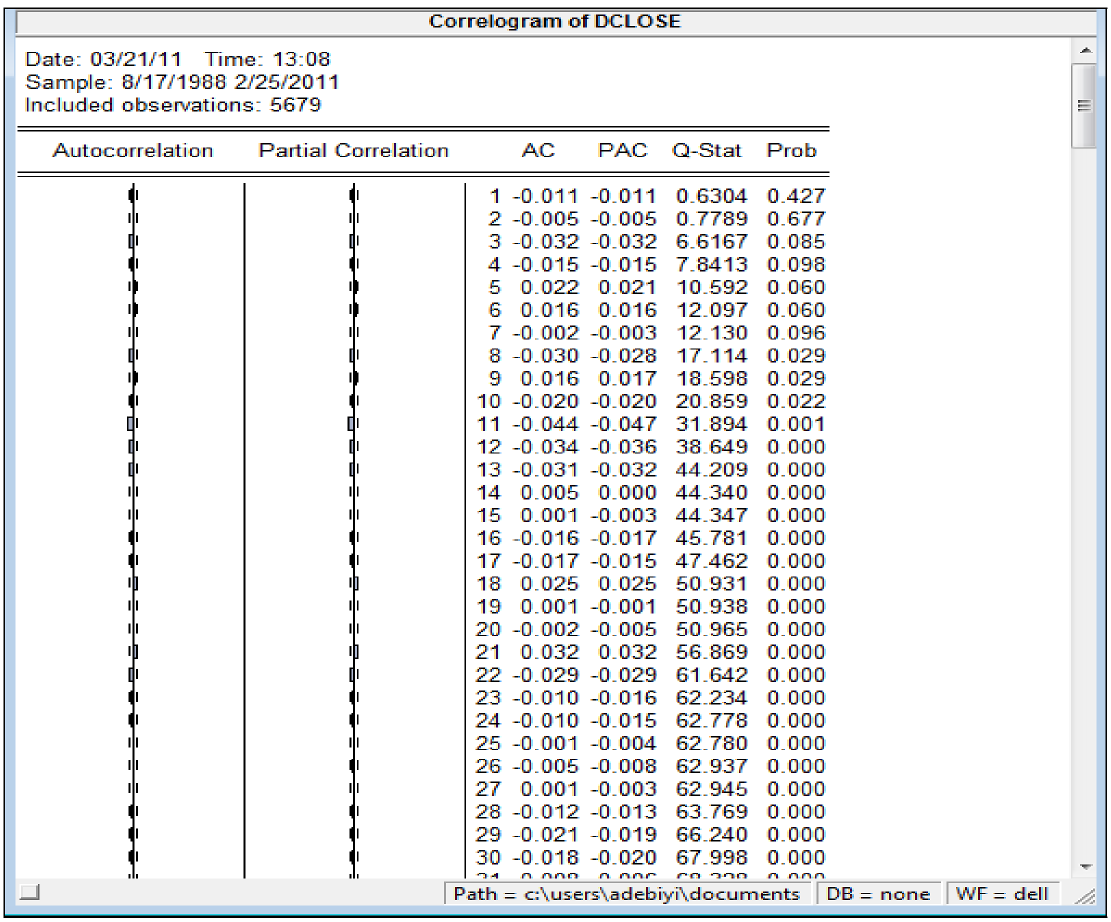
Figure 3.3: The correlogram of Dell stock price index.

Stock Price



Trading Years

Figure 3.4: Graphical representation of the Dell stock price index after differencing.



Correlogram of DCLOSE

Date: 03/21/11 Time: 13:08
Sample: 8/17/1988 2/25/2011
Included observations: 5679

	AC	PAC	Q-Stat	Prob
1	-0.011	-0.011	0.6304	0.427
2	-0.005	-0.005	0.7789	0.677
3	-0.032	-0.032	6.6167	0.085
4	-0.015	-0.015	7.8413	0.098
5	0.022	0.021	10.592	0.060
6	0.016	0.016	12.097	0.060
7	-0.002	-0.003	12.130	0.096
8	-0.030	-0.028	17.114	0.029
9	0.016	0.017	18.598	0.029
10	-0.020	-0.020	20.859	0.022
11	-0.044	-0.047	31.894	0.001
12	-0.034	-0.036	38.649	0.000
13	-0.031	-0.032	44.209	0.000
14	0.005	0.000	44.340	0.000
15	0.001	-0.003	44.347	0.000
16	-0.016	-0.017	45.781	0.000
17	-0.017	-0.015	47.462	0.000
18	0.025	0.025	50.931	0.000
19	0.001	-0.001	50.938	0.000
20	-0.002	-0.005	50.965	0.000
21	0.032	0.032	56.869	0.000
22	-0.029	-0.029	61.642	0.000
23	-0.010	-0.016	62.234	0.000
24	-0.010	-0.015	62.778	0.000
25	-0.001	-0.004	62.780	0.000
26	-0.005	-0.008	62.937	0.000
27	0.001	-0.003	62.945	0.000
28	-0.012	-0.013	63.769	0.000
29	-0.021	-0.019	66.240	0.000
30	-0.018	-0.020	67.998	0.000

Path = c:\users\adebiyi\documents DB = none WF = dell

Figure 3.5: The correlogram of Dell stock price index after first differencing

Now, the first-difference of the series “DCLOSE” shows that the series is stationary as shown in the line graph of figure 3.4 and the white noise shows no significant patterns in the graph of correlogram in figure 3.5. The Augmented Dickey-Fuller (ADF) unit root test on “DCLOSE” also confirms that the first-difference of the series becomes stationary. This indicates that there is strong evidence that support that the ARIMA (0, 1, 0) is suitable for the time series as shown in figure 3.6.

Augmented Dickey-Fuller Unit Root Test on DCLOSE

Null Hypothesis: DCLOSE has a unit root
Exogenous: Constant
Lag Length: 0 (Automatic based on SIC, MAXLAG=32)

		t-Statistic	Prob.*
Augmented Dickey-Fuller test statistic		-76.13704	0.0001
Test critical values:	1% level	-3.431322	
	5% level	-2.861854	
	10% level	-2.566980	

*MacKinnon (1996) one-sided p-values.

Augmented Dickey-Fuller Test Equation
Dependent Variable: D(DCLOSE)
Method: Least Squares
Date: 03/21/11 Time: 13:09
Sample (adjusted): 8/19/1988 2/25/2011
Included observations: 5678 after adjustments

Variable	Coefficient	Std. Error	t-Statistic	Prob.
DCLOSE(-1)	-1.010533	0.013273	-76.13704	0.0000
C	0.001091	0.031376	0.034758	0.9723

R-squared	0.505267	Mean dependent var	3.70E-05
Adjusted R-squared	0.505180	S.D. dependent var	3.361047
S.E. of regression	2.364277	Akaike info criterion	4.559174
Sum squared resid	31727.74	Schwarz criterion	4.561514
Log likelihood	-12941.49	F-statistic	5796.850
Durbin-Watson stat	2.000106	Prob(F-statistic)	0.000000

Figure 3.6: ADF unit root test for DCLOSE of Dell stock index.

In order to construct the best ARIMA model for Dell stock index. The next step is to determine how many autoregressive (p) and moving average (q) parameters are necessary to give an effective model. The following criteria are used in this study to determine the best model as follows:

- Relatively small of BIC (Bayesian or Schwarz Information Criterion which is measured by $n\log(SEE)+k\log(n)$)

- Relatively small of SEE (S.E. of regression)
- Relatively high of adjusted R^2
- Q-statistics and correlogram show that there is no significant pattern left in the ACFs and PACFs of the residuals, it means the residual of the selected model are white noise.

Table 3.1 shows the different parameters of autoregressive (*p*) and moving average (*q*) in the ARIMA model. ARIMA (1, 0, 0) is the best for Dell stock index as shown in figure 3.7.

Dependent Variable: CLOSE
Method: Least Squares
Date: 03/21/11 Time: 15:54
Sample (adjusted): 8/18/1988 2/25/2011
Included observations: 5679 after adjustments
Convergence achieved after 4 iterations

Variable	Coefficient	Std. Error	t-Statistic	Prob.
C	34.11484	6.028238	5.659173	0.0000
AR(1)	0.994802	0.001346	739.1456	0.0000

R-squared	0.989716	Mean dependent var	33.91262
Adjusted R-squared	0.989714	S.D. dependent var	23.28046
S.E. of regression	2.361101	Akaike info criterion	4.556485
Sum squared resid	31648.13	Schwarz criterion	4.558825
Log likelihood	-12936.14	F-statistic	546336.2
Durbin-Watson stat	2.015870	Prob(F-statistic)	0.000000

Inverted AR Roots .99

Figure 3.7: ARIMA (1, 0, 0) estimation output with CLOSE of Dell index.

In forecasting form, the best model selected can be expressed as follows:

$$Y_t = \phi_1 Y_{t-1} + \theta_0 + \varepsilon_t \qquad (3.13)$$

Where, $\varepsilon_t = Y_t - \hat{Y}_t$ (i.e., the difference between the actual value of the series and the forecast value)

Table 3.1: Statistical results of different ARIMA parameters for Dell Stock Index

ARIMA	BIC	Adjusted R^2	SEE
(1, 0, 0)	**4.5588**	**0.9897**	**2.3611**
(1, 0, 1)	4.5602	0.9897	2.3612
(2, 0, 0)	5.2389	0.9796	3.3174
(0, 0, 1)	7.8883	0.7127	12.4770
(0, 0, 2)	7.9369	0.6984	12.7839
(1, 1, 0)	4.5615	-0.0000	2.3642
(0, 1, 0)	4.5599	0.0000	2.3639
(0, 1, 1)	4.5615	-0.0000	2.3642
(1, 1,2)	4.5630	-0.0002	2.3644
(2, 1, 0)	4.5617	-0.0001	2.3645
(2, 1, 2)	4.5610	0.0019	2.3621

3.3.2 ARIMA (*p, d, q*) Model Development of Stock Price of Nokia Corporation

Similarly, the Nokia Inc. stock data used in this study covers the period from 25th April, 1995 to 25th February, 2011 having a total number of 3990 observations. Figure 3.8 depicts the original pattern of the time series of the index in order to get a general overview whether the time series is stationary or not. From the graph below the time series is likely to have random walk pattern.

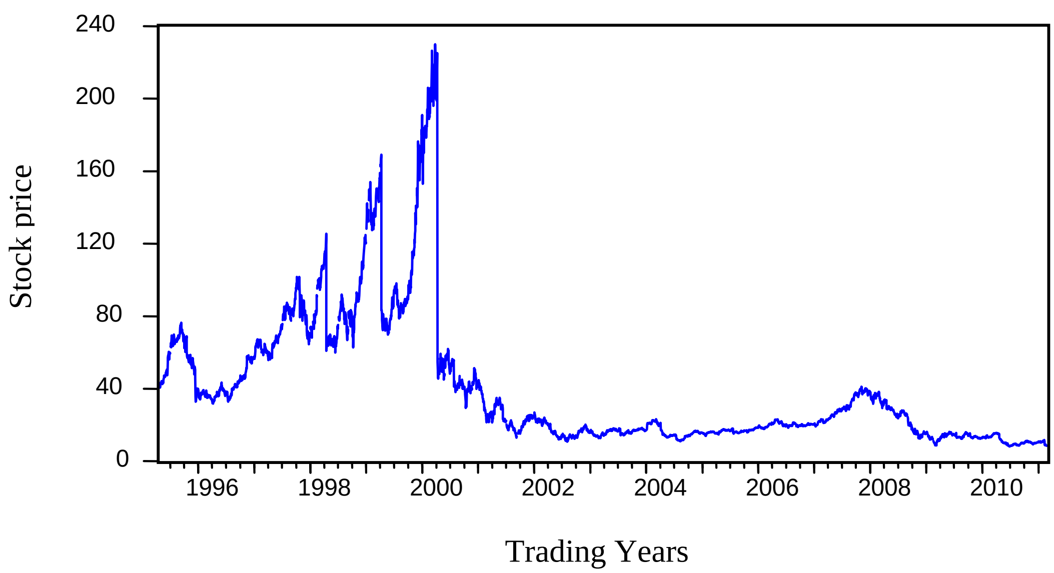
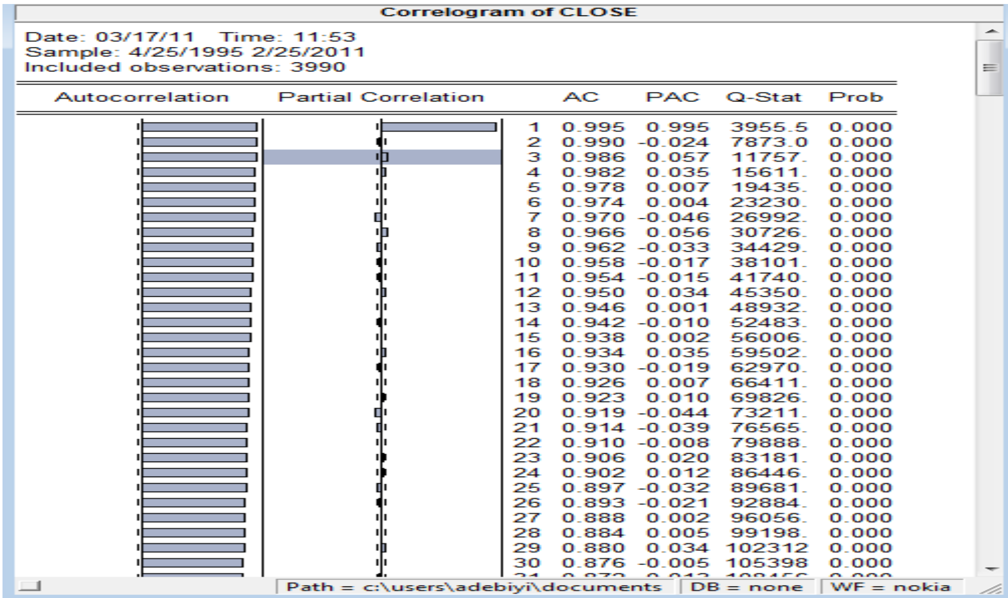


Figure 3.8: Graphical representation of the Nokia stock closing price index



Correlogram of CLOSE

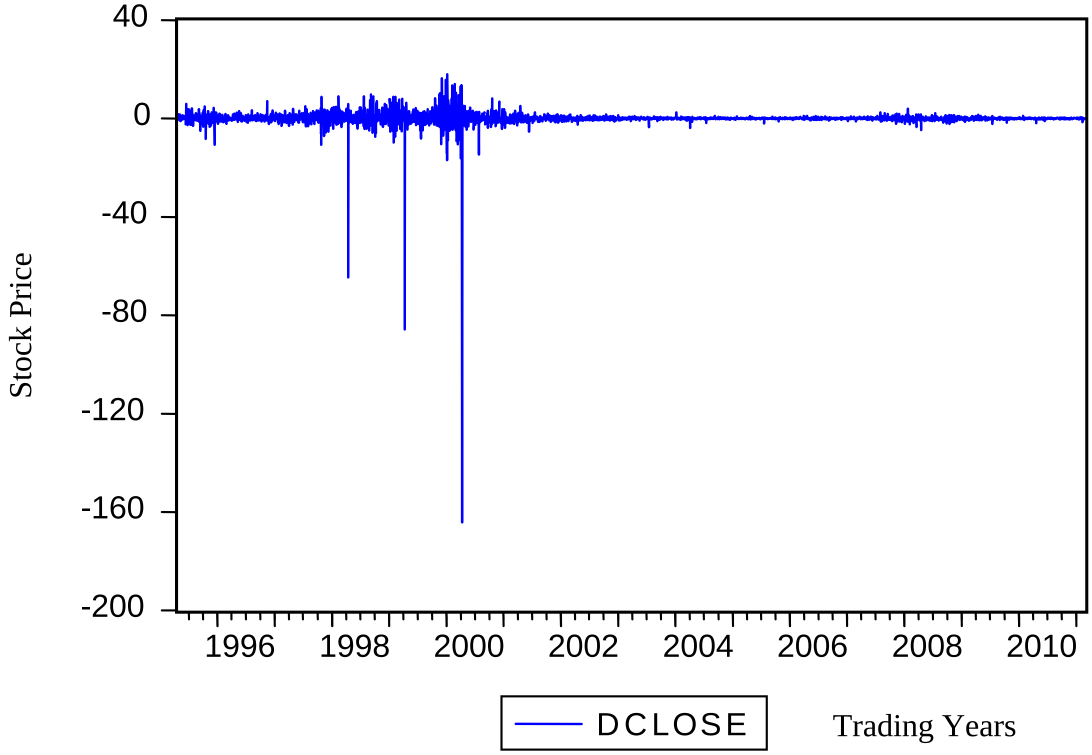
Date: 03/17/11 Time: 11:53
Sample: 4/25/1995 2/25/2011
Included observations: 3990

Autocorrelation	Partial Correlation		AC	PAC	Q-Stat	Prob
		1	0.995	0.995	3955.5	0.000
		2	0.990	-0.024	7873.0	0.000
		3	0.986	0.057	11757.	0.000
		4	0.982	0.035	15611.	0.000
		5	0.978	0.007	19435.	0.000
		6	0.974	0.004	23230.	0.000
		7	0.970	-0.046	26992.	0.000
		8	0.966	0.056	30726.	0.000
		9	0.962	-0.033	34429.	0.000
		10	0.958	-0.017	38101.	0.000
		11	0.954	-0.015	41740.	0.000
		12	0.950	0.034	45350.	0.000
		13	0.946	0.001	48932.	0.000
		14	0.942	-0.010	52483.	0.000
		15	0.938	0.002	56006.	0.000
		16	0.934	0.035	59502.	0.000
		17	0.930	-0.019	62970.	0.000
		18	0.926	0.007	66411.	0.000
		19	0.923	0.010	69826.	0.000
		20	0.919	-0.044	73211.	0.000
		21	0.914	-0.039	76565.	0.000
		22	0.910	-0.008	79888.	0.000
		23	0.906	0.020	83181.	0.000
		24	0.902	0.012	86446.	0.000
		25	0.897	-0.032	89681.	0.000
		26	0.893	-0.021	92884.	0.000
		27	0.888	0.002	96056.	0.000
		28	0.884	0.005	99198.	0.000
		29	0.880	0.034	102312	0.000
		30	0.876	-0.005	105398	0.000

Path = c:\users\adebiyi\documents DB = none WF = nokia

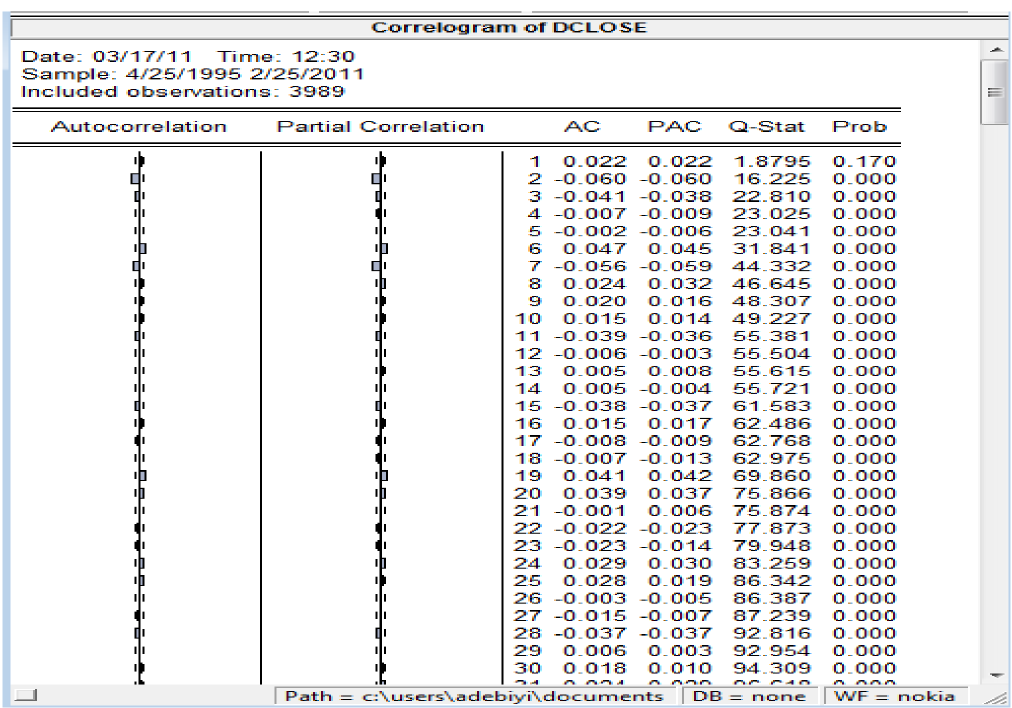
Figure 3.9: The correlogram of Nokia stock price index

Figure 3.9 is the correlogram of the time series of Nokia stock index. From the graph of the correlogram, the ACF dies down extremely slowly which simply means that the time series is nonstationary. If the series is not stationary, it can often be converted to a stationary series by differencing. After the first difference, the series “DCLOSE” of Nokia stock index becomes stationary as shown in figure 3.10 and figure 3.11 of the line graph and correlogram of the series after first differencing.



.

Figure 3.10: Graphical representation of the Nokia stock price index after differencing.



Correlogram of DCLOSE

Date: 03/17/11 Time: 12:30
Sample: 4/25/1995 2/25/2011
Included observations: 3989

Autocorrelation	Partial Correlation		AC	PAC	Q-Stat	Prob
		1	0.022	0.022	1.8795	0.170
		2	-0.060	-0.060	16.225	0.000
		3	-0.041	-0.038	22.810	0.000
		4	-0.007	-0.009	23.025	0.000
		5	-0.002	-0.006	23.041	0.000
		6	0.047	0.045	31.841	0.000
		7	-0.056	-0.059	44.332	0.000
		8	0.024	0.032	46.645	0.000
		9	0.020	0.016	48.307	0.000
		10	0.015	0.014	49.227	0.000
		11	-0.039	-0.036	55.381	0.000
		12	-0.006	-0.003	55.504	0.000
		13	0.005	0.008	55.615	0.000
		14	0.005	-0.004	55.721	0.000
		15	-0.038	-0.037	61.583	0.000
		16	0.015	0.017	62.486	0.000
		17	-0.008	-0.009	62.768	0.000
		18	-0.007	-0.013	62.975	0.000
		19	0.041	0.042	69.860	0.000
		20	0.039	0.037	75.866	0.000
		21	-0.001	0.006	75.874	0.000
		22	-0.022	-0.023	77.873	0.000
		23	-0.023	-0.014	79.948	0.000
		24	0.029	0.030	83.259	0.000
		25	0.028	0.019	86.342	0.000
		26	-0.003	-0.005	86.387	0.000
		27	-0.015	-0.007	87.239	0.000
		28	-0.037	-0.037	92.816	0.000
		29	0.006	0.003	92.954	0.000
		30	0.018	0.010	94.309	0.000

Path = c:\users\adebiyi\documents | DB = none | WF = nokia

Figure 3.11: The correlogram of Nokia stock price index after first differencing

In figure 3.12 the model checking was done with ADF unit root test on “DCLOSE” of Nokia of stock index. The result confirms that the series becomes stationary after the first-difference of the series.

Augmented Dickey-Fuller Unit Root Test on DCLOSE

Null Hypothesis: DCLOSE has a unit root
Exogenous: Constant
Lag Length: 1 (Automatic based on SIC, MAXLAG=30)

		t-Statistic	Prob.*
Augmented Dickey-Fuller test statistic		-46.89879	0.0001
Test critical values:	1% level	-3.431805	
	5% level	-2.862068	
	10% level	-2.567094	

*MacKinnon (1996) one-sided p-values.

Augmented Dickey-Fuller Test Equation
Dependent Variable: D(DCLOSE)
Method: Least Squares
Date: 03/17/11 Time: 12:37
Sample (adjusted): 4/28/1995 2/25/2011
Included observations: 3987 after adjustments

Variable	Coefficient	Std. Error	t-Statistic	Prob.
DCLOSE(-1)	-1.037429	0.022121	-46.89879	0.0000
D(DCLOSE(-1))	0.060437	0.015814	3.821732	0.0001
C	-0.008294	0.056703	-0.146267	0.8837

R-squared	0.491018	Mean dependent var	0.000100
Adjusted R-squared	0.490762	S.D. dependent var	5.017265
S.E. of regression	3.580364	Akaike info criterion	5.389558
Sum squared resid	51070.94	Schwarz criterion	5.394292
Log likelihood	-10741.08	F-statistic	1921.694
Durbin-Watson stat	2.004589	Prob(F-statistic)	0.000000

Path = c:\users\adebiyi\documents | DB = none | WF = nokia

Figure 3.12: ADF unit root test for DCLOSE of Nokia stock index.

Table 3.2 shows the different parameters of autoregressive (p) and moving average (q) in the ARIMA model. ARIMA (2, 1, 0) is the best for Nokia stock index as shown in figure 3.13.

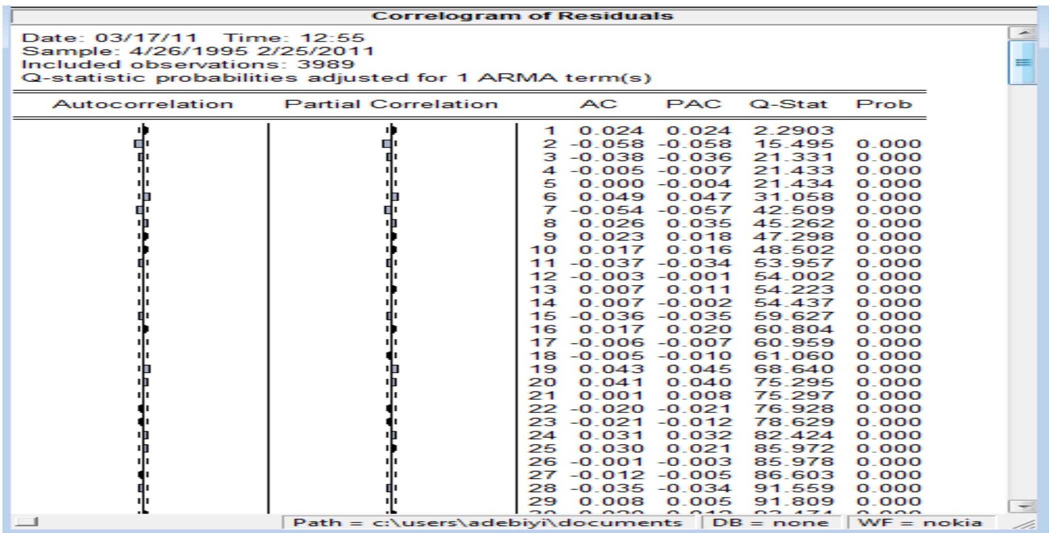
Dependent Variable: DCLOSE
Method: Least Squares
Date: 03/19/11 Time: 16:08
Sample (adjusted): 4/28/1995 2/25/2011
Included observations: 3987 after adjustments
Convergence achieved after 3 iterations

Variable	Coefficient	Std. Error	t-Statistic	Prob.
C	-0.007997	0.053504	-0.149458	0.8812
AR(2)	-0.059938	0.015813	-3.790524	0.0002

R-squared	0.003593	Mean dependent var	-0.007988
Adjusted R-squared	0.003343	S.D. dependent var	3.586866
S.E. of regression	3.580866	Akaike info criterion	5.389588
Sum squared resid	51098.07	Schwarz criterion	5.392744
Log likelihood	-10742.14	F-statistic	14.36807
Durbin-Watson stat	1.958555	Prob(F-statistic)	0.000153

Figure 3.13: ARIMA (2, 1, 0) estimation output with DCLOSE of Nokia index.

Figure 3.14 is the residual of the series. If the model is good, the residuals (difference between actual and predicted values) of the model are a series of random errors. Since there are no significant spikes of ACFs and PACFs, it means that the residual of the selected ARIMA model are white noise, no other significant patterns left in the time series. Therefore, there is no need to consider any AR(p) and MA(q) further.



Correlogram of Residuals

Date: 03/17/11 Time: 12:55
Sample: 4/26/1995 2/25/2011
Included observations: 3989
Q-statistic probabilities adjusted for 1 ARMA term(s)

	AC	PAC	Q-Stat	Prob
1	0.024	0.024	2.2903	
2	-0.058	-0.058	15.495	0.000
3	-0.038	-0.036	21.331	0.000
4	-0.005	-0.007	21.433	0.000
5	0.000	-0.004	21.434	0.000
6	0.049	0.047	31.058	0.000
7	-0.054	-0.057	42.509	0.000
8	0.026	0.035	45.262	0.000
9	0.023	0.018	47.298	0.000
10	0.017	0.016	48.502	0.000
11	-0.037	-0.034	53.957	0.000
12	-0.003	-0.001	54.002	0.000
13	0.007	0.011	54.223	0.000
14	0.007	-0.002	54.437	0.000
15	-0.036	-0.035	59.627	0.000
16	0.017	0.020	60.804	0.000
17	-0.006	-0.007	60.959	0.000
18	-0.005	-0.010	61.060	0.000
19	0.043	0.045	68.640	0.000
20	0.041	0.040	75.295	0.000
21	0.001	0.008	75.297	0.000
22	-0.020	-0.021	76.928	0.000
23	-0.021	-0.012	78.629	0.000
24	0.031	0.032	82.424	0.000
25	0.030	0.021	85.972	0.000
26	-0.001	-0.003	85.978	0.000
27	-0.012	-0.005	86.603	0.000
28	-0.035	-0.034	91.559	0.000
29	0.008	0.005	91.809	0.000

Path = c:\users\adebiyi\documents DB = none WF = nokia

Figure 3.14: Correlogram of residuals of the Nokia stock index.

In forecasting form, the best model selected can be expressed as follows:

$$Y_t = \theta_1 Y_{t-1} + \theta_2 Y_{t-2} + \varepsilon_t \quad (3.14)$$

Where, $\varepsilon_t = Y_t - \hat{Y}_t$ (i.e., the difference between the actual value of the series and the forecast value)

Table 3.2: Statistical results of different ARIMA parameters for Nokia Stock Index

ARIMA	BIC	Adjusted R^2	SEE
(1, 0, 0)	5.3936	0.9907	3.5824
(1, 0, 1)	5.3950	0.9907	3.5817
(2, 0, 0)	6.1061	0.9811	5.1157
(0, 0, 1)	8.8324	0.7126	19.9942
(0, 0, 2)	8.8871	0.6964	20.5490
(1, 1, 0)	5.3956	0.0002	3.5860
(0, 1, 0)	5.3937	0.0000	3.5859
(0, 1, 1)	5.3953	0.0002	3.5854
(1, 1,2)	5.3941	0.0035	3.5800
(2, 1, 0)	**5.3927**	**0.0033**	**3.5808**
(2, 1, 2)	5.3947	0.0031	3.5812

3.3.3 ARIMA (*p, d, q*) Model Development of Stock Price of Zenith Bank

The stock data of Zenith bank used in this study covered the period from 3rd January, 2006 to 25th February, 2011 with total of 1296 observations. Figure 3.15 is the original pattern of the series. From the graph there was upward movement of the index from 2006 and downward movement is observed from 2008 possibly because of world financial crisis experienced up till now.

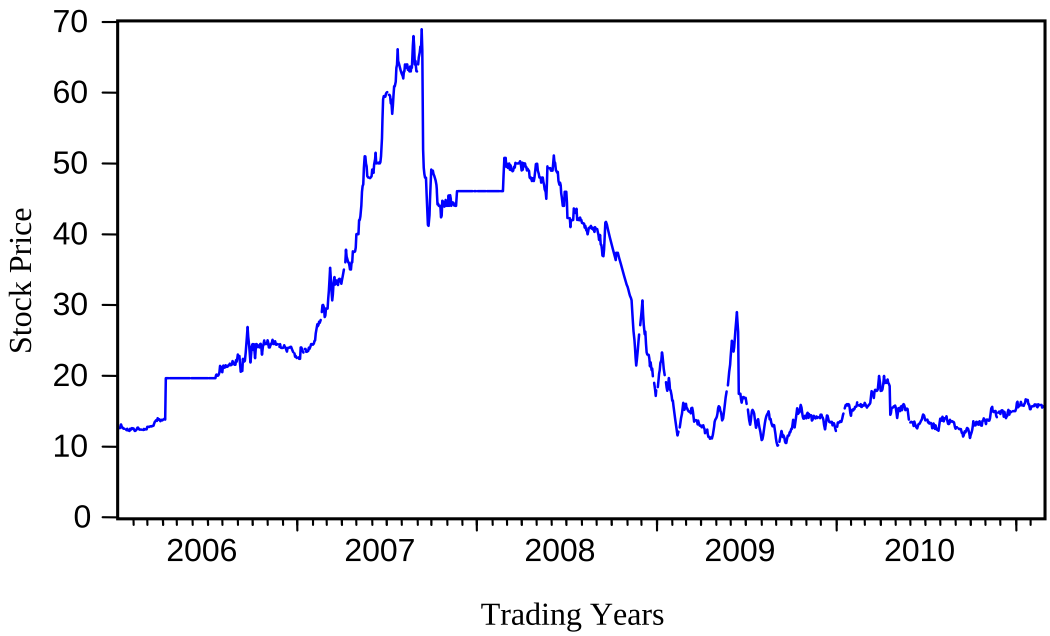
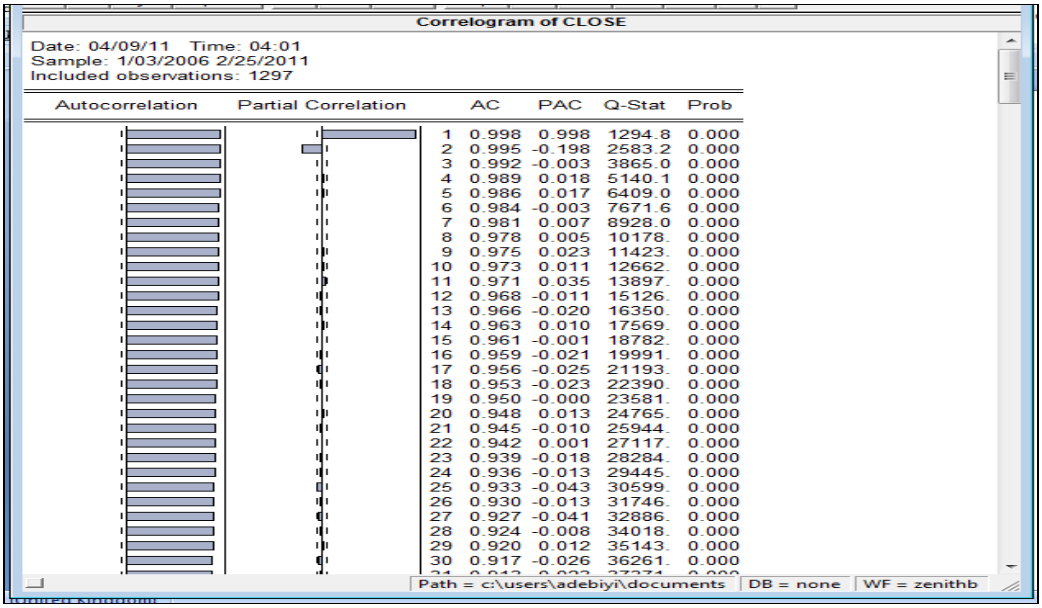


Figure 3.15: Graphical representation of the Zenith Bank stock price index.



Correlogram of CLOSE

Date: 04/09/11 Time: 04:01
Sample: 1/03/2006 2/25/2011
Included observations: 1297

Autocorrelation	Partial Correlation		AC	PAC	Q-Stat	Prob
		1	0.998	0.998	1294.8	0.000
		2	0.995	-0.198	2583.2	0.000
		3	0.992	-0.003	3865.0	0.000
		4	0.989	0.018	5140.1	0.000
		5	0.986	0.017	6409.0	0.000
		6	0.984	-0.003	7671.6	0.000
		7	0.981	0.007	8928.0	0.000
		8	0.978	0.005	10178.	0.000
		9	0.975	0.023	11423.	0.000
		10	0.973	0.011	12662.	0.000
		11	0.971	0.035	13897.	0.000
		12	0.968	-0.011	15126.	0.000
		13	0.966	-0.020	16350.	0.000
		14	0.963	0.010	17569.	0.000
		15	0.961	-0.001	18782.	0.000
		16	0.959	-0.021	19991.	0.000
		17	0.956	-0.025	21193.	0.000
		18	0.953	-0.023	22390.	0.000
		19	0.950	-0.000	23581.	0.000
		20	0.948	0.013	24765.	0.000
		21	0.945	-0.010	25944.	0.000
		22	0.942	0.001	27117.	0.000
		23	0.939	-0.018	28284.	0.000
		24	0.936	-0.013	29445.	0.000
		25	0.933	-0.043	30599.	0.000
		26	0.930	-0.013	31746.	0.000
		27	0.927	-0.041	32886.	0.000
		28	0.924	-0.008	34018.	0.000
		29	0.920	0.012	35143.	0.000
		30	0.917	-0.026	36261.	0.000

Path = c:\users\adebiyi\documents DB = none WF = zenithb

Figure 3.16: The correlogram of Zenith Bank stock price index

Figure 3.16 is the correlogram of the time series of Zenith bank stock index. From the graph of the correlogram, the ACF dies down extremely slowly which simply means that the time series is nonstationary. If the series is not stationary, it can often be converted to a stationary series by differencing. After the first difference, the series "DCLOSE" of Zenith bank stock index becomes stationary as shown in figure 3.17 and figure 3.18 of the line graph and correlogram of the series after first differencing.

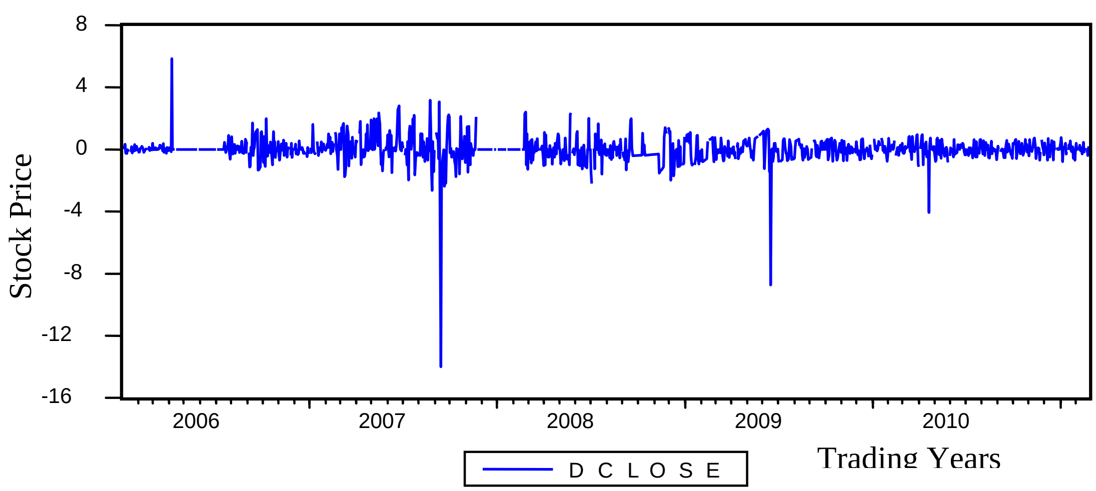


Figure 3.17: Graphical representation of the Zenith bank stock index first differencing

Correlogram of DCLOSE

Date: 04/09/11 Time: 04:03
Sample: 1/03/2006 2/25/2011
Included observations: 1296

Autocorrelation	Partial Correlation		AC	PAC	Q-Stat	Prob
		1	0.268	0.268	92.960	0.000
		2	0.063	-0.010	98.041	0.000
		3	-0.009	-0.025	98.141	0.000
		4	-0.036	-0.029	99.856	0.000
		5	-0.015	0.003	100.17	0.000
		6	-0.012	-0.008	100.36	0.000
		7	-0.020	-0.017	100.88	0.000
		8	-0.034	-0.028	102.43	0.000
		9	-0.027	-0.011	103.37	0.000
		10	-0.060	-0.053	108.09	0.000
		11	-0.004	0.026	108.11	0.000
		12	0.018	0.014	108.53	0.000
		13	-0.002	-0.016	108.54	0.000
		14	-0.002	-0.002	108.54	0.000
		15	0.030	0.034	109.69	0.000
		16	0.041	0.026	111.91	0.000
		17	0.040	0.019	114.03	0.000
		18	0.019	-0.000	114.52	0.000
		19	-0.007	-0.011	114.58	0.000
		20	0.003	0.010	114.59	0.000
		21	0.000	0.001	114.59	0.000
		22	0.013	0.017	114.82	0.000
		23	0.025	0.019	115.67	0.000
		24	0.064	0.059	121.11	0.000
		25	0.030	0.004	122.28	0.000
		26	0.059	0.057	126.82	0.000
		27	0.023	-0.003	127.53	0.000
		28	-0.015	-0.020	127.82	0.000
		29	0.027	0.041	128.78	0.000
		30	0.036	0.030	130.52	0.000

Path = c:\users\adebiyi\documents | DB = none | WF = zenithb

Figure 3.18: The correlogram of zenith bank stock price index after first differencing.

Figure 3.19 is the ADF unit root test on “DCLOSE” of the series which also indicates when the first difference of the series becomes stationary.

Augmented Dickey-Fuller Unit Root Test on DCLOSE

Null Hypothesis: DCLOSE has a unit root
Exogenous: Constant
Lag Length: 0 (Automatic based on SIC, MAXLAG=22)

		t-Statistic	Prob.*
Augmented Dickey-Fuller test statistic		-27.33425	0.0000
Test critical values:	1% level	-3.435188	
	5% level	-2.863564	
	10% level	-2.567897	

*MacKinnon (1996) one-sided p-values.

Augmented Dickey-Fuller Test Equation
Dependent Variable: D(DCLOSE)
Method: Least Squares
Date: 04/09/11 Time: 04:04
Sample (adjusted): 1/05/2006 2/25/2011
Included observations: 1295 after adjustments

Variable	Coefficient	Std. Error	t-Statistic	Prob.
DCLOSE(-1)	-0.732475	0.026797	-27.33425	0.0000
C	0.001779	0.021843	0.081457	0.9351

R-squared	0.366226	Mean dependent var	0.000139
Adjusted R-squared	0.365736	S.D. dependent var	0.986995
S.E. of regression	0.786049	Akaike info criterion	2.357948
Sum squared resid	798.9102	Schwarz criterion	2.365927
Log likelihood	-1524.772	F-statistic	747.1610
Durbin-Watson stat	1.994624	Prob(F-statistic)	0.000000

Path = c:\users\adebiyi\documents | DB = none | WF = zenithb

Figure 3.19: ADF unit root test for DCLOSE of Zenith bank stock index.

Table 3.3 shows the different parameters of autoregressive (p) and moveing average (q) of the ARIMA model in order to get the best fitted model. ARIMA (1, 0, 1) is relatively the best model as indicated in figure 3.20.

Dependent Variable: CLOSE
Method: Least Squares
Date: 04/09/11 Time: 04:24
Sample (adjusted): 1/04/2006 2/25/2011
Included observations: 1296 after adjustments
Convergence achieved after 7 iterations
Backcast: 1/03/2006

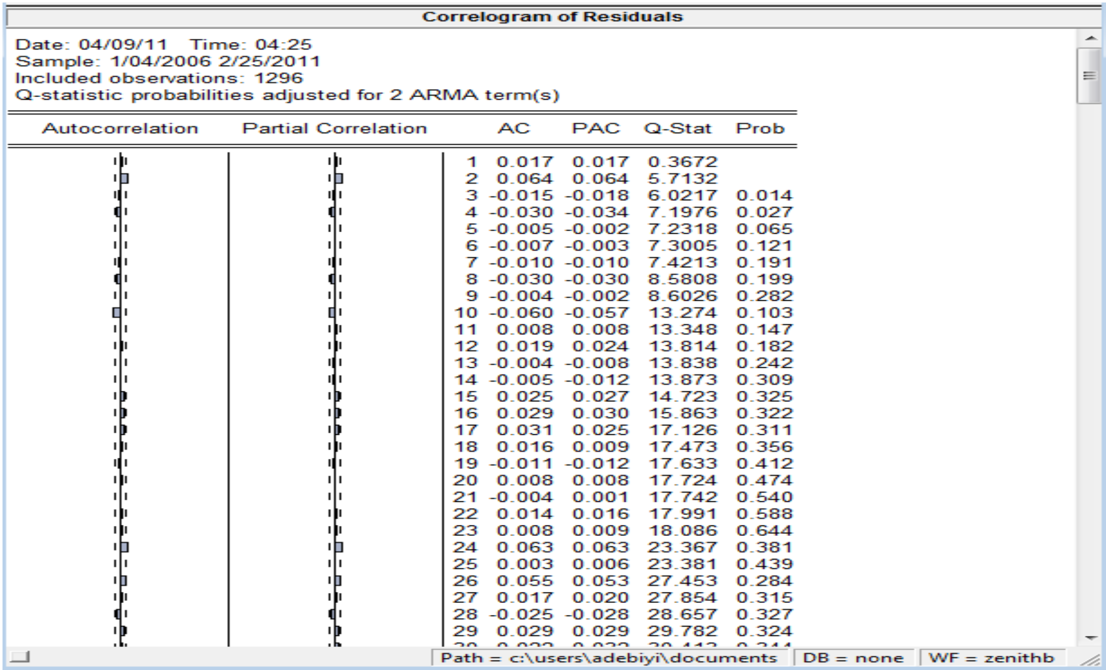
Variable	Coefficient	Std. Error	t-Statistic	Prob.
C	27.76649	11.97506	2.318694	0.0206
AR(1)	0.997704	0.001829	545.5149	0.0000
MA(1)	0.254140	0.026934	9.435526	0.0000

R-squared	0.997254	Mean dependent var	26.70431
Adjusted R-squared	0.997250	S.D. dependent var	15.01153
S.E. of regression	0.787215	Akaike info criterion	2.361682
Sum squared resid	801.2819	Schwarz criterion	2.373642
Log likelihood	-1527.370	F-statistic	234805.8
Durbin-Watson stat	1.966274	Prob(F-statistic)	0.000000

Inverted AR Roots	1.00
Inverted MA Roots	-.25

Figure 3.20: ARIMA (1, 0, 1) estimation output with DCLOSE of Zenith bank index.

Figure 3.21 is the correlogram of residual of the seies. From the figure it is obvious there is no significant spike of ACFs and PACFs. This means that the residual of this selected ARIMA model are white noise. There is no other significant patterns left in the time series and there is no need for further consideration of another AR(p) and MA(q).



Correlogram of Residuals

Date: 04/09/11 Time: 04:25
Sample: 1/04/2006 2/25/2011
Included observations: 1296
Q-statistic probabilities adjusted for 2 ARMA term(s)

Autocorrelation	Partial Correlation		AC	PAC	Q-Stat	Prob
		1	0.017	0.017	0.3672	
		2	0.064	0.064	5.7132	
		3	-0.015	-0.018	6.0217	0.014
		4	-0.030	-0.034	7.1976	0.027
		5	-0.005	-0.002	7.2318	0.065
		6	-0.007	-0.003	7.3005	0.121
		7	-0.010	-0.010	7.4213	0.191
		8	-0.030	-0.030	8.5808	0.199
		9	-0.004	-0.002	8.6026	0.282
		10	-0.060	-0.057	13.274	0.103
		11	0.008	0.008	13.348	0.147
		12	0.019	0.024	13.814	0.182
		13	-0.004	-0.008	13.838	0.242
		14	-0.005	-0.012	13.873	0.309
		15	0.025	0.027	14.723	0.325
		16	0.029	0.030	15.863	0.322
		17	0.031	0.025	17.126	0.311
		18	0.016	0.009	17.473	0.356
		19	-0.011	-0.012	17.633	0.412
		20	0.008	0.008	17.724	0.474
		21	-0.004	0.001	17.742	0.540
		22	0.014	0.016	17.991	0.588
		23	0.008	0.009	18.086	0.644
		24	0.063	0.063	23.367	0.381
		25	0.003	0.006	23.381	0.439
		26	0.055	0.053	27.453	0.284
		27	0.017	0.020	27.854	0.315
		28	-0.025	-0.028	28.657	0.327
		29	0.029	0.029	29.782	0.324

Path = c:\users\adebiyi\documents | DB = none | WF = zenithb

Figure 3.21: Correlogram of residuals of the Zenith bank stock index.

In forecasting form, the best model selected can be expressed as follows:

$$Y_t = \phi_1 Y_{t-1} - \theta_1 \varepsilon_{t-1} + \varepsilon_t \qquad 3.15$$

Where,

$\varepsilon_t = Y_t - \hat{Y}_t$ (i.e., the difference between the actual value of the series and the forecast value)

Table 3.3: Statistical results of different ARIMA parameters for Zenith bank Stock Index

ARIMA	BIC	Adjusted R^2	SEE
(1, 0, 0)	2.4385	0.9970	0.8151
(1, 0, 1)	**2.3736**	**0.9972**	**0.7872**
(2, 0, 0)	3.3682	0.9925	1.2974
(0, 0, 1)	6.9285	0.7372	7.6951
(0, 0, 2)	6.9815	0.7228	7.9018
(1, 1, 0)	2.3659	0.0708	0.7860
(0, 1, 0)	2.4338	0.0000	0.8151
(0, 1, 1)	2.3693	0.0669	0.7873
(1, 1,2)	2.3714	0.0701	0.7863
(2, 1, 0)	2.4370	0.0031	0.8144
(2, 1, 2)	2.4412	0.0036	0.8142

3.3.4 ARIMA (*p, d, q*) Model Development of Stock Price of UBA Bank

The stock data of UBA bank used in this study covered the period from 3rd January, 2006 to 25th February, 2011 with total of 1296 observations. Figure 3.22 is the original pattern of the series. From the graph there was upward movement of the index from 2006 and downward movement is observed from 2008 possibly because of world financial crisis experienced up till now.

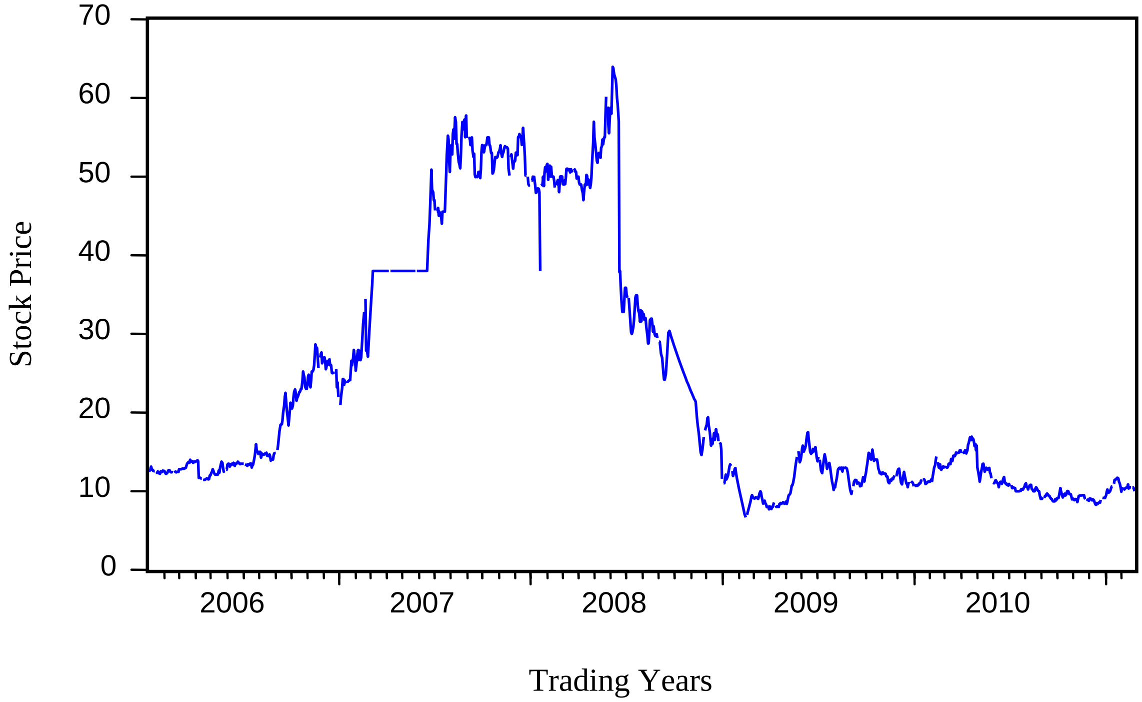
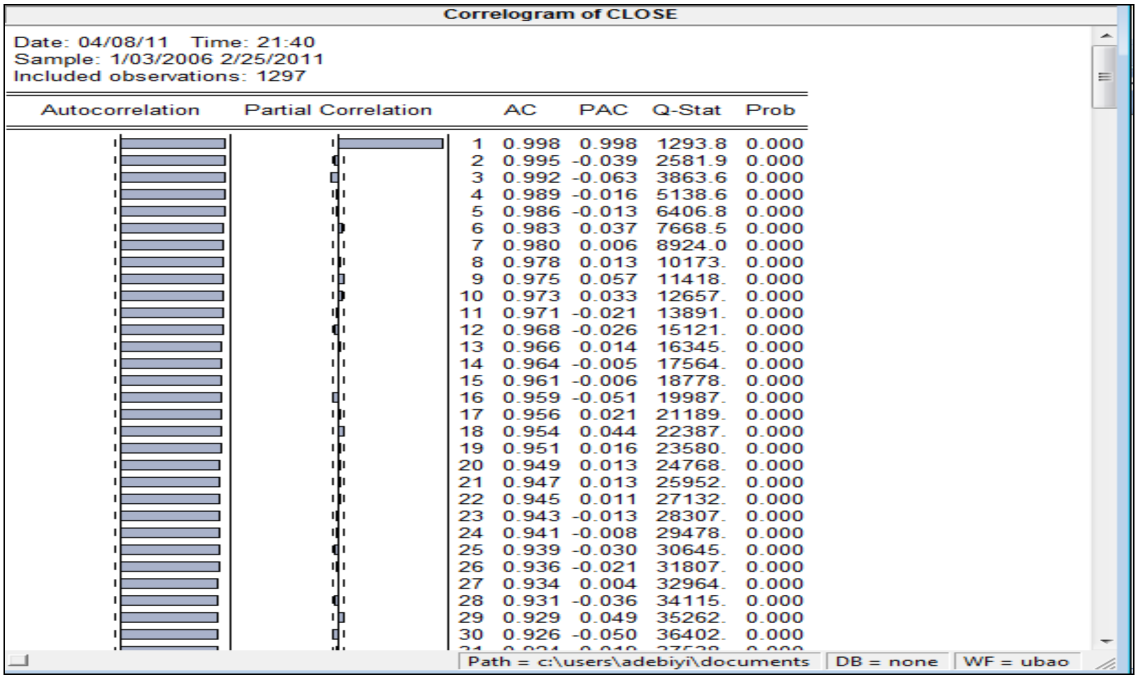


Figure 3.22: Graphical representation of UBA bank closing price of stock index


Correlogram of CLOSE

Date: 04/08/11 Time: 21:40
Sample: 1/03/2006 2/25/2011
Included observations: 1297

Autocorrelation	Partial Correlation		AC	PAC	Q-Stat	Prob
		1	0.998	0.998	1293.8	0.000
		2	0.995	-0.039	2581.9	0.000
		3	0.992	-0.063	3863.6	0.000
		4	0.989	-0.016	5138.6	0.000
		5	0.986	-0.013	6406.8	0.000
		6	0.983	0.037	7668.5	0.000
		7	0.980	0.006	8924.0	0.000
		8	0.978	0.013	10173.	0.000
		9	0.975	0.057	11418.	0.000
		10	0.973	0.033	12657.	0.000
		11	0.971	-0.021	13891.	0.000
		12	0.968	-0.026	15121.	0.000
		13	0.966	0.014	16345.	0.000
		14	0.964	-0.005	17564.	0.000
		15	0.961	-0.006	18778.	0.000
		16	0.959	-0.051	19987.	0.000
		17	0.956	0.021	21189.	0.000
		18	0.954	0.044	22387.	0.000
		19	0.951	0.016	23580.	0.000
		20	0.949	0.013	24768.	0.000
		21	0.947	0.013	25952.	0.000
		22	0.945	0.011	27132.	0.000
		23	0.943	-0.013	28307.	0.000
		24	0.941	-0.008	29478.	0.000
		25	0.939	-0.030	30645.	0.000
		26	0.936	-0.021	31807.	0.000
		27	0.934	0.004	32964.	0.000
		28	0.931	-0.036	34115.	0.000
		29	0.929	0.049	35262.	0.000
		30	0.926	-0.050	36402.	0.000

Path = c:\users\adebiyi\documents DB = none WF = ubao

Figure 3.23: The correlogram of UBA Bank stock price index

Figure 3.23 is the correlogram of the time series of UBA bank stock index. From the graph of the correlogram, the ACF dies down extremely slowly which simply means that the time series is nonstationary. If the series is not stationary, it can often be converted to a stationary series by differencing. After the first difference, the series “DCLOSE” of UBA bank stock index becomes stationary as shown in figure 3.24 and figure 3.25 of the line graph and correlogram of the series after first differencing.

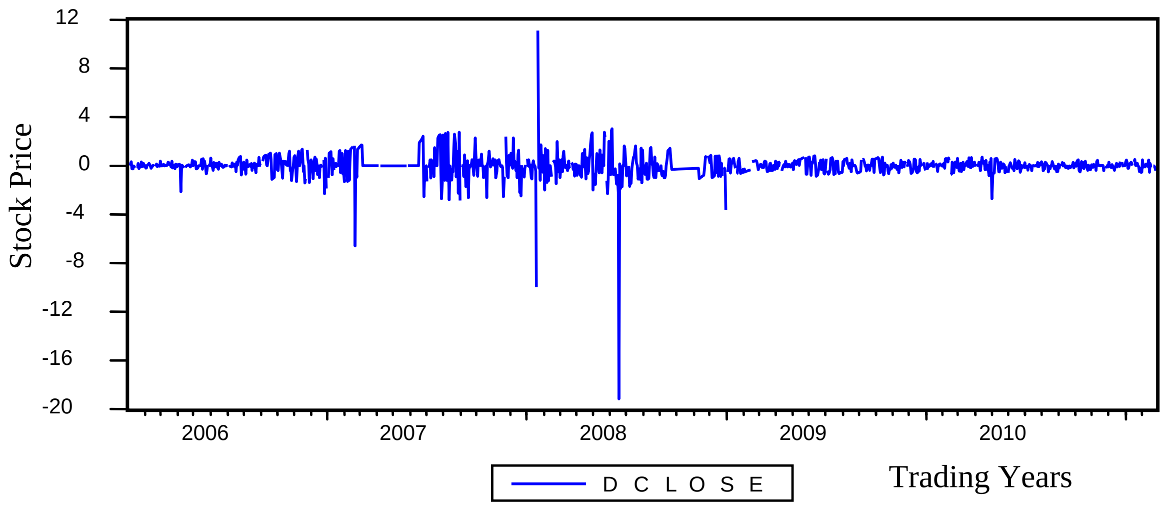
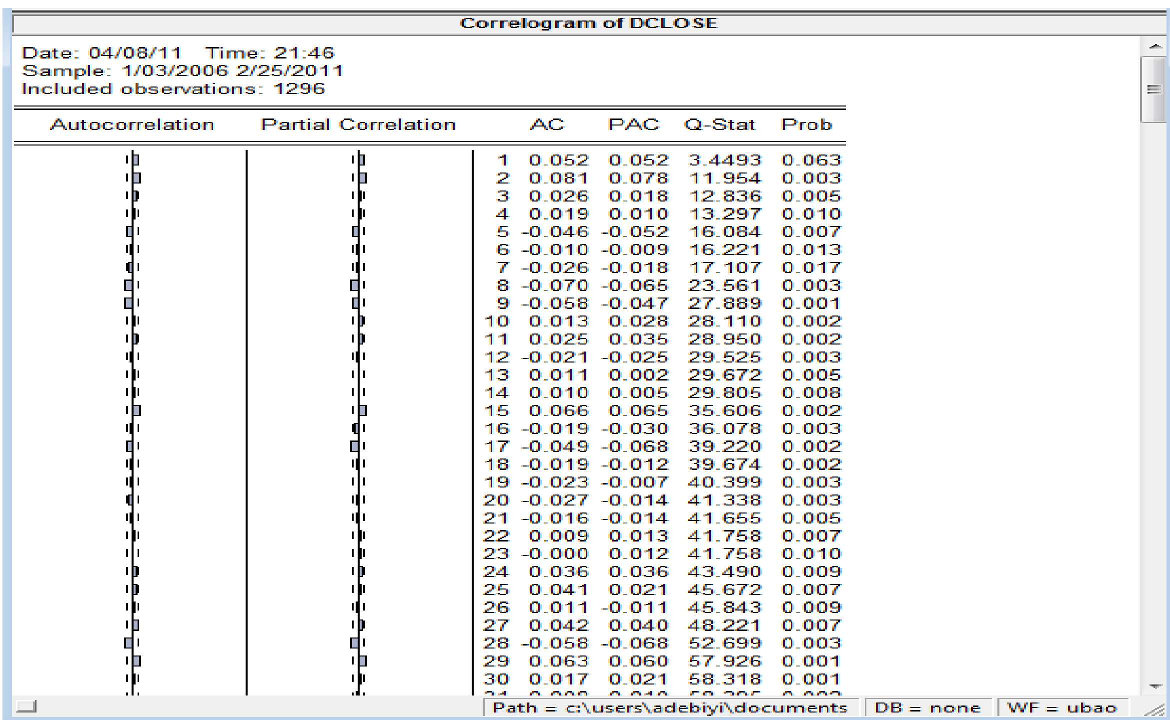


Figure 3.24: Graphical representation of UBA bank stock index after differencing



Correlogram of DCLOSE

Date: 04/08/11 Time: 21:46
Sample: 1/03/2006 2/25/2011
Included observations: 1296

	AC	PAC	Q-Stat	Prob
1	0.052	0.052	3.4493	0.063
2	0.081	0.078	11.954	0.003
3	0.026	0.018	12.836	0.005
4	0.019	0.010	13.297	0.010
5	-0.046	-0.052	16.084	0.007
6	-0.010	-0.009	16.221	0.013
7	-0.026	-0.018	17.107	0.017
8	-0.070	-0.065	23.561	0.003
9	-0.058	-0.047	27.889	0.001
10	0.013	0.028	28.110	0.002
11	0.025	0.035	28.950	0.002
12	-0.021	-0.025	29.525	0.003
13	0.011	0.002	29.672	0.005
14	0.010	0.005	29.805	0.008
15	0.066	0.065	35.606	0.002
16	-0.019	-0.030	36.078	0.003
17	-0.049	-0.068	39.220	0.002
18	-0.019	-0.012	39.674	0.002
19	-0.023	-0.007	40.399	0.003
20	-0.027	-0.014	41.338	0.003
21	-0.016	-0.014	41.655	0.005
22	0.009	0.013	41.758	0.007
23	-0.000	0.012	41.758	0.010
24	0.036	0.036	43.490	0.009
25	0.041	0.021	45.672	0.007
26	0.011	-0.011	45.843	0.009
27	0.042	0.040	48.221	0.007
28	-0.058	-0.068	52.699	0.003
29	0.063	0.060	57.926	0.001
30	0.017	0.021	58.318	0.001

Path = c:\users\adebiyi\documents DB = none WF = ubao

Figure 3.25: The correlogram of UBA Bank stock price index after differencing

Figure 3.26 is the ADF unit root test on "DCLOSE" of the series which also indicates the first difference of the series becomes stationary.

Augmented Dickey-Fuller Unit Root Test on DCLOSE

Null Hypothesis: DCLOSE has a unit root
Exogenous: Constant
Lag Length: 1 (Automatic based on SIC, MAXLAG=22)

		t-Statistic	Prob.*
Augmented Dickey-Fuller test statistic		-22.87128	0.0000
Test critical values:	1% level	-3.435192	
	5% level	-2.863566	
	10% level	-2.567898	

*MacKinnon (1996) one-sided p-values.

Augmented Dickey-Fuller Test Equation
Dependent Variable: D(DCLOSE)
Method: Least Squares
Date: 04/08/11 Time: 21:51
Sample (adjusted): 1/06/2006 2/25/2011
Included observations: 1294 after adjustments

Variable	Coefficient	Std. Error	t-Statistic	Prob.
DCLOSE(-1)	-0.874061	0.038217	-22.87128	0.0000
D(DCLOSE(-1))	-0.078446	0.027747	-2.827164	0.0048
C	-0.001947	0.027432	-0.070988	0.9434

R-squared	0.477443	Mean dependent var	-0.000440
Adjusted R-squared	0.476634	S.D. dependent var	1.364002
S.E. of regression	0.986774	Akaike info criterion	2.813565
Sum squared resid	1257.077	Schwarz criterion	2.825541
Log likelihood	-1817.377	F-statistic	589.7725
Durbin-Watson stat	2.002758	Prob(F-statistic)	0.000000

Path = c:\users\adebiyi\documents | DB = none | WF = ubao

Figure 3.26: ADF unit root test for DCLOSE of Zenith bank stock index.

Table 3.3 shows the different parameters of autoregressive (p) and moving average (q) of the ARIMA model in order to get the best fitted model. ARIMA (1, 0, 1) is relatively the best model as indicated in figure 3.27.

Dependent Variable: CLOSE
Method: Least Squares
Date: 04/09/11 Time: 00:58
Sample (adjusted): 1/04/2006 2/25/2011
Included observations: 1296 after adjustments
Convergence achieved after 7 iterations
Backcast: 1/03/2006

Variable	Coefficient	Std. Error	t-Statistic	Prob.
C	22.75593	14.43034	1.576951	0.1151
AR(1)	0.998006	0.001802	553.9323	0.0000
MA(1)	0.045431	0.027836	1.632089	0.1029

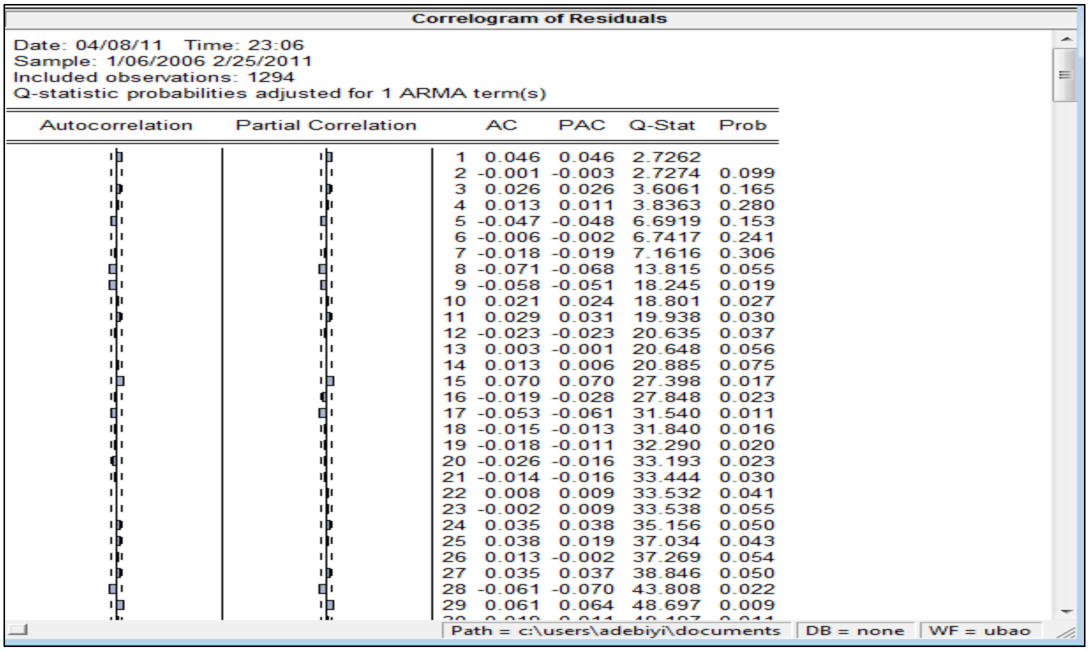
R-squared	0.996174	Mean dependent var	23.76762
Adjusted R-squared	0.996168	S.D. dependent var	15.97357
S.E. of regression	0.988780	Akaike info criterion	2.817621
Sum squared resid	1264.147	Schwarz criterion	2.829582
Log likelihood	-1822.819	F-statistic	168337.2
Durbin-Watson stat	1.992613	Prob(F-statistic)	0.000000

Inverted AR Roots	1.00
Inverted MA Roots	-.05

Figure 3.27: ARIMA (1, 0, 0) estimation output with DCLOSE of UBA bank index.

Figure 3.28 is the correlogram of residual of the seies. From the figure it is obvious there is no significant spike of ACFs and PACFs. This means that the residual of this selected

ARIMA model are white noise. There is no other significant patterns left in the time series and there is no need for further consideration of another AR(p) and MA(q).


Correlogram of Residuals

Date: 04/08/11 Time: 23:06
Sample: 1/06/2006 2/25/2011
Included observations: 1294
Q-statistic probabilities adjusted for 1 ARMA term(s)

Autocorrelation	Partial Correlation		AC	PAC	Q-Stat	Prob
		1	0.046	0.046	2.7262	
		2	-0.001	-0.003	2.7274	0.099
		3	0.026	0.026	3.6061	0.165
		4	0.013	0.011	3.8363	0.280
		5	-0.047	-0.048	6.6919	0.153
		6	-0.006	-0.002	6.7417	0.241
		7	-0.018	-0.019	7.1616	0.306
		8	-0.071	-0.068	13.815	0.055
		9	-0.058	-0.051	18.245	0.019
		10	0.021	0.024	18.801	0.027
		11	0.029	0.031	19.938	0.030
		12	-0.023	-0.023	20.635	0.037
		13	0.003	-0.001	20.648	0.056
		14	0.013	0.006	20.885	0.075
		15	0.070	0.070	27.398	0.017
		16	-0.019	-0.028	27.848	0.023
		17	-0.053	-0.061	31.540	0.011
		18	-0.015	-0.013	31.840	0.016
		19	-0.018	-0.011	32.290	0.020
		20	-0.026	-0.016	33.193	0.023
		21	-0.014	-0.016	33.444	0.030
		22	0.008	0.009	33.532	0.041
		23	-0.002	0.009	33.538	0.055
		24	0.035	0.038	35.156	0.050
		25	0.038	0.019	37.034	0.043
		26	0.013	-0.002	37.269	0.054
		27	0.035	0.037	38.846	0.050
		28	-0.061	-0.070	43.808	0.022
		29	0.061	0.064	48.697	0.009

Path = c:\users\adebiyi\documents DB = none WF = ubao

Figure 3.28: Correlogram of residuals of the UBA bank stock index.

In forecasting form, the best model selected can be expressed as follows:

$$Y_t = \phi_1 Y_{t-1} + \theta_0 + \varepsilon_t \qquad (3.16)$$

Where, $\varepsilon_t = Y_t - \hat{Y}_t$ (i.e., the difference between the actual value of the series and the forecast value)

Table 3.4: Statistical results of different ARIMA parameters for UBA bank Stock Index

ARIMA	BIC	Adjusted R^2	SEE
(1, 0, 0)	**2.8257**	**0.9961**	**0.9892**
(1, 0, 1)	2.8289	0.9961	0.9884
(2, 0, 0)	3.5695	0.9919	1.4348
(0, 0, 1)	7.0891	0.7273	8.3385
(0, 0, 2)	7.1035	0.7234	8.3987
(1, 1, 0)	2.8254	0.0018	0.9890
(0, 1, 0)	2.8217	0.0000	0.9896
(0, 1, 1)	2.8250	0.0015	0.9898
(1, 1,2)	2.8251	0.0069	0.9865
(2, 1, 0)	2.8222	0.0057	0.9875
(2, 1, 2)	2.8277	0.0051	0.9878

3.4 STEPS IN DESIGNING ANN FORECASTING MODEL

There are eight-step design methodologies for designing a neural network forecasting model as indicated in table 3.5 by Kaastra and Boyd, 1996. The subsection that follows gives a brief description of each of the steps.

Table 3.5: Eight steps in designing a neural network forecasting model

Step1: Variable selection
Step2: Data collection
Step3: Data preprocessing
Step4: Training, testing, and validation sets
Step5: Neural network design
- *Number of hidden layers*
- *Number of hidden neurons*
- *Number of output neurons*
- *Transfer functions*

Step6: Evaluation criteria
Step 7: Neural network training
- *Number of training iterations*
- *Learning rate and momentum*

Step 8: Implementation

3.4.1 Step 1: Variable Selection

Success in designing a neural network depends largely on a clear understanding of the problem. Knowing the right input variables is critical to the outcome of any predictive model. For instance, economic theory can assist researcher in finance in choosing variables that are important predictors.

3.4.2 **Step 2: Data Collection**

It is important for researchers to consider the cost and availability of data for the variables chosen in the previous step. Technical data are readily available from many vendors with reasonable cost. However, fundamental and expert opinion data are difficult to obtain. Data collected from vendors should be checked to ensure it is error free and of high quality data.

3.4.3 **Step 3: Data Preprocessing**

Data preprocessing simply means analyzing and transforming the input and output variables to minimize noise, highlight important relationships, detect trends and fatten the distribution of the variable to assist the neural network in learning important relevant patterns.

3.4.4 **Step 4: Training, Testing and Validation**

It is common practice to divide the data collected into three distinct set called the training, testing and validation. Usually, the training set is the largest set and is used by the neural network to learn patterns present in the data. The testing set, ranging in size from 10% to 30% of the training set, is used to evaluate the generalization ability of the trained network. The researcher will select the network(s) which perform best on the testing set. A final check on the performance of the trained network is done with validation set.

3.4.5 Step 5: Neural Network Design

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

3.4.6 Step 6: Evaluation

The most common error function minimized in neural network is the sum of squared errors. The importance of evaluation is to determine accuracy level of the predictive model.

3.4.7 Step 7: Neural Network Training

Training a neural network to learn patterns in the data involves iteratively presenting it with examples of the correct known answers. The objective of training is to find the set of weights between neurons that determine the global minimum of the error function. Unless the model is overfitted, this set of weights should provide good generalization. In training a neural network, the number of training iteration is determined by the researcher. Also the learning rate and momentum should not be too small or big. If the learning rate is set too high, the algorithm can oscillate and become unstable. If the learning rate is too small, the algorithm takes too long time to converge.

3.4.8 Step 8: Implementation

Most neural network software vendors provide the means by which trained networks can be implemented either in the neural network program itself or as an executable file (Kaastra and Boyd, 1996).

3.5 ANN MODEL DEVELOPMENTS

The data used in developing ANN models are termed technical, fundamental and expert opinion respectively. The technical and fundamental data were sourced from LSE and NSE. The expert opinion was obtained through questionnaire from stock brokers and financial experts though it is not easily obtainable. The same company data used in ARIMA model selection was also used in ANN model construction. Zenith and UBA banks from NSE and Dell and Nokia Corporation from LSE. In this research the closing price is chosen to represent the price of the index to be predicted. The choice is because the closing price reflects all the activities of the index in a day.

3.5.1 Backpropagation Neural Networks

Backpropagation (BP) is the generalization of the Widrow-Hoff learning rule to multiple-layer networks and nonlinear differentiable transfer function. BP consists of a collection of inputs and processing units known as neurons or nodes. The neurons in each layer are fully interconnected by connection strengths called weights which, along with the network architecture, store the knowledge of a trained network.

BP networks are a class of feedforward neural networks with supervised learning rules. Feedforward refers to the direction of information flow from the input to the output layer. Inputs are passed through the neural network once to determine the output. Supervised training is the process of comparing each of the network's forecasts with the known answer and adjusting the weights based on the resulting forecast error to minimize the error function. The BP network is the most common multi-layer network estimated to be used in 80% of all application and the most widely used in financial time series forecasting (Kasstra and Boyd, 1996). According to Kaastra and Boyd, table 3.6 presents the common parameters in designing a backpropagation neural networks for financial forecasting modeling.

Table 3.6: Common parameters in designing BP networks for financial predictive model

Data Preprocessing	
	- Frequency of data – daily, weekly, monthly, quarterly
	- Type of data – technical, fundamental
	- Method of data sampling
	- Method of data scaling – maximum/minimum, mean/standard deviation
Training	
	- Learning rate
	- Momentum term
	- Training tolerance
	- Epoch size
	- Learning rate limit
	- Number of times to randomize weights
	- Size of training, testing and validation sets
Topology	
	- Number of input neurons
	- Number of hidden layers
	- Number of hidden neurons in each layer
	- Number of output neurons
	- Transfer function for each neuron
	- Error function

3.5.2 Multi-layer Perceptron Model

Multi-layer perceptrons (MLPs) model are feedforward neural networks model trained with backpropagation algorithm. This model usually consists of three layers, input layer, hidden layer and output layer. According to Hornik et al. (1989), Cybenko (1989) and Hornik et al. (1990) three layered feedforward neural network models with nonlinear function in the hidden layers could approximate any continuous function well if there were sufficient hidden nodes in the hidden layer. This study utilised three-layer (one hidden layer) multilayer perceptron models (feedforward neural network models), as these models are mathematically proved to be universal approximator for any function. Multi-layer perceptrons model was chosen in this study because it is the most common network architecture used for financial neural networks.

3.5.3 **Input Variables**

The basic input data includes: raw data such as the daily open, high, low and close prices, and trading volumes which formed the technical variables in table 3.7. Table 3.8 consists of fundamental variables while the market expert opinion variables are listed in table 3.9. The form in which the expert's opinions were captured on stock indices is described in figure 3.29.

Table 3.7: Stock Variables (Technical Indicators)

Variable	**Description**
O	Opening price of a stock for a specific trading day
C	Closing price of a stock for a specific trading day
V	Stock transactions volume (Buy/Sell)
H	Highest stock price within a specified time interval (day, month etc.)
L	Lowest stock price within a specified time interval (day, month etc.)

Table 3.8: Stock Variables (Fundamental Indicators)

Variable	**Description**
P/E	Price per annual earning
ROA	Return on Asset
ROE	Return on Common Equity

Table 3.9: Possible Stock Price Influence Factors (Experts Opinion)

Variable	**Description**
M	Management Quality
F	Investors Confidence
I	Inflation Rate
B	Business Sector Growth
G	Government Policy

Management Quality (MQ):	0-2 = Poor, 3-5 = Normal, 6-8 = Good, 9-10 = Very Good
Investor Confidence (IC):	0-2 = Low, 3-5 = Normal, 6-8 = High, 9-10 = Very High
Inflation Rate (IR):	0-2 = Low, 3-5 = Normal, 6-8 = High, 9-10 = Very High
Business Sector Growth (BSG):	0-2 = Low, 3-5 = Normal, 6-8 = High, 9-10 = Very High
Government Policy (GP):	0-2 = Bad, 3-5 = Normal, 6-8 = Good, 9-10 = Very Good

Figure 3.29: Format for capturing expert's opinion

3.5.4 Data Preprocessing

Data selection and pre-processing are crucial step in any modeling effort. In order to generalize the new predictive model. The stock data are divided into two sets: the training and testing data which are scaled to the range of (0, 1) using min-max normalization equation (3.17).

$$x_{ni} = \frac{x_i - x_{\min}}{x_{\max} - x_{\min}} \tag{3.17}$$

where x_i is the real-world stock value, x_{ni} is the scaled input value of the real-world stock value x_i, $x_{\min}$ and $x_{\max}$ are the minimum and maximum values of the unscaled dataset. The network predicted values, which are in the range (0, 1) are transformed to real-world values with the following equation:

$$x_i = x_{ni}(x_{\max} - x_{\min}) + x_{\min} \tag{3.18}$$

3.5.5 The Proposed Predictive Model

In this research work, a predictive model based on technical and fundamental indicators, and experts' opinions using neural network architecture is proposed. The aim is to explore more accurate results in stock price prediction. Based on the idea behind technical analysis of investment trading, it is assumed that the behaviour of stock market in the future could be predicted with previous information given in the history (Li, 2005). Therefore, there exists a function in equation (3.19)

$$p(t+1) = f(p_{t-k}, \ldots, p_t; x_{t-1}, \ldots, x_t; y_{t-m}, \ldots, y_t; \ldots) \tag{3.19}$$

where p is the stock price, x and y are the other influence factors such as daily highest price, daily lowest price, experts opinion etc. Artificial neural network is used in order to model the nonlinear data. Thus,

$$y_t = w_0 + \sum_{j=1}^{q} w_j . g\left(w_{0j} + \sum_{i=1}^{p} w_{ij} . y_{t-1} \right) + \varepsilon_t \qquad (3.20)$$

where, $w_{ij}\,(i = 0,1,2,..., p,\ j = 1,2,...,q)$ and $w_j\,(j = 0,1,2,...,q)$ are model parameter often called connection weights; p is the number of input nodes; and q is the number hidden nodes. The study used three-layer (one hidden layer) multilayer perceptron models (a feedforward neural network model) trained with backpropagation algorithm. The activation function that was used is sigmoid function. Figure 3.30 depicts the neural network architecture used in this study.

In this study, three different models for the empirical investigation and validation of the proposed model were used as indicated in table 3.10. The models were created with ANN. The inputs to the first model contained purely technical analysis variables of historical stock data. The second and third models contained hybrid of market indicators. The inputs to the second model consist of technical and fundamental analysis variables while the inputs to the third model combined both the technical and fundamental variables with the market experts' opinion variables. The fundamental variables consist of financial ratios such as price per annual earning (P/E), return on asset (ROA), and return on equity (ROE). P/E is equal to the market price per share of stock divided by the earning per share. The ROA measures a firm's performance in using the asset to generate income. ROE measures the rate of return earned on the common stockholders' investment. Both the ROA and ROE are used to determined management effectiveness of a firm. The experts opinion consist of inflation rate (I), management quality (M), investors confidence(F), government policy (G) and political factor (T) etc. For the hybridized approach 18 input variables was identified and used to train the network comprising both technical, fundamental variables, and experts' opinion variables as indicated in model 3 of table 3.10. Table 3.11 gives the description of input variable used in this study.

Table 3.10: The Input and Output Parameters of the Models used in this Study

Model	Technique	Input	Output
1	ANN	$O_{i-1}, O_{i-2}, H_{i-1}, H_{i-2}, L_{i-1}, L_{i-2}, C_{i-1}, C_{i-2}, V_{i-1}, V_{i-2}$	y(t+1)
2	ANN	$O_{i-1}, O_{i-2}, H_{i-1}, H_{i-2}, L_{i-1}, L_{i-2}, C_{i-1}, C_{i-2}, V_{i-1}, V_{i-2}, P_{i-1}, R_{i-1}, E_{i-1}$	y(t+1)
3	ANN	$O_{i-1}, O_{i-2}, H_{i-1}, H_{i-2}, L_{i-1}, L_{i-2}, C_{i-1}, C_{i-2}, V_{i-1}, V_{i-2}, P_{i-1}, R_{i-1}, E_{i-1}$, M_{i-1}, $F_{i-1}, I_{i-1}, G_{i-1}, B_{i-1}$	y(t+1)

Table 3.11: Description of Input Variables used in this study

Technical Analysis Variables		Fundamental and Expert Opinion Variables	
O_{i-1}	the opening price of day *i-1*	P_{i-1}	the price per annual earning of year *i-1*
O_{i-2}	the opening price of day *i-2*	R_{i-1}	return on asset of trading year *i-1*
H_{i-1}	the daily high price of day *i-1*	E_{i-1}	return on equity of trading year *i-1*
H_{i-2}	the daily high price of day *i-2*	M_{i-1}	management quality as at trading day *i-1*
L_{i-1}	the daily low price of day *i-1*	F_{i-1}	investors confidence as at trading day *i-1*
L_{i-2}	the daily low price of day *i-2*	I_{i-1}	inflation rate as at trading day *i-1*
C_{i-1}	the closing price of day *i-1*	B_{i-1}	business sector growth as at trading day *i-1*
C_{i-1}	the closing price of day *i-2*	G_{i-1}	government policy as at trading day *i-1*
V_{i-1}	the trading volume of day *i-1*		
V_{i-2}	the trading volume of day *i-2*		

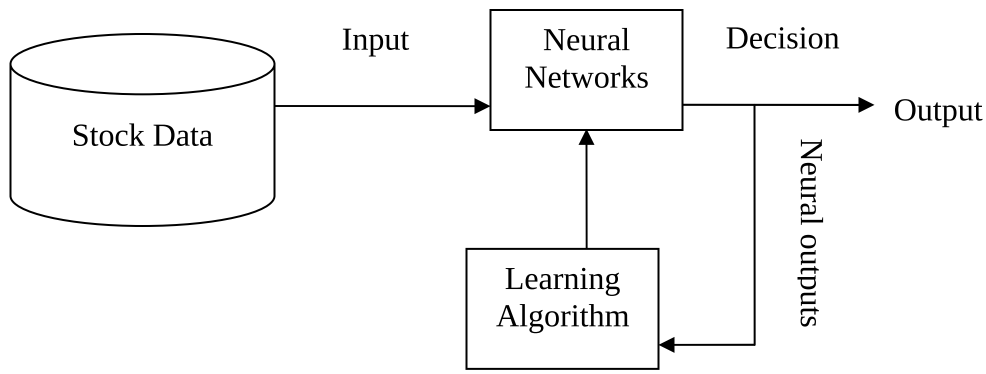


Figure 3.30: Neural Network Architecture for Stock Prediction

We experimented with the different neural network model configurations to determine the best performance in each of the models using Matlab Neural Network Tools Box version 7. The algorithm of the ANN experiment used in this study is shown in figure 3.31 below. Training data and testing data was carefully selected. In training the network models, the test data were not used. Each model with the different network structures was trained with 1000 epochs, 2000 epochs and 5000 epochs respectively. The mean squared error (MSE) returned for each training session of the different network structure models was noted and recorded.

(1) Define the output
(2) Choose the appropriate network architecture and algorithm. Multi-layer perceptron model trained with backpropagation algorithm was primarily chosen.
(3) Determine the input data and preprocess if necessary.
(4) Choose appropriate learning function.
(5) Choose the appropriate network structure.
(6) Perform the training and testing for each cycle.
(7) If the network produced acceptable results for all cycles, perform step 8 else perform step 5 to try other appropriate network structures else perform step 4 to try with other learning algorithm else perform step 3 to add or remove from input set. Otherwise, go back to step 2 to try different neural network architecture.
(8) Finish - record the results.

Figure 3.31: Algorithm for ANN predictive model.

3.5.5.1 **ANN Model Construction for Dell Stock Index**

Creating the neural network predictive model for Dell stock index involves the following activities:

Creating a network topology

- Select the number of input neurons (in this case 10 inputs for model 1, 13 inputs for model 2, and 18 inputs for model 3)
- Select the number of hidden layer (one hidden layer selected)
- Select the number of hidden neurons in the hidden layer
- Select number of output neuron (one was selected)

Training the network

- Select the network type/training algorithm (feed-forward backpropagation algorithm)
- Input the training and target data
- Select the training function (TRAINGDM)
- Select adaptation learning function (LEARNGDM)
- Select the performance function (MSE)
- Select the transfer function (TANSIG was selected)

Training parameter

- Learning rate = 0.01
- Momentum term = 0.9
- Epoch size = 1000, 2000, 5000
- Goal = 0
- Show = 25

Testing the network

Testing the network is useful for estimating a network's ability to generalize. The trained network generalization ability is determined by testing the model using the testing data set.

Figures 3.32 – 3.34 is the graph of network training showing the best performance in each of the model in the different training sessions. The network structure that returns the smallest mean square error in each of the model was adjudged the best model that can give the best accurate prediction.

Similarly, tables 3.12 – 3.14 is the outcome of the various training session in each of the model. It was observed in most cases that the best model was obtained when the network was well trained.

Table 3.12: Statistical performance of Model 1 of Dell Stock index

	MSE			
	Network Structure	**1000 Epochs**	**2000 Epochs**	**5000 Epochs**
Model 1	10-10-1	0.129054	0.112363	0.093539
	10-11-1	0.144086	0.108245	0.090521
	10-12-1	0.125668	0.099301	0.088157
	10-13-1	0.148646	0.115732	0.092649
	10-14-1	0.141474	0.099241	0.085206
	10-15-1	0.118226	0.096651	0.083664
	10-16-1	0.116773	0.099222	0.080534
	10-17-1	**0.097826**	**0.085111**	**0.071589**
	10-18-1	0.119719	0.093576	0.079150

Bold characters indicate the best results for each of epoch session

Table 3.13: Statistical performance of Model 2 of Dell Stock index

	MSE			
	Network Structure	**1000 Epochs**	**2000 Epochs**	**5000 Epochs**
Model 2	13-13-1	0.123184	0.107412	0.090236
	13-14-1	0.130018	0.095150	0.075755
	13-15-1	0.179753	0.121711	0.090089
	13-16-1	0.118892	0.093771	0.076748
	13-17-1	0.128121	0.103660	0.087496
	13-18-1	0.123907	0.097827	0.079230
	13-19-1	0.115329	**0.093167**	**0.067901**
	13-20-1	0.118568	0.101955	0.085272
	13-21-1	**0.113894**	0.094207	0.077037

Bold characters indicate the best results for each of epoch session

Table 3.14: Statistical performance of Model 3 of Dell Stock index

	MSE			
	Network Structure	**1000 Epochs**	**2000 Epochs**	**5000 Epochs**
Model 3	18-18-1	0.125340	0.101465	0.083380
	18-19-1	0.153357	0.143036	0.144930
	18-20-1	0.125132	0.121522	0.131991
	18-21-1	0.110519	0.090850	0.072907
	18-22-1	0.115685	0.093702	0.069479
	18-23-1	**0.106685**	0.090579	0.070017
	18-24-1	0.129842	0.107310	0.078529
	18-25-1	0.112151	0.090779	0.070529
	18-26-1	0.112049	**0.088974**	**0.061821**

Bold characters indicate the best results for each of epoch session.

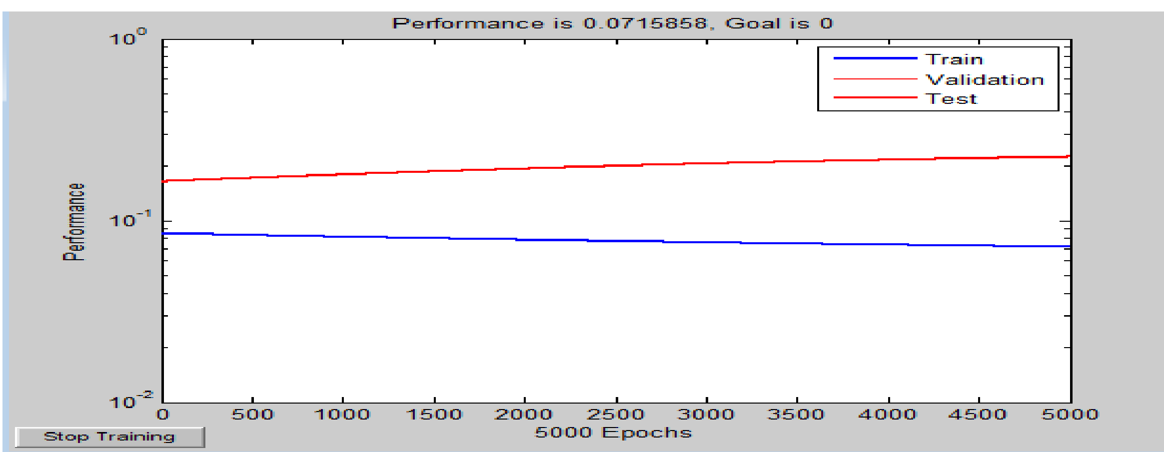


Figure 3.32: Graph of best result achieved in network training of Model 1 of Dell index

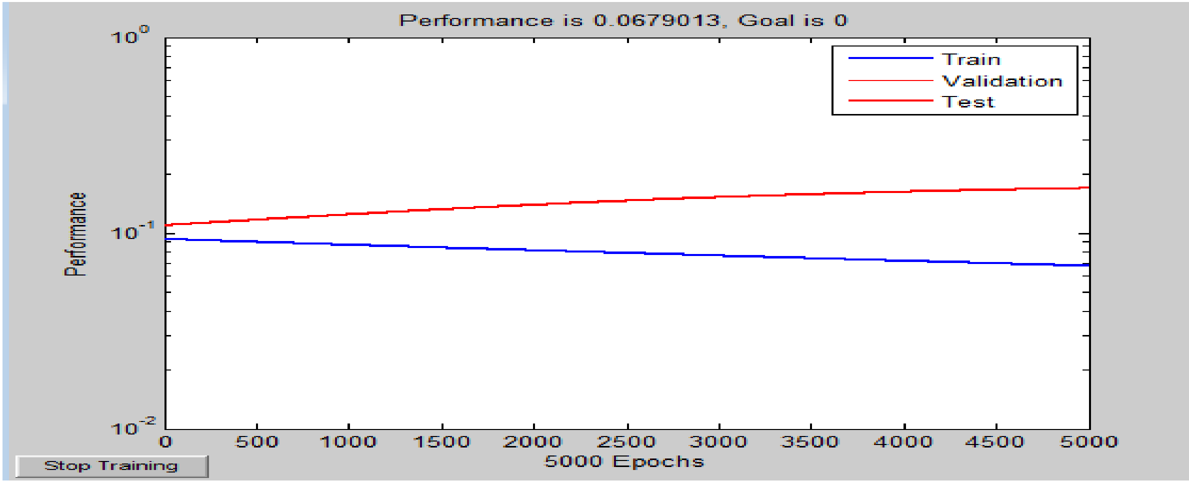


Figure 3.33: Graph best result achieved in network training of Model 2 of Dell index

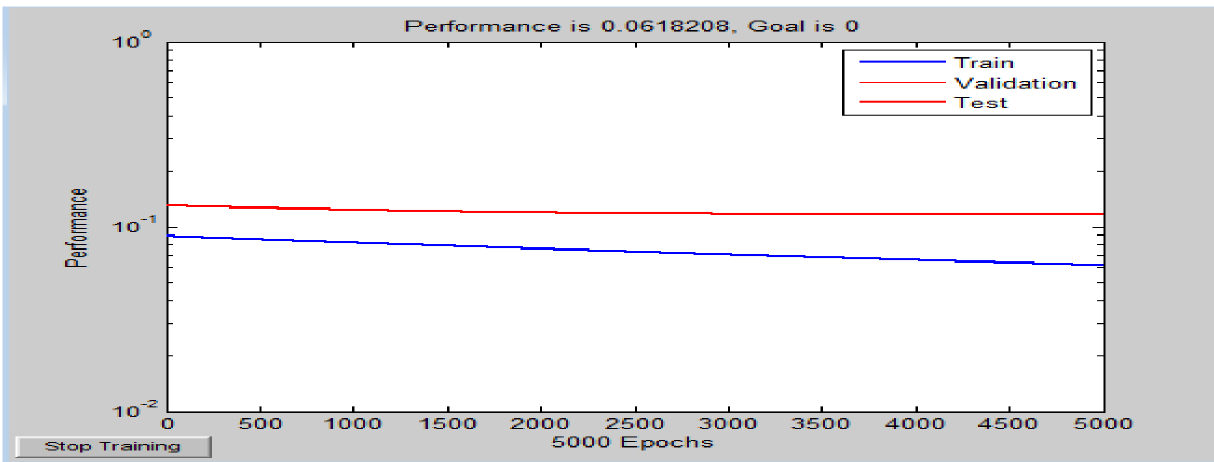


Figure 3.34: Graph of best result achieved in network training of Model 3 of Dell index

3.5.5.2 **ANN Model Construction for Nokia Stock Index**

We used the same parameters and procedures earlier mentioned to create the neural network predictive model for Nokia stock index. Figures 3.35 – 3.37 are the graph of network training showing the best performance in each of the models in the different training sessions. The network structure that returns the smallest mean square error in each of the model was adjudged the best model that can give best accurate prediction.

Similarly, tables 3.15 – 3.17 are the outcomes of the various training sessions in each of the models.

Table 3.15: Statistical performance of Model 1 of Nokia Stock index

	MSE			
	Network Structure	**1000 Epochs**	**2000 Epochs**	**5000 Epochs**
Model 1	10-10-1	0.131413	0.085673	0.065958
	10-11-1	0.130266	0.100147	0.074376
	10-12-1	**0.104179**	**0.076013**	0.055319
	10-13-1	0.148378	0.123140	0.094993
	10-14-1	0.113599	0.077468	**0.052256**
	10-15-1	0.120586	0.094881	0.063535
	10-16-1	0.112569	0.087723	0.590738
	10-17-1	0.121481	0.099204	0.064114
	10-18-1	0.129727	0.088369	0.074229

Bold characters indicate the best results for each of epoch session

Table 3.16: Statistical performance of Model 2 of Nokia Stock index

	MSE			
	Network Structure	**1000 Epochs**	**2000 Epochs**	**5000 Epochs**
Model 2	13-13-1	0.142468	0.111148	0.083957
	13-14-1	0.125704	0.085771	**0.048449**
	13-15-1	0.130336	0.092512	0.063329
	13-16-1	0.120522	**0.070893**	0.051680
	13-17-1	2.328160	0.092799	0.053544
	13-18-1	0.132760	0.109483	0.068056
	13-19-1	**0.110694**	0.081275	0.053134
	13-20-1	4.077420	0.110826	0.067431
	13-21-1	0.145350	0.100590	0.052527

Bold characters indicate the best results for each of epoch session

Table 3.17: Statistical performance of Model 3 of Nokia Stock index

	MSE			
	Network Structure	**1000 Epochs**	**2000 Epochs**	**5000 Epochs**
Model 3	18-18-1	0.109847	**0.048198**	0.041309
	18-19-1	0.131226	0.102727	0.066412
	18-20-1	**0.109616**	0.073874	0.052122
	18-21-1	0.136433	0.070053	**0.038898**
	18-22-1	0.130056	0.096053	0.054354
	18-23-1	0.118424	0.083801	0.049239
	18-24-1	0.170919	0.083517	0.050821
	18-25-1	4.497720	4.374340	4.347050
	18-26-1	0.222150	0.202463	0.062659

Bold characters indicate the best results for each of epoch session.

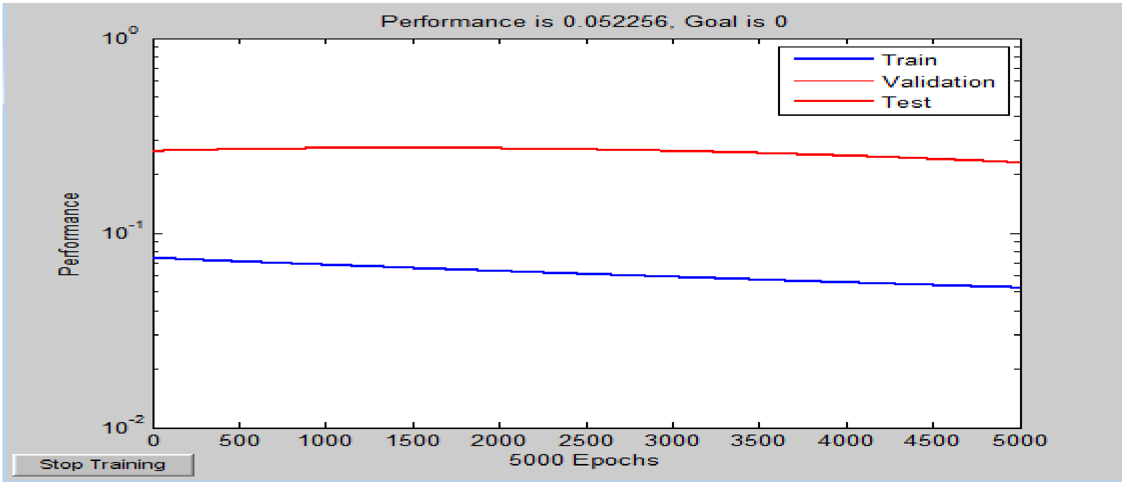


Figure 3.35: Graph of best result achieved in network training of Model 1 of Nokia index

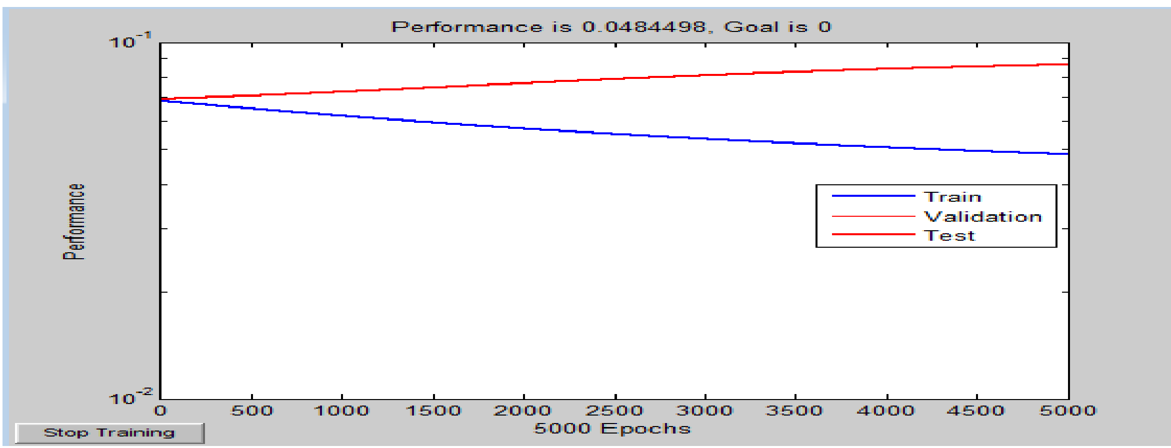


Figure 3.36: Graph of best result achieved in network training of Model 2 of Nokia index

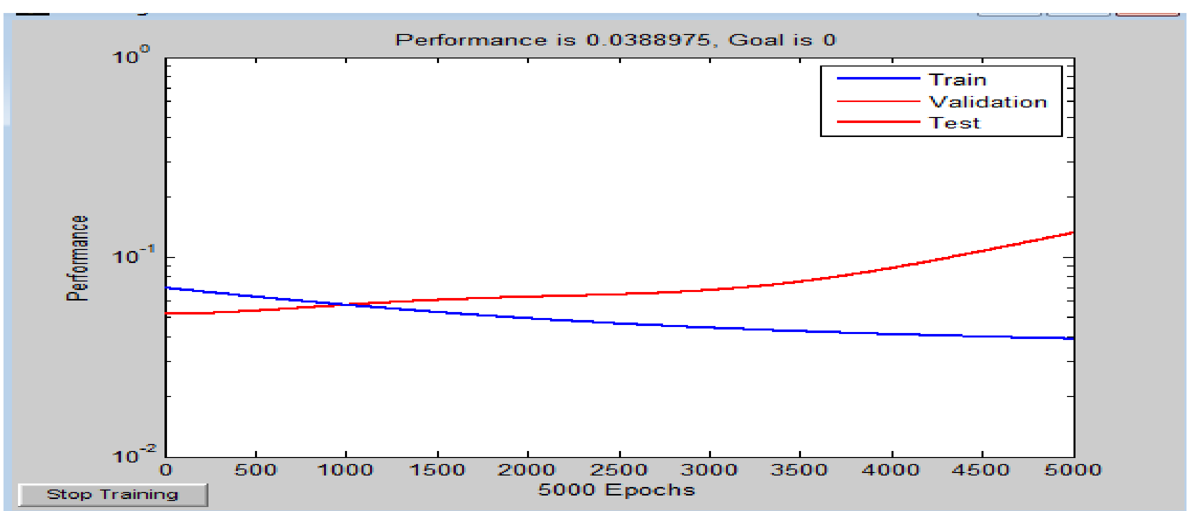


Figure 3.37: Graph of best result achieved in network training of Model 3 of Nokia index

3.5.5.3 **ANN Model Construction for Zenith Bank Stock Index**

We carried out same experiments in creating the neural network predictive model for Zenith bank stock index as previously done for other stock indexes. Figures 3.38 – 3.40 are the graphs of network training showing the best performance in each of the model in the different training sessions. The network structure that returns the smallest mean square error in each of the model was adjudged the best model that can give best accurate prediction. Similarly, tables 3.18 – 3.20 are the outcomes of the various training session in each of the model.

Table 3.18: Statistical performance of Model 1 of Zenith Bank Stock index

	MSE			
	Network Structure	**1000 Epochs**	**2000 Epochs**	**5000 Epochs**
Model 1	10-10-1	4.108300	4.099800	4.098230
	10-11-1	0.035962	0.019059	0.010922
	10-12-1	**0.018442**	**0.009902**	**0.005750**
	10-13-1	4.102440	4.100620	4.098390
	10-14-1	0.027818	0.018080	0.009482
	10-15-1	4.028170	0.020783	0.011648
	10-16-1	4.110630	4.108430	0.013559
	10-17-1	4.126330	4.126040	4.063250
	10-18-1	4.111820	4.111760	4.103740

Bold characters indicate the best results for each of epoch session

Table 3.19: Statistical performance of Model 2 of Zenith Bank Stock index

	MSE			
	Network Structure	**1000 Epochs**	**2000 Epochs**	**5000 Epochs**
Model 2	13-13-1	4.121220	4.114060	4.098280
	13-14-1	0.035358	0.018686	0.008857
	13-15-1	4.111620	4.111430	4.106180
	13-16-1	0.031099	0.019543	0.010096
	13-17-1	4.109700	4.098990	4.098160
	13-18-1	4.100310	4.099320	4.098830
	13-19-1	**0.019570**	**0.008205**	**0.005829**
	13-20-1	4.099930	4.099340	4.098750
	13-21-1	4.116100	4.098810	4.098320

Bold characters indicate the best results for each of epoch session

Table 3.20: Statistical performance of Model 3 of Zenith Bank Stock index

	MSE			
	Network Structure	**1000 Epochs**	**2000 Epochs**	**5000 Epochs**
Model 3	18-18-1	4.120570	0.036615	0.005744
	18-19-1	0.029132	0.011933	0.005900
	18-20-1	0.015882	**0.007599**	**0.004163**
	18-21-1	0.029142	0.012597	0.006514
	18-22-1	**0.015248**	0.009414	0.007592
	18-23-1	4.047870	4.047670	0.019429
	18-24-1	4.110580	4.099050	4.097920
	18-25-1	0.021306	0.010797	0.007444
	18-26-1	4.102400	4.100080	4.098580

Bold characters indicate the best results for each of epoch session

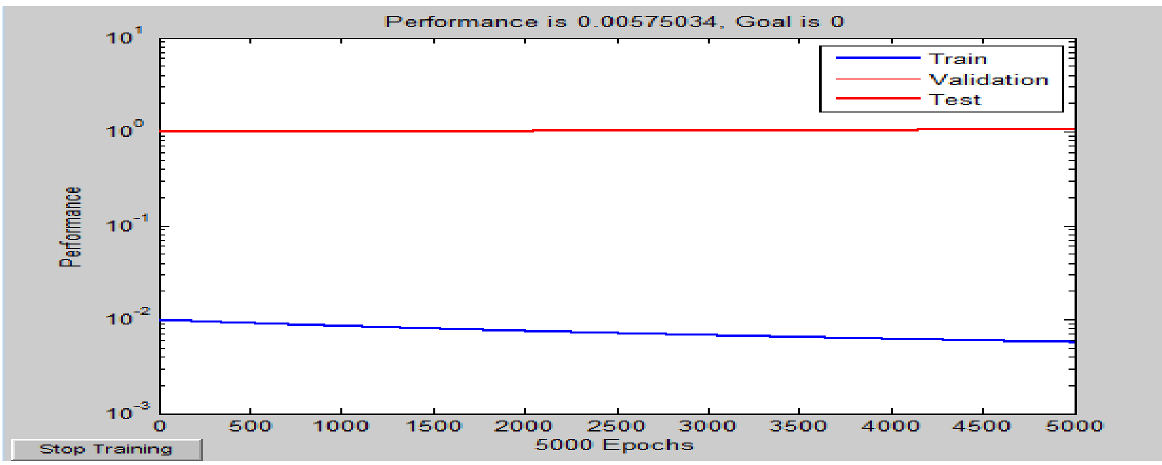


Figure 3.38: Graph of best result achieved in network training of Model 1 of Zenith index

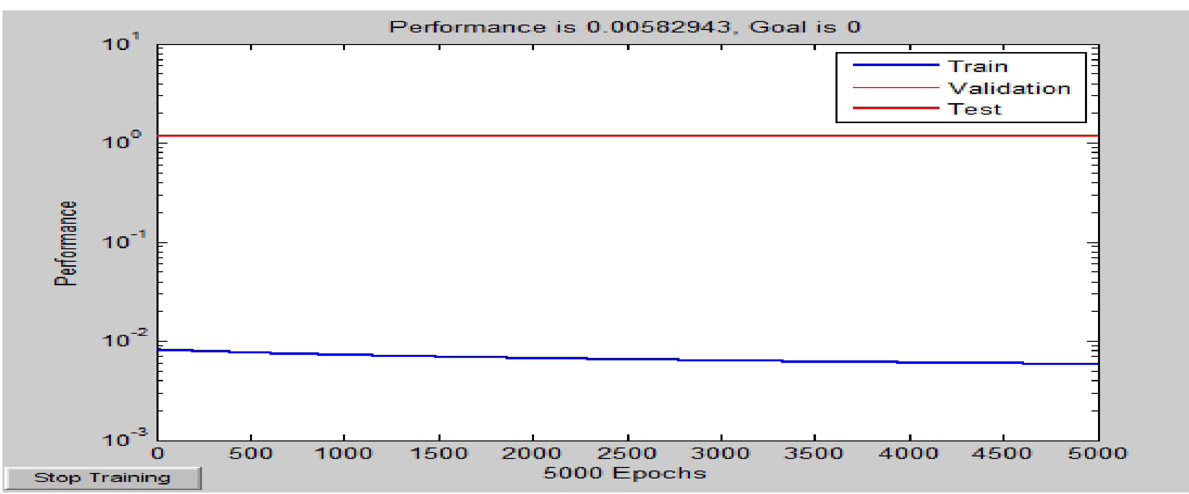


Figure 3.39: Graph of best result achieved in network training of Model 2 of Zenith index

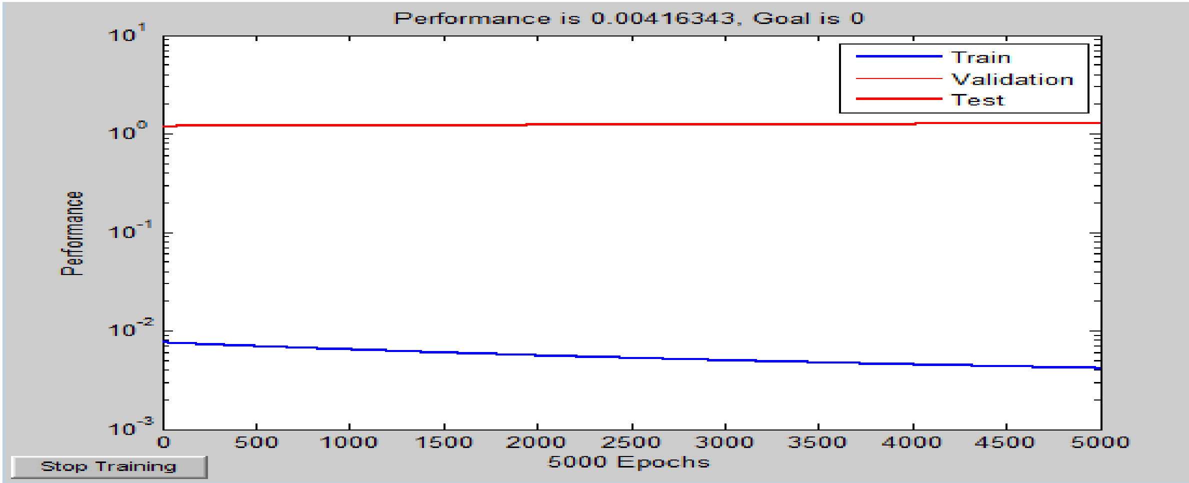


Figure 3.40: Graph of best result achieved in network training of Model 2 of Zenith index

3.5.5.4 **ANN Model Construction for UBA Bank Stock Index**

Similarly, we carried out experiments in order to create neural network predictive model for UBA bank stock index.

Figures 3.41 – 3.43 are the graphs of network training showing the best performance in each of the models in the different training sessions. The network structure that returns the smallest mean square error in each of the model was adjudged the best model that can give best accurate prediction.

Similarly, tables 3.21 – 3.23 are the outcomes of the various training session in each of the model. It was observed in most cases that the best model was obtained when the network was well trained.

Table 3.21: Statistical performance of Model 1 of UBA Bank Stock index

	MSE			
	Network Structure	**1000 Epochs**	**2000 Epochs**	**5000 Epochs**
Model 1	10-10-1	0.065771	0.039392	0.015526
	10-11-1	0.040859	0.018023	0.007149
	10-12-1	**0.022149**	**0.009799**	0.006476
	10-13-1	0.045920	0.029234	0.014437
	10-14-1	0.029664	0.018399	0.011376
	10-15-1	0.410990	0.030402	0.008053
	10-16-1	0.026901	0.011973	0.006532
	10-17-1	0.026850	0.012953	**0.006317**
	10-18-1	0.035467	0.014900	0.006513

Bold characters indicate the best results for each of epoch session

Table 3.22: Statistical performance of Model 2 of UBA Bank Stock index

	MSE			
	Network Structure	**1000 Epochs**	**2000 Epochs**	**5000 Epochs**
Model 2	13-13-1	0.045267	0.024678	0.012737
	13-14-1	0.049285	0.020317	0.009248
	13-15-1	0.049302	0.028288	0.014649
	13-16-1	0.033183	0.012358	0.005490
	13-17-1	**0.027930**	**0.011691**	**0.005297**
	13-18-1	0.028951	0.012218	0.006627
	13-19-1	0.033560	0.018832	0.009782
	13-20-1	0.035910	0.018103	0.008589
	13-21-1	0.037007	0.024169	0.014154

Bold characters indicate the best results for each of epoch session

Table 3.23: Statistical performance of Model 3 of UBA Bank Stock index

	MSE			
	Network Structure	**1000 Epochs**	**2000 Epochs**	**5000 Epochs**
Model 3	18-18-1	0.033091	0.012269	0.005957
	18-19-1	0.033951	0.019018	0.009853
	18-20-1	0.022090	0.012050	0.007072
	18-21-1	0.029914	0.014159	0.006764
	18-22-1	**0.020594**	0.008664	0.003818
	18-23-1	0.030647	0.013578	0.006147
	18-24-1	0.034620	**0.008276**	**0.003816**
	18-25-1	0.028813	0.011532	0.004925
	18-26-1	0.058627	0.017453	0.007322

Bold characters indicate the best results for each of epoch session

Figure 3.41: Graph of best result achieved in network training of Model 1 of UBA index

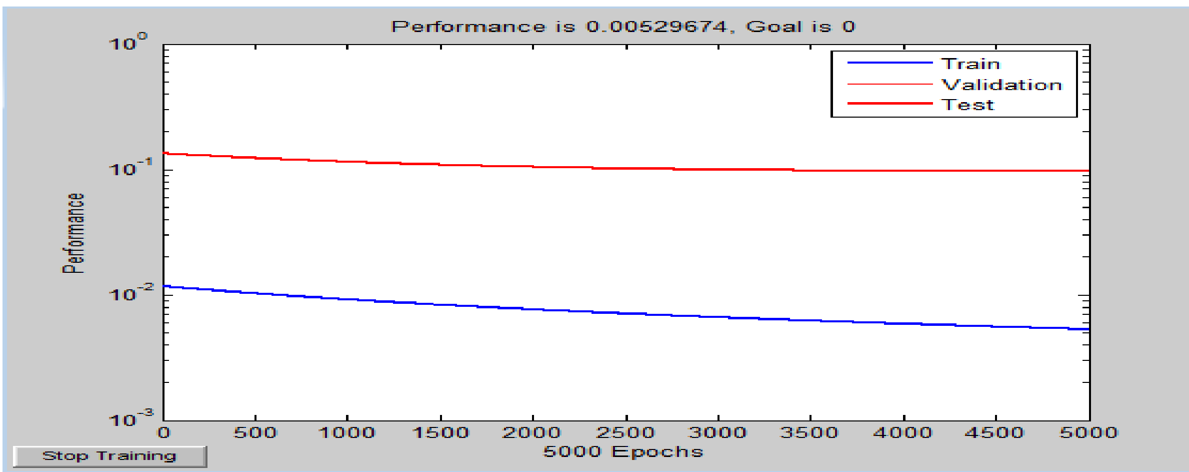


Figure 3.42: Graph of best result achieved in network training of Model 2 of UBA index

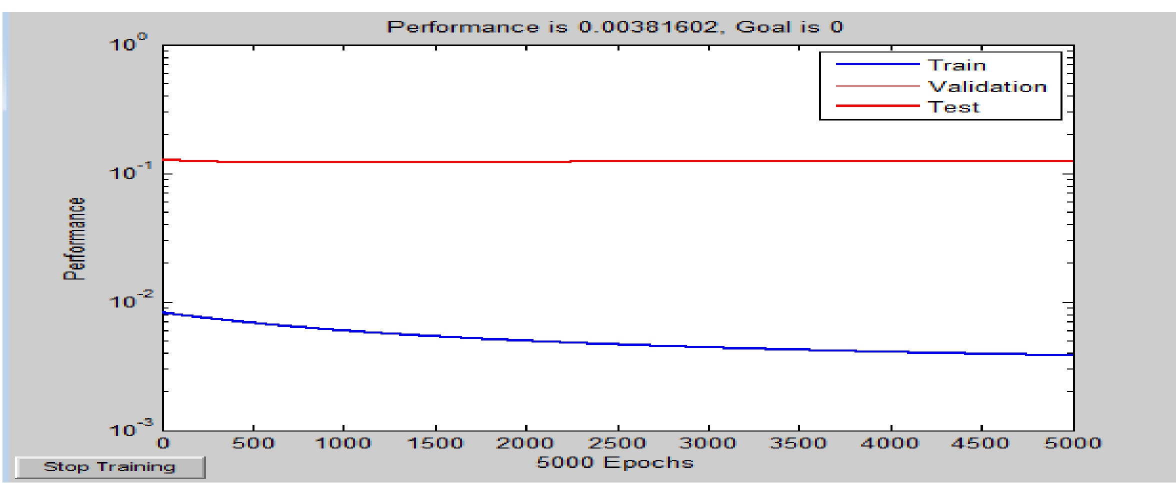


Figure 3.43: Graph of best result achieved in network training of Model 3 of UBA index

3.6 **Performance Measures**

An essential part of financial prediction is the evaluation of the prediction algorithm. Several performance measures are widely used in literature. This section gives an overview of the performance measurement tools used in this study.

3.6.1 **Confusion Matrix**

A confusion matrix (Kohavi and Provost, 1998) contains information about actual and predicted classifications done by a classification system. Performance of such systems is commonly evaluated using the data in the matrix. The following table shows the confusion matrix for a two class classifiers.

The entries in table 3.24 in the confusion matrix have the following meanings in the context of this study:

- a is the number of correct predictions that an instance is up,
- b is the number of incorrect predictions that an instance is down,
- c is the number of incorrect of predictions that an instance is up, and
- d is the number of correct predictions that an instance is down.

Table 3.24: A Confusion Matrix

		Predicted	
		Up	Down
Actual	Up	a	b
	Down	c	d

Several standard terms have been defined for the 2 class matrix:

The accuracy (AC) is the proportion of the total number of predictions that are correct. It is determined using the equation:

$$AC = \frac{a+d}{a+b+c+d} \tag{3.21}$$

The accuracy determined using the equation may not be an adequate performance measure when the number of negative cases is much greater than the number of positive cases. In financial forecasting, classification can be done by predicting whether the next day's price of the index is higher or lower than the previous day's price. This is done by doing a direct comparison between the two prices. This helps to measure the performance of an algorithm on its ability to predict between the two states (higher or lower) compared to the target. The results are presented in a confusion matrix.

3.6.2 Statistical Method

There are many different statistical methods existing in literature, the most common methods used in performance evaluation of financial modeling are sum of squared errors (SSE), mean absolute error (MAE), mean squared error (MSE), root mean squared error (RMSE), and mean absolute percentage error (MAPE).

However, the basic objective of the forecasting efforts is to beat the market, or in other words, gaining more returns than the average market return (Refenes, 1995). The statistical performance measures are providing a clue about the performance of the neural network models, but do not guarantee the profitability of the forecasts (Yao, *et al*., 1999).

Thus, in this study the performance of the neural network predictive models was analyzed by comparing the mean squared error of each network structure of different hidden neurons. The one that return the smallest error is regarded as the best model.

3.7 Tools for Model Implementation

The tools used in this study to implement the model developed are briefly described in the subsection below.

3.7.1 MATLAB

MATLAB® is a high-level technical computing language and interactive environment for algorithm development, data visualization, data analysis, and numeric computation. MATLAB can be used to solve technical computing problems faster than with traditional programming languages, such as C, C++, and Fortran.

MATLAB can be used in a wide range of applications, including signal and image processing, communications, control design, test and measurement, financial modelling and analysis, and computational biology. Add-on toolboxes (collections of special-purpose MATLAB functions are available separately) extend the MATLAB environment to solve particular classes of problems in these application areas.

MATLAB provides a number of features for documenting and sharing your work. MATLAB can integrate MATLAB code with other languages and applications, and distribute MATLAB algorithms and applications. Other remarkable features of MATLAB are enumerated below:

- High-level language for technical computing.
- Development environment for managing code, files, and data.
- Interactive tools for iterative exploration, design, and problem solving.

- Mathematical functions for linear algebra, statistics, Fourier analysis, filtering, optimization, and numerical integration.
- 2-D and 3-D graphics functions for visualizing data.
- Tools for building custom graphical user interfaces.
- Functions for integrating MATLAB based algorithms with external applications and languages, such as C, C++, Fortran, Java, COM, and Microsoft Excel.

3.7.2 **Eviews**

Eviews (Econometric Views) is a statistical package for windows that is used mainly for time series oriented econometric analysis. It was developed by Quantitative Micro Software (QMS). Eviews can be used for general statistical and econometric analyses, such as time series estimation and forecasting and cross-section and panel data analysis.

Eviews offers academic researchers, corporations, government agencies, and students access to powerful statistical, forecasting, and modeling tools.

CHAPTER FOUR

RESULTS AND DISCUSSION

4.1 INTRODUCTION

In this chapter, the results of both ARIMA modeling and artificial neural networks for stock price prediction are presented and discussed. As earlier mentioned four companies stock data were drawn from two countries stock exchange namely New York Stock Exchange (NYSE) and Nigeria Stock Exchange (NSE). The companies from NYSE are Dell Inc. and Nokia Inc. from IT sector and the companies from NSE are Zenith bank and UBA bank from financial sector. The forecast variable is the closing price. The evaluation of the predictive model for accurate prediction was done with confusion matrix and mean square error.

4.2 ARIMA MODEL RESULTS

As earlier mentioned, the following criteria are used in this study to determine the best ARIMA model for each stock index used.

- Relatively small of BIC (Bayesian or Schwarz Information Criterion which is measured by $n\log(SEE) + k\log(n)$)
- Relatively small of SEE (S.E. of regression)
- Relatively high of adjusted R^2
- Q-statistics and correlogram show that there is no significant pattern left in the ACFs and PACFs of the residuals, it means the residual of the selected model are white noise.

4.2.1 **Result of ARIMA Model for Dell Stock Price Prediction**

ARIMA (1, 0, 0) was selected as the best model based on the criteria listed in the previous section. The actual stock price and predicted values are presented in table 4.1 below and the figure 4.1 is the graph of predicted price against actual stock price to see the performance of the ARIMA model selected. From the predicted values, it was observed that a constant number is added to the subsequent values from the previous value and this account for the linear graph of the predicted values in the figure 4.1 below.

The model was evaluated in terms of ability to accurately predict whether the stock price will go up or down in the next trading day with confusion matrix. The accuracy of the model was found to be 65% as indicated the table 4.2.

Table 4.1: Sample of Empirical Results of ARIMA (1, 0, 0) of Dell Stock Index.

Sample Period	Actual Values	Predicted Values
1/3/2010	13.57	13.35
2/3/2010	13.68	13.46
3/3/2010	13.71	13.56
4/3/2010	13.67	13.67
5/3/2010	13.88	13.78
8/3/2010	14.01	13.88
9/3/2010	14.18	13.99
10/3/2010	14.31	14.09
11/3/2010	14.21	14.20
12/3/2010	14.26	14.30
3/15/2010	14.26	14.40
3/16/2010	14.3	14.51
3/17/2010	14.59	14.61
3/18/2010	14.55	14.71
3/19/2010	14.41	14.81
3/22/2010	14.62	14.91
3/23/2010	15.22	15.01
3/24/2010	14.99	15.11
3/25/2010	14.87	15.21
3/26/2010	14.99	15.31
3/29/2010	14.96	15.40
3/30/2010	14.97	15.50
3/31/2010	15.02	15.60

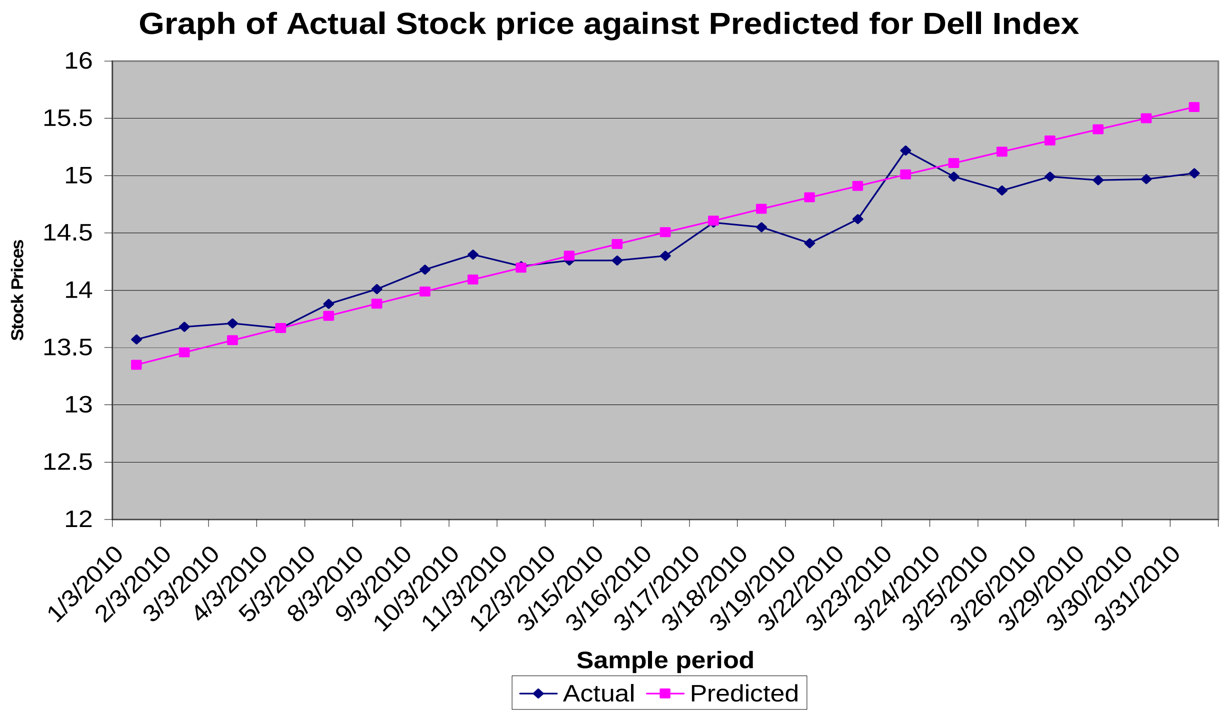


Figure 4.1: Graph of Actual Stock Price vs Predicted values of Dell Stock Index

Table 4.2: Confusion matrix of predicted result of ARIMA model for Dell index

		Predicted	
		Up	Down
Actual	Up	15	0
	Down	8	0

AC = 65%

4.2.2 **Result of ARIMA Model for Nokia Stock Price Prediction**

Table 4.3 is the result of the predicted values of ARIMA (2, 1, 0) considered the best model for Nokia stock index. Figure 4.2 gives graphical illustration of the level accuracy of the predicted price against actual stock price to see the performance of the ARIMA model selected. From the graph, is obvious that the performance is not satisfactory. The reason could be that valuable information had been lost due to differencing the time series for stationarity.

The accuracy of the model was found to be 57% as indicated the table 4.4 with confusion matrix.

Table 4.3: Sample of Empirical Results of ARIMA (2, 1, 0) of Nokia Stock Index.

Sample Period	Actual Values	Predicted Values
1/3/2010	13.28	13.58
2/3/2010	13.51	13.69
3/3/2010	13.86	13.80
4/3/2010	13.78	13.91
5/3/2010	14.13	14.02
8/3/2010	14.17	14.13
9/3/2010	14.12	14.24
10/3/2010	14.56	14.35
11/3/2010	14.49	14.45
12/3/2010	14.84	14.56
3/15/2010	14.81	14.67
3/16/2010	15.14	14.77
3/17/2010	15.42	14.88
3/18/2010	15.28	14.98
3/19/2010	15.07	15.09
3/22/2010	15.11	15.19
3/23/2010	15.26	15.30
3/24/2010	15.07	15.40
3/25/2010	15.20	15.50
3/26/2010	15.46	15.60
3/29/2010	15.42	15.71
3/30/2010	15.41	15.81
3/31/2010	15.54	15.91

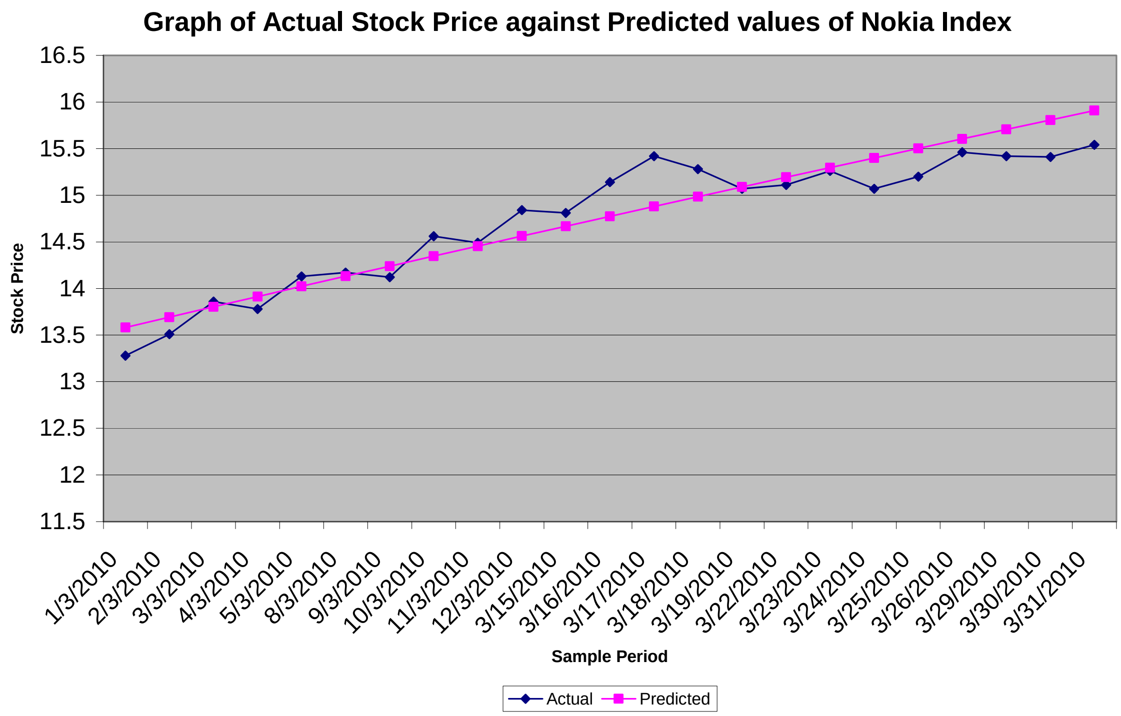


Figure 4.2: Graph of Actual Stock Price vs Predicted values of Nokia Stock Index

Table 4.4: Confusion matrix of predicted result of ARIMA model for Nokia index

		Predicted	
		Up	Down
Actual	Up	13	0
	Down	10	0

AC = 57%

4.2.3 **Result of ARIMA Model for Zenith Bank Stock Price Prediction**

In this case, ARIMA (1, 0, 1) was selected as the best model for Zenith bank stock index after several adjustment of the autoregressive (p) and moving average (q) parameters in Eviews software used. Table 4.5 contained the predicted values of the model selected and figure 4.3 is the graph of predicted price against actual stock price to demonstrate the correlation of accuracy. From the graph, the performance of the ARIMA model selected is satisfactory at the onset and dwindles as the days go by.

The model was evaluated with confusion matrix. The accuracy of the model was found to be 65% as indicated the table 4.6.

Table 4.5: Sample of Empirical Results of ARIMA (1, 0, 1) of Zenith Bank Index

Sample Period	Actual Values	Predicted Values
1/3/2010	16.19	15.83
2/3/2010	15.98	15.86
3/3/2010	15.71	15.89
4/3/2010	15.50	15.91
5/3/2010	15.70	15.94
8/3/2010	15.75	15.97
9/3/2010	15.86	15.99
10/3/2010	16.00	16.02
11/3/2010	16.19	16.05
12/3/2010	16.99	16.08
3/15/2010	17.83	16.10
3/16/2010	17.71	16.13
3/17/2010	17.50	16.16
3/18/2010	16.85	16.18
3/19/2010	17.69	16.21
3/22/2010	18.00	16.24
3/23/2010	18.00	16.26
3/24/2010	17.85	16.29
3/25/2010	17.89	16.31
3/26/2010	18.11	16.34
3/29/2010	19.01	16.37
3/30/2010	19.96	16.39
3/31/2010	18.97	16.42

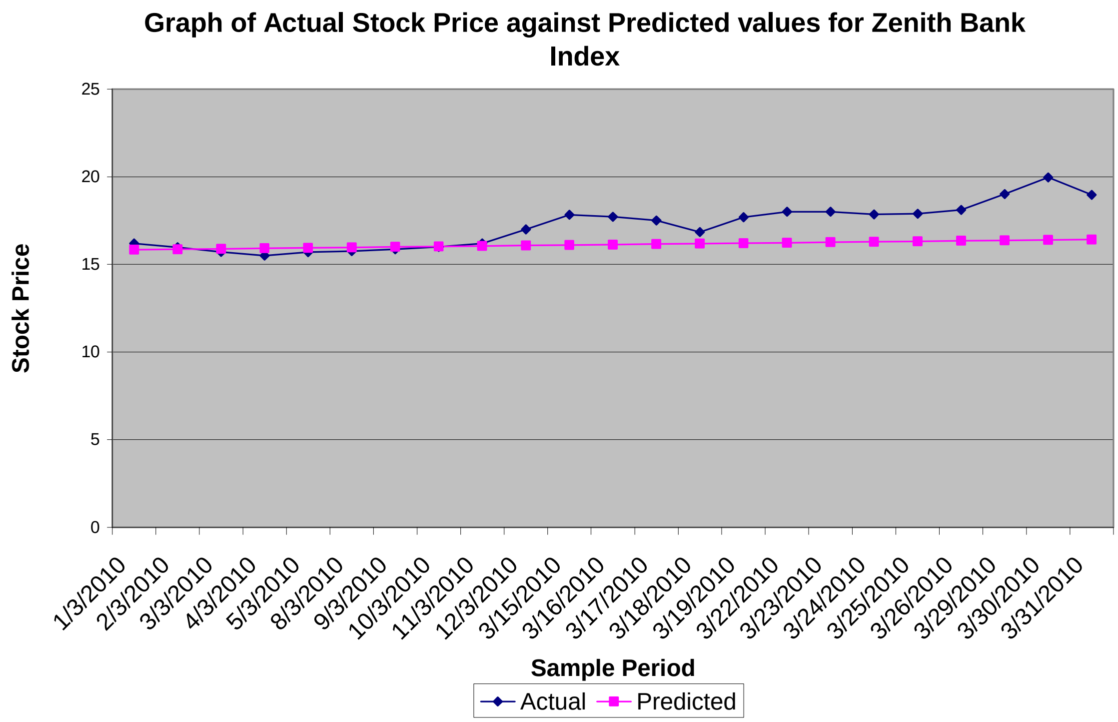


Figure 4.3: Graph of Actual Stock Price vs Predicted values of Zenith Bank Stock Index

Table 4.6: Confusion matrix of predicted result of ARIMA model for Zenith Bank index

		Predicted	
		Up	Down
Actual	Up	15	0
	Down	8	0

AC = 65%

4.2.4 **Result of ARIMA Model for UBA Bank Stock Price Prediction**

Similarly, ARIMA (1, 0, 0) was selected as the best model based on the criteria earlier mentioned. The results obtained are presented in table 4.7. Also, the graph to show level of accuracy of the model selected is in figure 4.4. From the graph, the performance of the ARIMA model is not satisfactory because there are wide margins between the actual values and the predicted values of the stock index. This result further showed that ARIMA model might not be suitable for financial forecasting.

The accuracy of the model was found to be 48% with confusion matrix as shown in the table 4.8.

Table 4.7: Sample of Empirical Results of ARIMA (1, 0, 0) of UBA Bank Index.

Sample Period	Actual Values	Predicted Values
1/3/2010	13.10	13.02
2/3/2010	13.09	13.04
3/3/2010	13.05	13.05
4/3/2010	13.00	13.07
5/3/2010	13.00	13.09
8/3/2010	13.15	13.11
9/3/2010	13.50	13.12
10/3/2010	13.51	13.14
11/3/2010	13.45	13.16
12/3/2010	14.11	13.17
3/15/2010	14.00	13.19
3/16/2010	13.82	13.21
3/17/2010	14.50	13.23
3/18/2010	14.50	13.24
3/19/2010	14.45	13.26
3/22/2010	14.61	13.28
3/23/2010	14.91	13.30
3/24/2010	14.84	13.31
3/25/2010	14.84	13.33
3/26/2010	14.89	13.35
3/29/2010	14.91	13.36
3/30/2010	15.20	13.38
3/31/2010	15.20	13.40

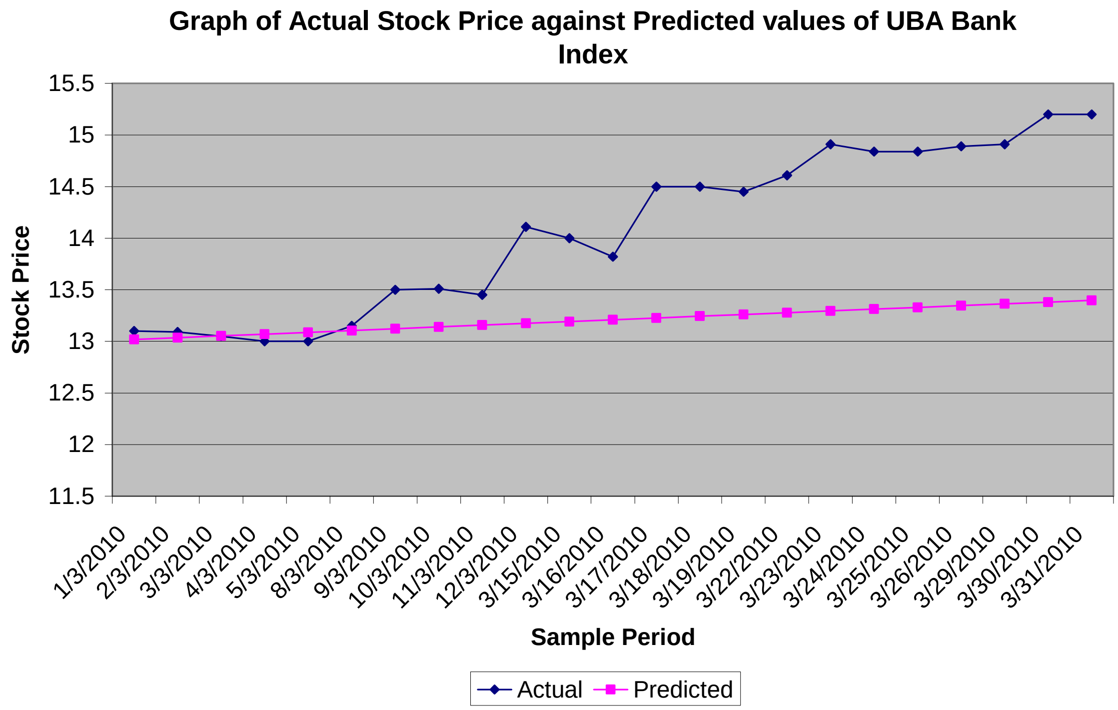


Figure 4.4: Graph of Actual Stock Price vs Predicted values of UBA Bank Stock Index

Table 4.8: Confusion matrix of predicted result of ARIMA model for UBA Bank index

		Predicted	
		Up	Down
Actual	Up	11	0
	Down	12	0

AC = 48%

4.3 ANN MODEL RESULTS

After several experiments with different network architectures, the network predictive model that gave the most accurate daily stock price prediction in model 1, model 2 and model 3 are presented in a graphical form and also in a table that compares the prediction accuracies of the different models developed.

The input neurons of model 1 consist of ten technical analysis variables. The inputs neurons of Model 2 consist of thirteen neurons of combined technical and fundamental analysis variables. The proposed model (model 3), consists of eighteen neurons of combined technical, fundamental and experts' opinion variables respectively. The results of each of the stock index used are presented in following subsections.

4.3.1 Results of ANN Model for Dell Stock Price Prediction

The network structure that returns the smallest mean squared error (MSE) was noted to give the best forecasting accuracy with the test data. Table 4.9 contained the results of the mean squared errors recorded in the course of the experiment. In this case, the network predictive model that gave the most accurate daily price prediction in model 1 was 10-17-1 (ten input neurons, seventeen hidden neurons and one output neuron). Similarly, for model 2, the network predictive model that returns smallest MSE was 13-19-1 (thirteen input neurons, nineteen hidden neurons, one output). The best-fitted network that gave the best forecasting accuracy with test data composed of eighteen inputs, twenty-six hidden neurons and one output neuron 18-26-1. The results presented in table 4.10 were the findings from testing period (out of sample test data) over the three different models. Also, figure 4.5 - 4.7 illustrates the correlation of the level accuracy among different models.

From the empirical results, the forecasting accuracy level of model 1 compared with model 2 are quite impressive. However, the performance of model 2 was better than model 1 in the level of accuracy on many occasions from the different test data. From the figure 4.7, it is obvious that model 3 is the best of all the three predictive models. There is a great improvement in terms of forecasting accuracy in comparison to results of model 1 and 2. The stock price prediction accuracy of the proposed model that combined technical, fundamental indicators and experts' opinion to create a predictive model was the best with average of 74% accuracy as shown in table 4.11. Hence, the proposed predictive model can be used successfully as decision-support in real-life trading in a way that will enhance the profit margin of investors or traders in daily trading.

Table 4.9: Statistical performance of Mode 1-3 of Dell Stock index

	MSE			
	Network Structure	**1000 Epochs**	**2000 Epochs**	**5000 Epochs**
Model 1	10-10-1	0.129054	0.112363	0.093539
	10-11-1	0.144086	0.108245	0.090521
	10-12-1	0.125668	0.099301	0.088157
	10-13-1	0.148646	0.115732	0.092649
	10-14-1	0.141474	0.099241	0.085206
	10-15-1	0.118226	0.096651	0.083664
	10-16-1	0.116773	0.099222	0.080534
	10-17-1	**0.097826**	**0.085111**	**0.071589**
	10-18-1	0.119719	0.093576	0.079150
Model 2	13-13-1	0.123184	0.107412	0.090236
	13-14-1	0.130018	0.095150	0.075755
	13-15-1	0.179753	0.121711	0.090089
	13-16-1	0.118892	0.093771	0.076748
	13-17-1	0.128121	0.103660	0.087496
	13-18-1	0.123907	0.097827	0.079230
	13-19-1	0.115329	**0.093167**	**0.067901**
	13-20-1	0.118568	0.101955	0.085272
	13-21-1	**0.113894**	0.094207	0.077037
Model 3	18-18-1	0.125340	0.101465	0.083380
	18-19-1	0.153357	0.143036	0.144930
	18-20-1	0.125132	0.121522	0.131991
	18-21-1	0.110519	0.090850	0.072907
	18-22-1	0.115685	0.093702	0.069479
	18-23-1	**0.106685**	0.090579	0.070017
	18-24-1	0.129842	0.107310	0.078529
	18-25-1	0.112151	0.090779	0.070529
	18-26-1	0.112049	**0.088974**	**0.061821**

Bold characters indicate the best results for each of epoch session

Table 4.10: Sample of Empirical Results of ANN Models of Dell Stock Index

Sample Period	Actual Value	Predicted Values		
		Model 1	Model 2	Model 3
3/31/2010	15.02	14.97	14.93	15.03
3/30/2010	14.97	15.05	14.96	15.11
3/29/2010	14.96	14.59	14.69	14.63
3/26/2010	14.99	14.50	14.66	14.62
3/25/2010	14.87	14.96	15.19	15.52
3/24/2010	14.99	14.66	15.23	14.96
3/23/2010	15.22	14.23	14.30	14.57
3/22/2010	14.62	13.67	13.98	14.02
3/19/2010	14.41	13.85	14.37	14.05
3/18/2010	14.55	14.25	14.36	14.51
3/17/2010	14.59	14.00	13.88	14.19
3/16/2010	14.30	13.89	13.76	14.11
3/15/2010	14.26	13.96	13.91	14.13
3/12/2010	14.26	13.59	13.60	13.76
3/11/2010	14.21	14.18	14.33	14.40
3/10/2010	14.31	13.85	13.92	14.24
3/09/2010	14.18	13.92	13.95	14.26
3/08/2010	14.01	13.57	13.66	13.90
3/05/2010	13.88	13.33	13.39	13.53
3/04/2010	13.67	13.09	13.25	13.17
3/03/2010	13.71	13.70	13.76	13.89
3/02/2010	13.68	13.55	13.69	13.94
3/01/2010	13.57	13.16	13.19	13.31

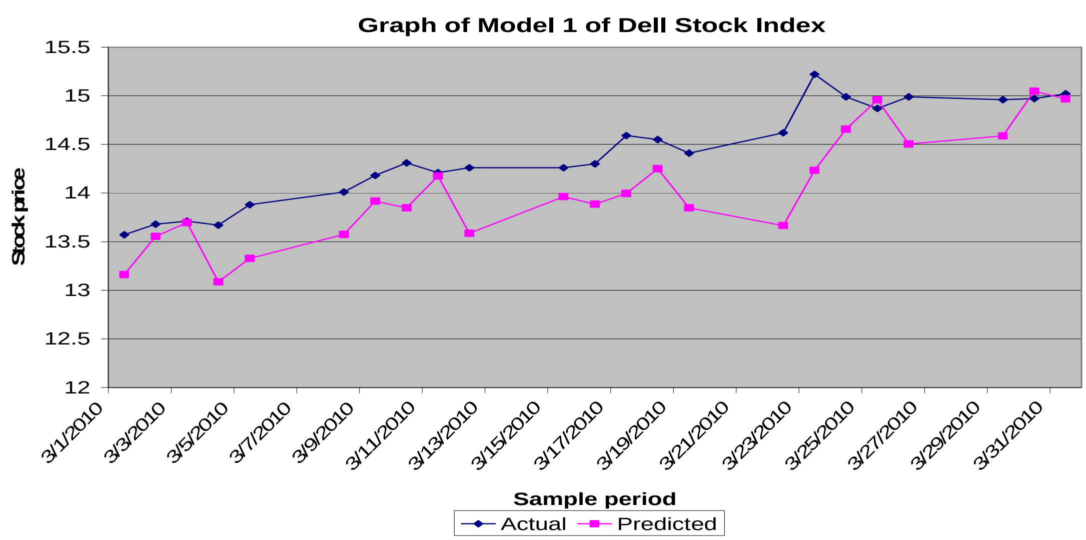


Figure 4.5: Graph of Actual Stock Price vs Predicted values of Model 1 of Dell Index

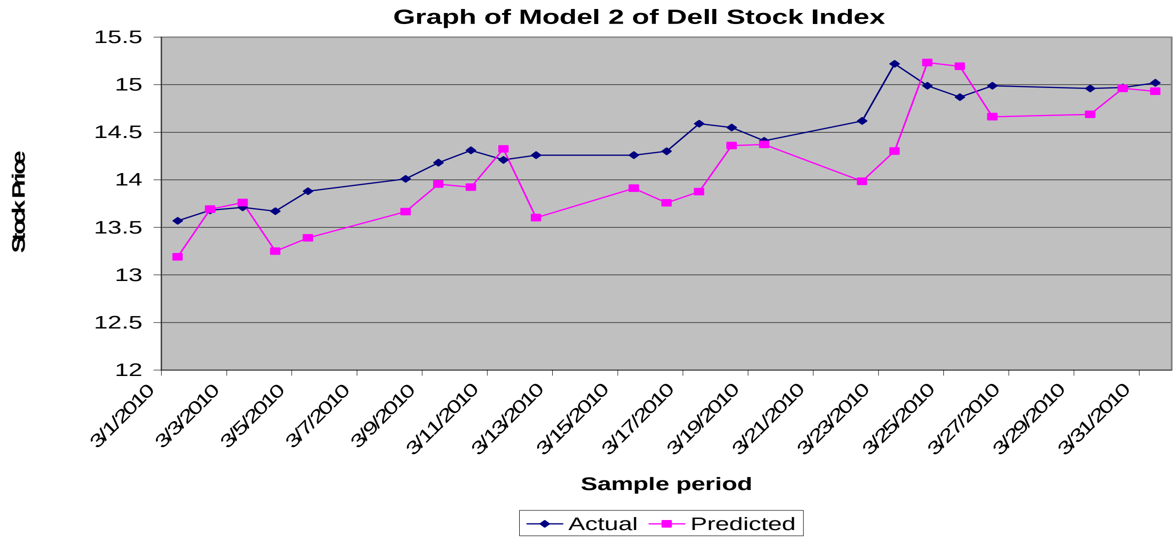


Figure 4.6: Graph of Actual Stock Price vs Predicted values of Model 2 of Dell Index

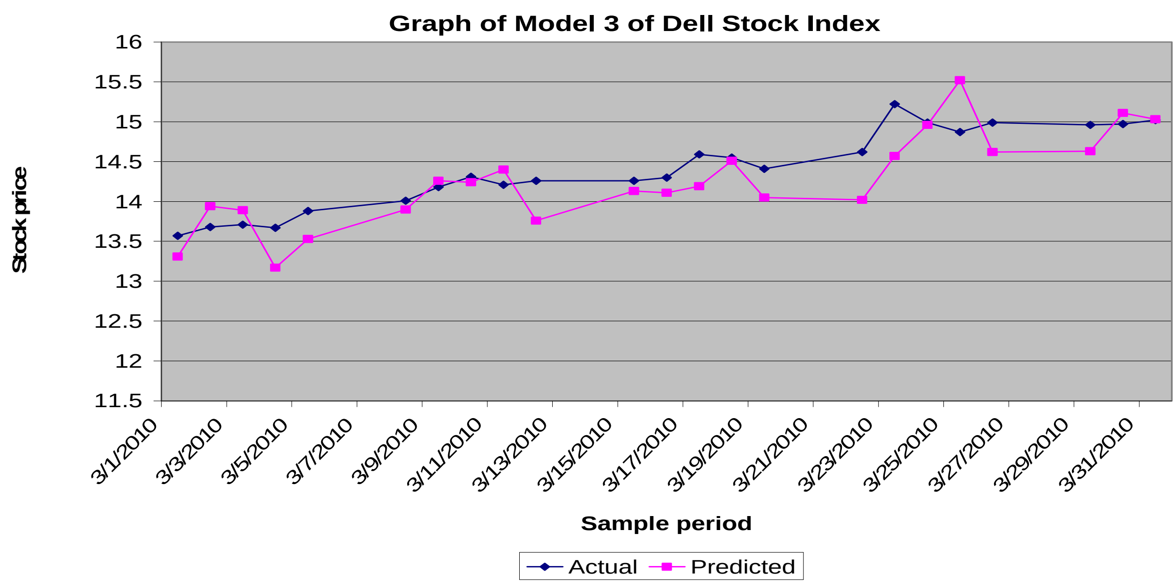


Figure 4.7: Graph of Actual Stock Price vs Predicted values of Model 3 of Dell Index

Table 4.11: Confusion matrix of predicted result of ANN model for Dell index

		Predicted	
		Up	Down
Actual	Up	13	3
	Down	5	4

AC = 74%

4.3.2 **Results of ANN Model for Nokia Stock Price Prediction**

In table 4.12, the highlighted values indicate the ANN structure that returns smallest mean square errors which are noted to give best performance over other structures. For model 1, we have 10-14-1 (ten input neurons, fourteen hidden neurons and one output neuron). For model 2, we have 13-14-1 (thirteen input neurons, fourteen hidden neurons, one output) and for model 3, we have 18-21-1 (eighteen inputs, twenty-one hidden neurons and one output neuron). However, model 3 is noted to give the best prediction over the other two models. The results presented in table 4.13 are the predicted values of each of the model earlier mentioned.

Also, figures 4.8 - 4.10 illustrate the correlation of the level accuracy among different models. From the figure 4.10, it is obvious that model 3 is the best of all the three stock price predictive models with average of 70% accuracy as shown in table 4.14.

Table 4.12: Statistical performance of Mode 1-3 of Nokia Stock index

	MSE			
	Network Structure	**1000 Epochs**	**2000 Epochs**	**5000 Epochs**
Model 1	10-10-1	0.131413	0.085673	0.065958
	10-11-1	0.130266	0.100147	0.074376
	10-12-1	**0.104179**	**0.076013**	0.055319
	10-13-1	0.148378	0.123140	0.094993
	10-14-1	0.113599	0.077468	**0.052256**
	10-15-1	0.120586	0.094881	0.063535
	10-16-1	0.112569	0.087723	0.590738
	10-17-1	0.121481	0.099204	0.064114
	10-18-1	0.129727	0.088369	0.074229
Model 2	13-13-1	0.142468	0.111148	0.083957
	13-14-1	0.125704	0.085771	**0.048449**
	13-15-1	0.130336	0.092512	0.063329
	13-16-1	0.120522	**0.070893**	0.051680
	13-17-1	2.328160	0.092799	0.053544
	13-18-1	0.132760	0.109483	0.068056
	13-19-1	**0.110694**	0.081275	0.053134
	13-20-1	4.077420	0.110826	0.067431
	13-21-1	0.145350	0.100590	0.052527
Model 3	18-18-1	0.109847	**0.048198**	0.041309
	18-19-1	0.131226	0.102727	0.066412
	18-20-1	**0.109616**	0.073874	0.052122
	18-21-1	0.136433	0.070053	**0.038898**
	18-22-1	0.130056	0.096053	0.054354
	18-23-1	0.118424	0.083801	0.049239
	18-24-1	0.170919	0.083517	0.050821
	18-25-1	4.497720	4.374340	4.347050
	18-26-1	0.222150	0.202463	0.062659

Bold characters indicate the best results for each of epoch session

Table 4.13: Sample of Empirical Results of ANN Models of Nokia Stock Index

Sample Period	Actual Value	Predicted Values		
		Model 1	Model 2	Model 3
3/31/2010	15.54	15.42	15.46	15.34
3/30/2010	15.41	15.43	15.44	15.44
3/29/2010	15.42	15.32	15.39	15.31
3/26/2010	15.46	15.15	15.31	15.07
3/25/2010	15.2	15.19	15.26	15.16
3/24/2010	15.07	15.21	15.33	15.16
3/23/2010	15.26	15.13	15.23	15.20
3/22/2010	15.11	15.17	15.24	15.13
3/19/2010	15.07	15.43	15.40	15.47
3/18/2010	15.28	15.21	15.16	15.07
3/17/2010	15.42	14.99	15.14	14.79
3/16/2010	15.14	15.05	15.27	15.04
3/15/2010	14.81	14.91	15.21	14.68
3/12/2010	14.84	14.52	14.89	14.75
3/11/2010	14.49	13.81	14.33	14.45
3/10/2010	14.56	13.98	14.40	14.62
3/09/2010	14.12	14.31	14.89	14.40
3/08/2010	14.17	13.94	14.58	13.92
3/05/2010	14.13	13.79	14.67	14.19
3/04/2010	13.78	13.47	14.16	13.71
3/03/2010	13.86	13.47	14.13	13.82
3/02/2010	13.51	13.39	13.79	13.50
3/01/2010	13.28	13.02	13.59	13.10

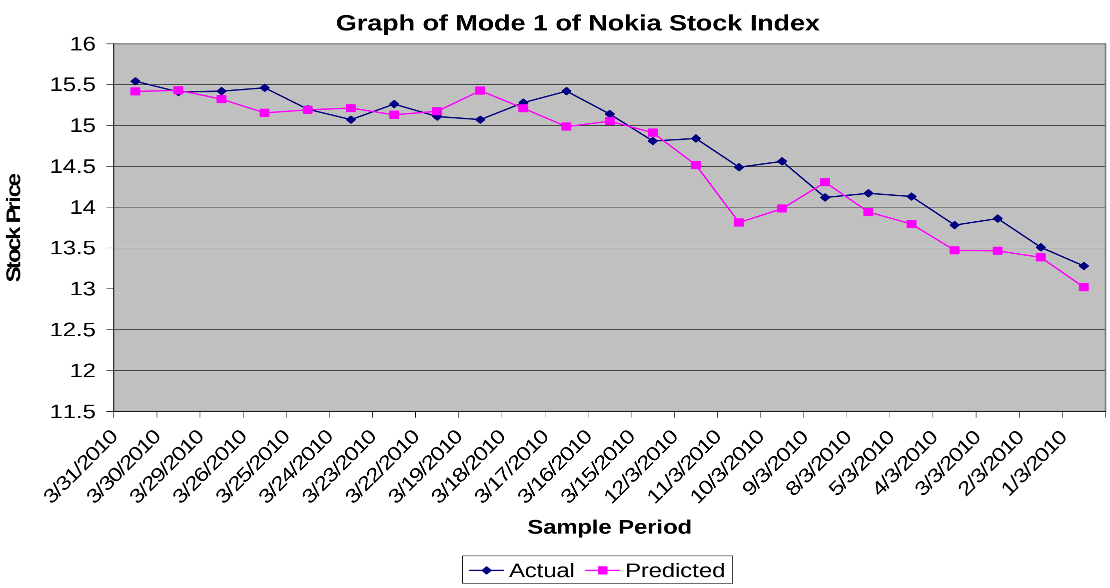


Figure 4.8: Graph of Actual Stock Price vs Predicted values of Model 1 of Nokia Index

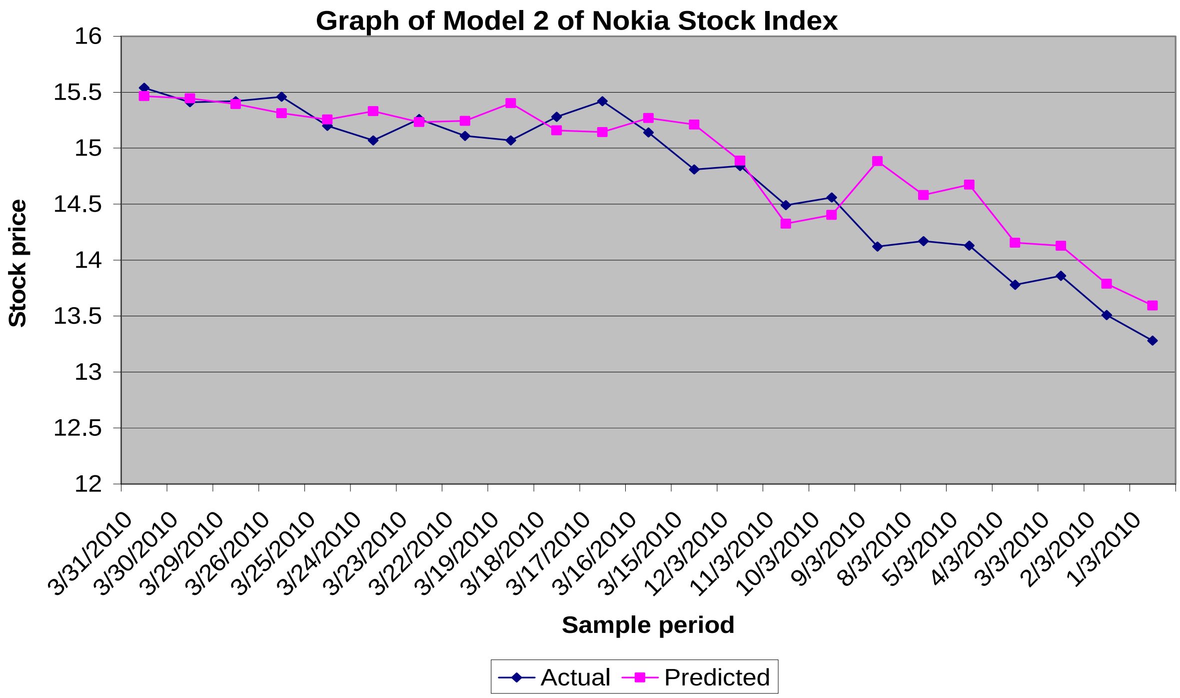


Figure 4.9: Graph of Actual Stock Price vs Predicted values of Model 2 of Nokia Index

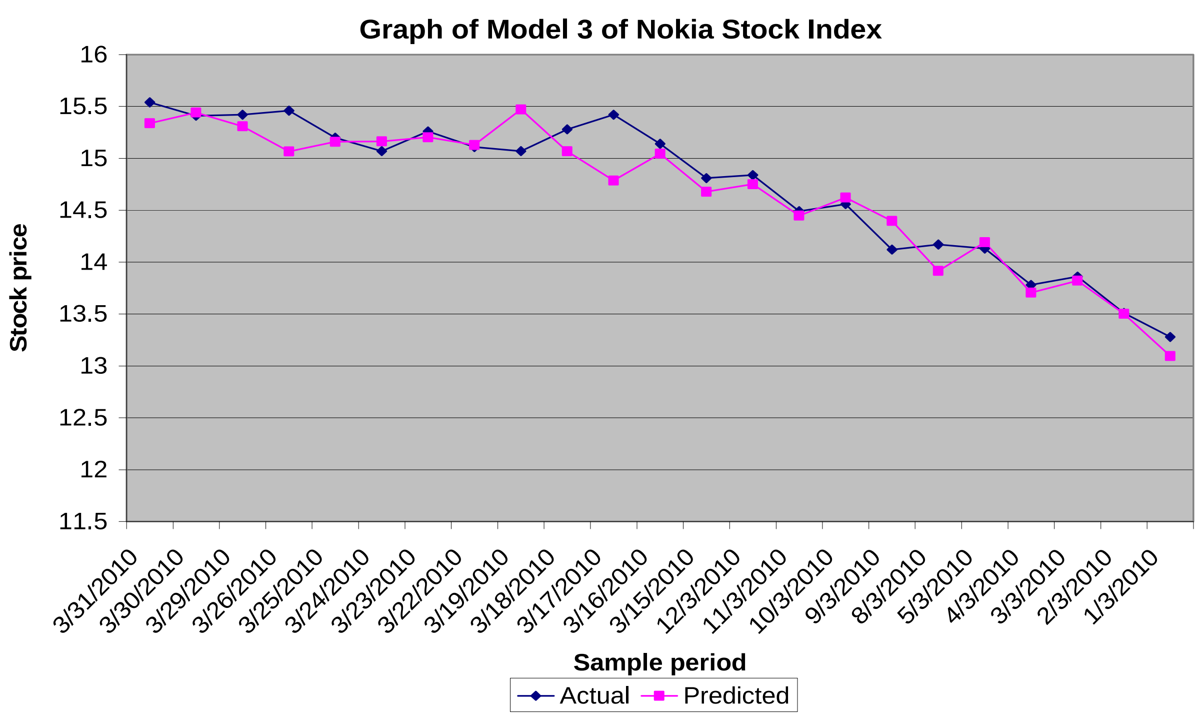


Figure 4.10: Graph of Actual Stock Price vs Predicted values of Model 3 of Nokia Index

Table 4.14: Confusion matrix of predicted result of ANN model for Nokia index

		Predicted	
		Up	Down
Actual	Up	9	4
	Down	3	7

AC = 70%

4.3.3 **Results of ANN Model for Zenith Bank Stock Price Prediction**

In this case, best network structure in model 1 is 10-12-1 (ten input neurons, twelve hidden neurons and one output neuron). Also, for mode 2 is 13-19-1 (thirteen input neurons, nineteen hidden neurons, one output) and model 3 is 18-20-1 (eighteen inputs, twenty hidden neurons and one output neuron). Table 4.15 is the outcomes of the mean square errors recorded in the various experiments with different networks configurations.

Table 4.16 is the results of the predicted values of each of the model mentioned.

Figures 4.11 - 4.13 shows the level forecasting accuracy among different models. From the figures 4.13 below, model 3 gave the best result above model 1 and model 2 with average of 74% accuracy as shown in table 4.17.

Table 4.15: Statistical performance of Mode 1-3 of Zenith Bank Stock index

	MSE			
	Network Structure	**1000 Epochs**	**2000 Epochs**	**5000 Epochs**
Model 1	10-10-1	4.108300	4.099800	4.098230
	10-11-1	0.035962	0.019059	0.010922
	10-12-1	**0.018442**	**0.009902**	**0.005750**
	10-13-1	4.102440	4.100620	4.098390
	10-14-1	0.027818	0.018080	0.009482
	10-15-1	4.028170	0.020783	0.011648
	10-16-1	4.110630	4.108430	0.013559
	10-17-1	4.126330	4.126040	4.063250
	10-18-1	4.111820	4.111760	4.103740
Model 2	13-13-1	4.121220	4.114060	4.098280
	13-14-1	0.035358	0.018686	0.008857
	13-15-1	4.111620	4.111430	4.106180
	13-16-1	0.031099	0.019543	0.010096
	13-17-1	4.109700	4.098990	4.098160
	13-18-1	4.100310	4.099320	4.098830
	13-19-1	**0.019570**	**0.008205**	**0.005829**
	13-20-1	4.099930	4.099340	4.098750
	13-21-1	4.116100	4.098810	4.098320
Model 3	18-18-1	4.120570	0.036615	0.005744
	18-19-1	0.029132	0.011933	0.005900
	18-20-1	0.015882	**0.007599**	**0.004163**
	18-21-1	0.029142	0.012597	0.006514
	18-22-1	**0.015248**	0.009414	0.007592
	18-23-1	4.047870	4.047670	0.019429
	18-24-1	4.110580	4.099050	4.097920
	18-25-1	0.021306	0.010797	0.007444
	18-26-1	4.102400	4.100080	4.098580

Bold characters indicate the best results for each of epoch session

Table 4.16: Sample of Empirical Results of ANN Models of Zenith Bank Index

Sample Period	Actual Value	Predicted Values		
		Model 1	Model 2	Model 3
3/1/2010	15.98	15.64	16.12	16.01
3/2/2010	15.71	17.10	15.22	16.37
3/3/2010	15.50	15.43	15.78	15.74
3/4/2010	15.70	15.71	15.93	15.88
3/5/2010	15.75	15.82	15.70	15.73
3/8/2010	15.86	16.39	16.08	16.20
3/9/2010	16.00	15.96	16.75	16.78
3/10/2010	16.19	17.25	17.13	17.36
3/11/2010	16.99	18.59	18.57	18.12
3/12/2010	17.83	18.39	18.11	18.09
3/15/2010	17.71	17.35	18.01	17.41
3/16/2010	17.50	15.26	15.20	15.19
3/17/2010	15.25	15.36	15.37	15.31
3/18/2010	15.25	19.01	18.92	18.69
3/19/2010	18.00	18.55	18.39	18.32
3/22/2010	18.00	17.94	18.62	18.38
3/23/2010	17.85	18.14	18.41	18.17
3/24/2010	17.89	18.66	18.55	18.60
3/25/2010	18.11	19.14	19.03	19.08
3/26/2010	19.01	19.35	19.34	19.15
3/29/2010	19.96	18.80	18.90	18.97
3/30/2010	18.97	18.98	19.07	19.11
3/31/2010	19.30	19.11	19.11	18.90

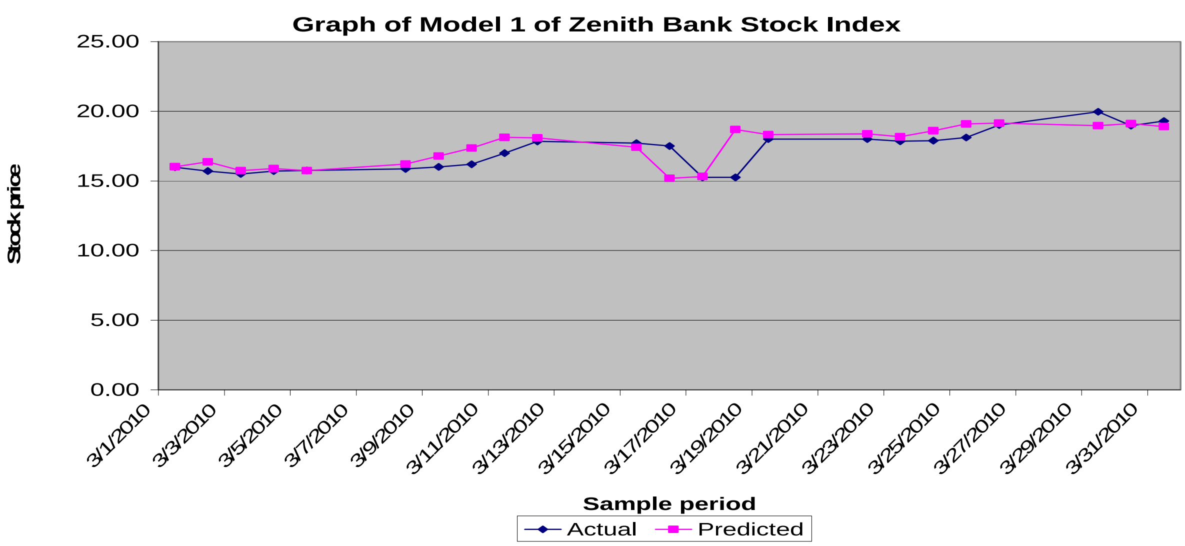


Figure 4.11: Graph of Actual Stock Price vs Predicted values of Model 1 of Zenith Index

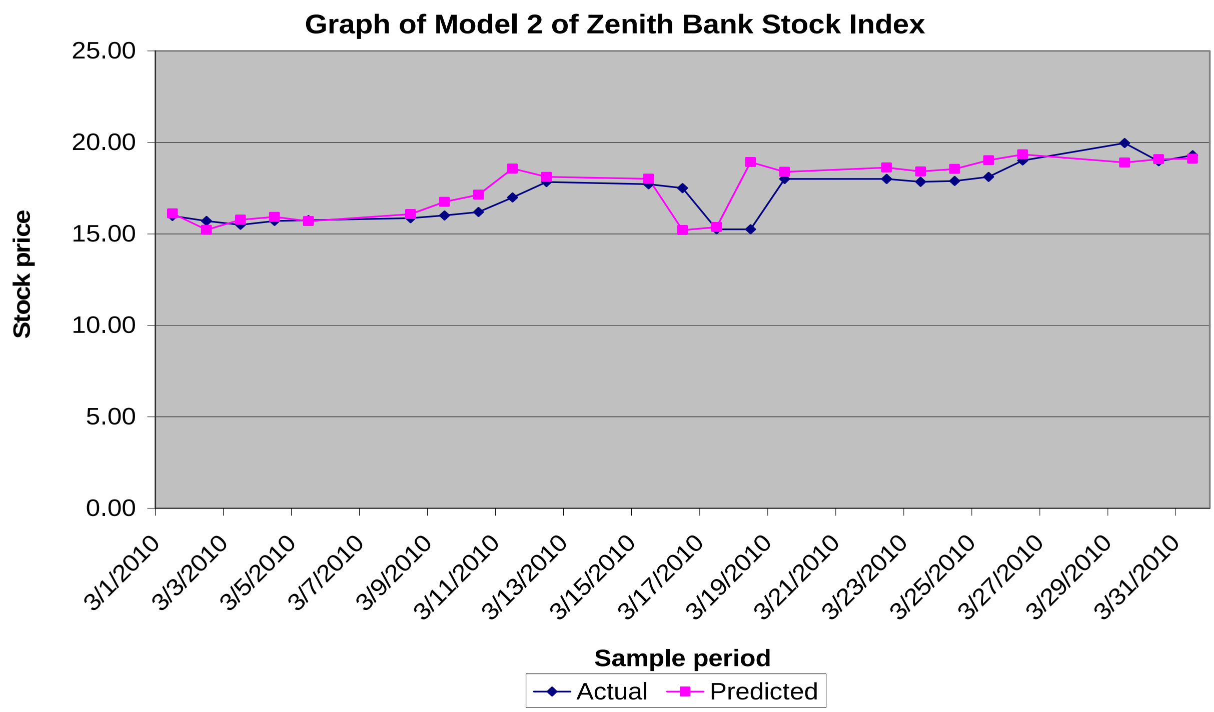


Figure 4.12: Graph of Actual Stock Price vs Predicted values of Model 2 of Zenith Index

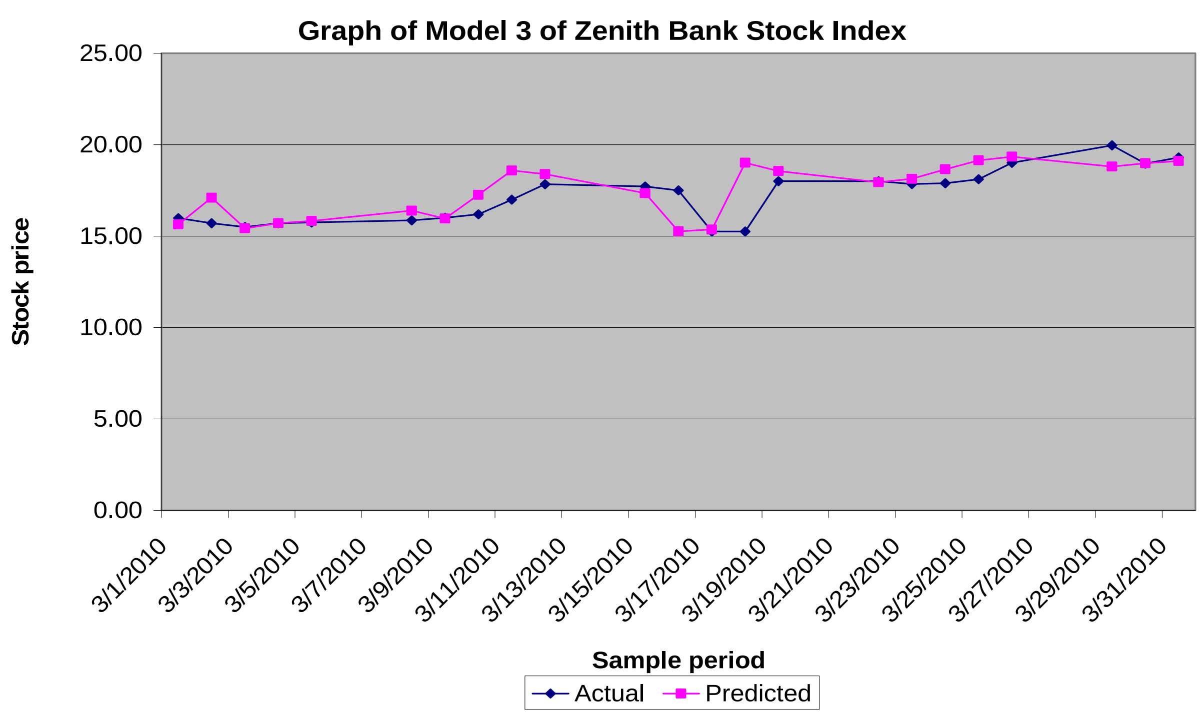


Figure 4.13: Graph of Actual Stock Price vs Predicted values of Model 3 of Zenith Index

Table 4.17: Confusion matrix of predicted result of ANN model for Zenith index

		Predicted	
		Up	Down
Actual	Up	12	3
	Down	3	5

AC = 74%

4.3.4 Results of ANN Model for UBA Bank Stock Price Prediction

Similarly, for UBA stock index, the network topology that returns the smallest mean square error was noted to give the best forecasting accuracy with the test data is contained in table 4.18. In this case, the network predictive model that gave the most accurate daily price prediction in model 1 was 10-17-1 (ten input neurons, seventeen hidden neurons and one output neuron). For model 2, we have 13-17-1 (thirteen input neurons, seventeen hidden neurons, one output) and model 3, we have 18-24-1 (eighteen inputs, twenty-four hidden neurons and one output neuron). The results presented in table 4.19 were the findings from testing period (out of sample test data) over the three different network models. Also, figures 4.14 - 4.16 illustrate the correlation of the level accuracy among different models.

From the figures 4.14 – 4.16, we observed that there is no significant difference between model 1 and model 2. However, there is remarkable improvement in model 3 over the two other models. Model 3 is the proposed stock price predictive model that combined technical, fundamental indicators and experts' opinion as input to create a predictive model. The average accuracy of model 3 is 78% accuracy as shown in table 4.20.

Table 4.18: Statistical performance of Mode 1-3 of UBA Bank Stock index

	MSE			
	Network Structure	**1000 Epochs**	**2000 Epochs**	**5000 Epochs**
Model 1	10-10-1	0.065771	0.039392	0.015526
	10-11-1	0.040859	0.018023	0.007149
	10-12-1	**0.022149**	**0.009799**	0.006476
	10-13-1	0.045920	0.029234	0.014437
	10-14-1	0.029664	0.018399	0.011376
	10-15-1	0.410990	0.030402	0.008053
	10-16-1	0.026901	0.011973	0.006532
	10-17-1	0.026850	0.012953	**0.006317**
	10-18-1	0.035467	0.014900	0.006513
Model 2	13-13-1	0.045267	0.024678	0.012737
	13-14-1	0.049285	0.020317	0.009248
	13-15-1	0.049302	0.028288	0.014649
	13-16-1	0.033183	0.012358	0.005490
	13-17-1	**0.027930**	**0.011691**	**0.005297**
	13-18-1	0.028951	0.012218	0.006627
	13-19-1	0.033560	0.018832	0.009782
	13-20-1	0.035910	0.018103	0.008589
	13-21-1	0.037007	0.024169	0.014154
Model 3	18-18-1	0.033091	0.012269	0.005957
	18-19-1	0.033951	0.019018	0.009853
	18-20-1	0.022090	0.012050	0.007072
	18-21-1	0.029914	0.014159	0.006764
	18-22-1	**0.020594**	0.008664	0.003818
	18-23-1	0.030647	0.013578	0.006147
	18-24-1	0.034620	**0.008276**	**0.003816**
	18-25-1	0.028813	0.011532	0.004925
	18-26-1	0.058627	0.017453	0.007322

Bold characters indicate the best results for each of epoch session

Table 4.19: Sample of Empirical Results of ANN Models of UBA Bank Index

Sample Period	Actual Value	Predicted Values		
		Model 1	Model 2	Model 3
3/1/2010	13.39	13.30	13.38	13.49
3/2/2010	13.05	13.05	13.08	12.99
3/3/2010	13.00	12.99	13.06	12.90
3/4/2010	13.00	13.01	13.06	13.04
3/5/2010	13.15	13.09	13.13	13.06
3/8/2010	13.50	13.33	13.31	13.27
3/9/2010	13.51	13.45	13.39	13.13
3/10/2010	13.45	13.13	13.23	12.58
3/11/2010	14.11	13.86	13.85	13.90
3/12/2010	14.00	13.85	14.03	13.59
3/15/2010	13.82	13.98	13.91	13.49
3/16/2010	13.61	14.70	15.80	13.63
3/17/2010	14.50	14.45	14.09	14.45
3/18/2010	14.50	14.39	14.25	14.40
3/19/2010	14.61	14.31	14.36	14.55
3/22/2010	14.91	14.63	14.80	14.65
3/23/2010	14.84	14.55	14.69	14.86
3/24/2010	14.84	14.66	14.52	14.81
3/25/2010	14.89	14.78	14.87	14.74
3/26/2010	14.91	14.86	14.90	14.76
3/29/2010	15.20	15.22	15.10	15.05
3/30/2010	15.20	14.98	15.48	15.26
3/31/2010	14.80	14.81	14.95	14.86

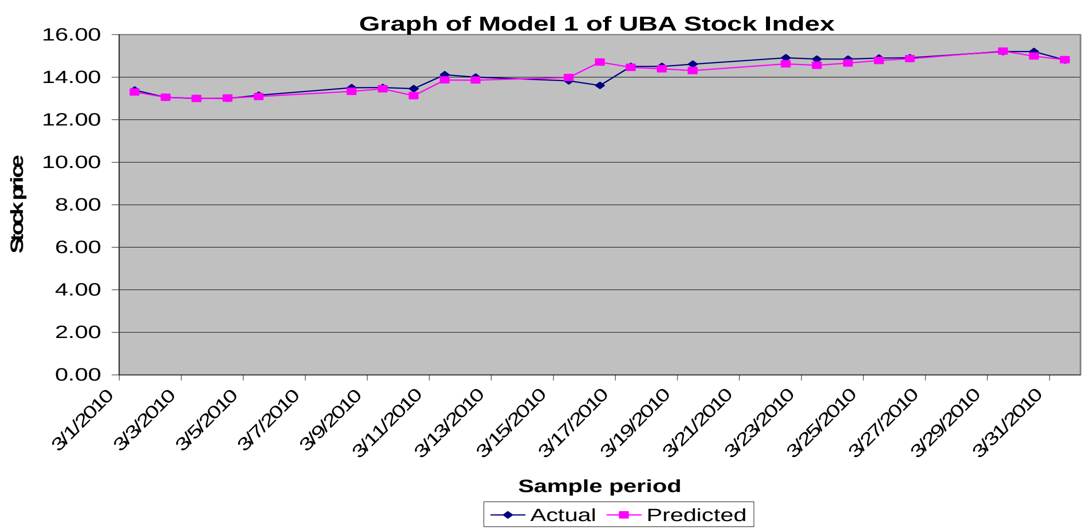


Figure 4.14: Graph of Actual Stock Price vs Predicted values of Model 1 of UBA Index

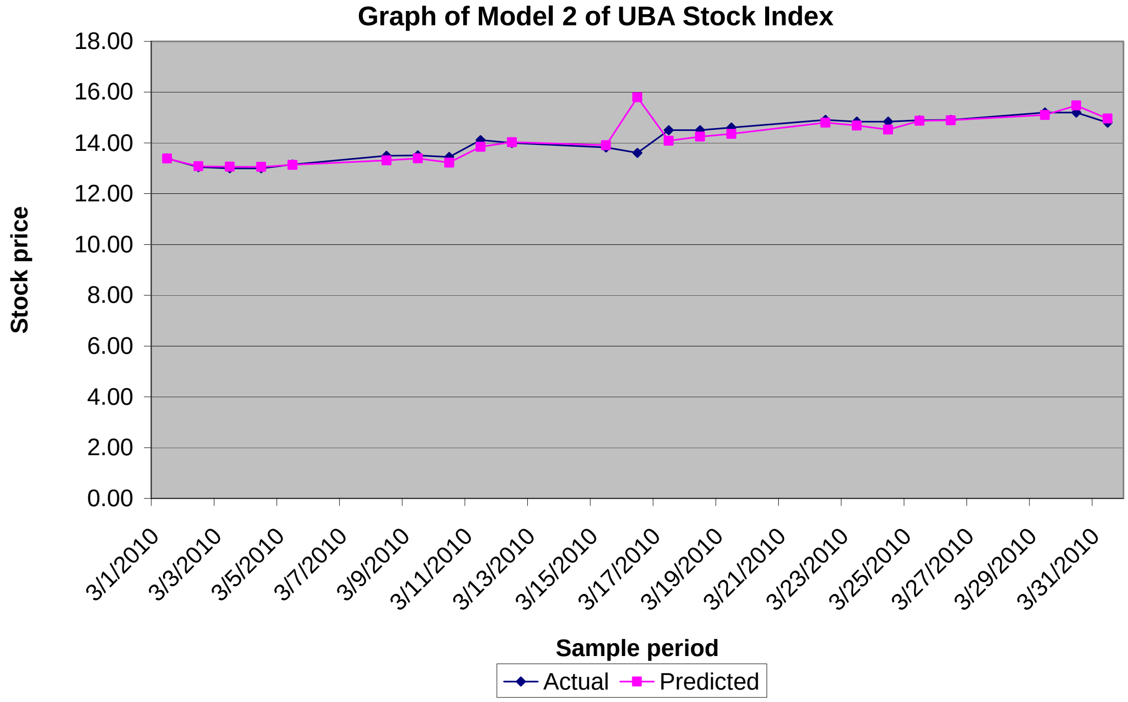


Figure 4.15: Graph of Actual Stock Price vs Predicted values of Model 2 of UBA Index

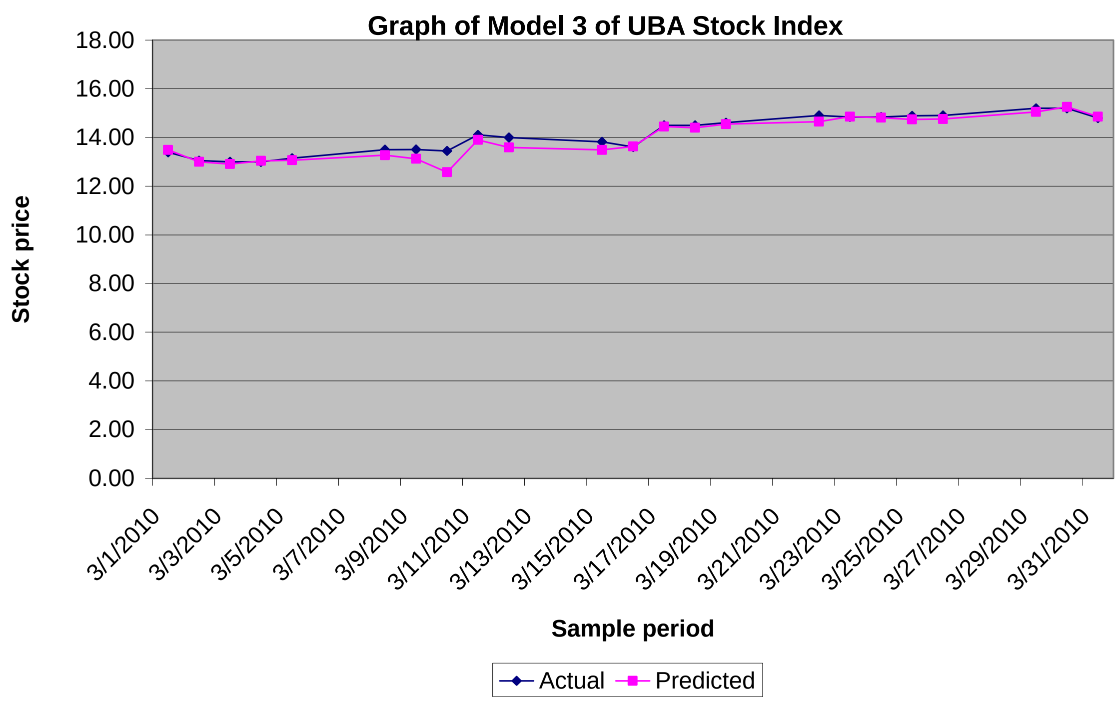


Figure 4.16: Graph of Actual Stock Price vs Predicted values of Model 3 of UBA Index

Table 4.20: Confusion matrix of predicted result of ANN model for UBA index

		Predicted	
		Up	Down
Actual	Up	11	1
	Down	4	7

AC = 78 %

4.4 SUMMARY OF RESULTS OF THE FORECASTING TECHNIQUES

Going by the results of the two forecasting techniques (ARIMA and ANN) engaged in this research work. The results presented above showed the forecasting performance of each of them. The observation noted in the experiments carried out with the two forecasting techniques is that the forecasting ability of ARIMA model is not satisfactory for financial forecasting particularly stock data when compared with the results obtained with ANN model with the same set of data used. Based on the results obtained in this work, ANN is superior to ARIMA model for time series forecasting. These results also resolved the discrepancies in the literature about the superiority of each of the model over one another.

In this study, three ANN models were developed with different input parameters in each of the model. It was noted that the proposed ANN model that combined technical, fundamental and expert's opinion variables gave best accurate forecasting results than the other two models with four companies stock data sourced NSE and NYSE that was used in this study.

CHAPTER FIVE

SUMMARY AND CONCLUSION

5.1 INTRODUCTION

This chapter gives the summary of the work done and discusses the contributions of the study, and presents an outlook of the opportunities for future research work.

5.2 SUMMARY

The focus of this study is to evolve an improved predictive model for stock price prediction. Two forecasting techniques were used in this study. The statistical technique of autoregressive integrated moving average (ARIMA) model and the soft computing technique (artificial neural networks). Stock data that were used to evaluate the performance of the models developed with the two forecasting techniques were sourced from New York stock exchange and Nigerian stock exchange respectively. Three different predictive models were created with artificial neural network namely model 1, model 2 and model 3 respectively. Model 1 consisted of technical analysis variables as input to the neural network models with different network configurations to ascertain the model with highest performance. Similarly, model 2 composed of technical and fundamental analysis variables while model 3 combined technical, fundamental and expert opinion variables to create the proposed predictive model. The performance measure used in this study to evaluate the model developed are adjusted R-squared, Bayesian or Schwarz Information criterion (BIC) and Standard error of regression for ARIMA model and mean square error was used for ANN model. The accuracy of forecasting of the model was evaluated with the confusion matrix. The empirical results showed that model 3 showed an improvement over model 1 and model 2 in accurate forecasting of the stock prices from the sampled test data used. Also our findings revealed that the soft computing forecasting technique (ANN) is superior to statistical technique in forecasting stock prices.

The main contributions of this thesis stem from the following: Firstly, the study provide an improved predictive model for stock price prediction using the ANN approach with hybrid market indicators that combined the parameters of technical and fundamental analyses as well as the experts' opinion. The review of previous studies on stock price forecasting shows that the use of technical indicators with ANN model is prevalent while there are only few cases of the use of fundamental indicators. However, this study contrasts previous approaches by combining technical indicators, fundamental indicators and experts' opinion to improve stock price prediction using the soft computing approach.

Secondly, this study was able to resolve and clarify contradictory findings reported in literature on the superiority of statistical techniques over soft computing techniques in time series prediction and vice-versa. The findings in this study showed ANN model outperformed the statistical forecasting techniques particularly the widest used statistical forecasting technique – autoregressive integrated moving average (ARIMA) model.

5.3 CONCLUSION

Technical indicators had been widely used in forecasting stock prices with artificial neural networks. Nevertheless, their performance is not always satisfactory. Also, in recent times, hybrid models that combine ANNs and other intelligent techniques with technical indices had been engaged in order to improve accuracy level of stock price prediction with varying results.

In this thesis, an improved predictive model for stock price prediction based on experts' opinion with technical and fundamental analysis variables using the soft computing approach of artificial neural model is presented. What distinguishes this research work from earlier work is that it combined different market indicators to evolve stock price predictive model. In particular, it introduced a new variable called expert's opinion as input to the neural network predictive model.

In this research, we have developed novel approach to forecast stock prices. We evaluated the performance of our proposed model (model 3) using daily stock prices of four companies listed in NYSE and NSE respectively. The empirical results confirmed superior performance of the proposed model to improve forecasting accuracy of stock price over the conventional approach of using ANN model with technical indicators. Therefore, the proposed predictive model has the potential to enhance the quality of decision making of investors in the stock market by offering more accurate stock prediction. It is worth noting that forecasting time series has been researched for a long time with varying success and it would be notably significant even if a little improvement in performance is achieved.

5.4 FUTURE RESEARCH WORK

The study provides several opportunities for further research in the immediate future. Full implementation of the model in real time and on mobile platform would no doubt contribute to the development of electronic finance. Also, the critical impact of specific experts' opinion variables on quality of stock price prediction can be a promising research endeavour. Finally, combining different soft computing approaches with hybridized market indicators that include experts' opinions look promising to further improve time series forecasting.

APPENDIX: LIST OF PUBLICATIONS

Refereed Conference Proceedings

1. **Adebiyi, A.A**., Ayo, C.K. and Otokiti S.O. (2009) "Stock Price Prediction using Hybridized Market Indicators", Proceedings of International Conference of Artificial Intelligence and Pattern Recognition, MultConf'2009, USA, pp.372-379.

Journal Articles

1. **Adebiyi, A.A**., Ayo, C.K. and Otokiti S.O. (2010) "An Improved Model for Stock Price Prediction using Market Experts Opinion", International Journal of Computer Science and Systems, Vol. 1, No.3, pp.28-39.
2. **Adebiyi, A.A**., Ayo, C.K. and Otokiti S.O. (2010) "Stock Price Prediction Using Fuzzy-neural Approach", International Journal of Emerging Trends in Engineering and Applied Sciences, Vol. 1, No.1, pp.9-13.
3. **Adebiyi, A.A**., Ayo, C.K. and Otokiti S.O. (2011) "Fuzzy-neural model with hybrid market indicators for stock forecasting", International Journal of Electronic Finance (Inderscience Journal Series), Vol. 5, No.3, pp. 286-297.

REFERENCES

Abu-Mostafa Y.S., Atiya A.F., Magdon-Ismail M., and White H. (2001) "Neural Networks in Financial Engineering", *IEEE Transactions on Neural Networks* 12(4): pp. 653-656.

Adebiyi, A.A., Ayo, C.K. and Otokiti, S.O. (2009) 'Stock Price Prediction using Hybridized Market Indicators', *Proceedings of International Conference on Artificial Intelligence and Pattern Recognition*, MutiConf'09, Orlando USA, pp. 362-367.

Aiken, M. and M. Bsat. (1999) "Forecasting Market Trends with Neural Networks" *Information Systems Management*, 16(4), pp.42-48.

Ajith, A., Sajith, N., & Sarathchandran, P. P. (2003) "Modelling Chaotic Behaviour of Stock indices using Intelligent Paradigms", *Neural, Parallel & Scientific Computations Archive*, 11, 143–160.

Alain N.K. (2002) "Macroeconomic Forecasting: A Comparison Between Artificial Neural Networks and Econometric Models", published M.Sc. thesis, Rand Afrikaans University.

Al-Qaheri H., Hassanien A. E., Abraham A (2008) "Discovering stock price prediction rules using rough sets", *Neural Network World*, 18, pp. 181-198.

Atiya, A., Noha Talaat & Samir Shaheen, (1997) "An efficient stock market forecasting model using neural networks", *Proceedings of the IEEE International Conference on Neural Networks*, Houston, TX, USA, 4, pp. 2112-2115.

Avci, E. (2007) "Forecasting Daily and Sessional Returns of the Ise-100 Index with Neural Network Models", *Dogus Universitesi Dergisi*, 2(8), pp.128-142**.**

Atsalakis, G., & Valavanis K. (2006) "Neuro-fuzzy and technical analysis for stock prediction", Working paper, Technical University of Crete.

Ayob, M., Nasrudin, M. F., Omar, K., and Surip, M. (2001) "The Effects of Returns Function on Individual Stock Price (KLSE) Prediction Model using Neural Networks", *Proceedings of the International Conference on Artificial Intelligence*, IC-AI 2001, pp. 409–415.

Barnes, M. B., Rimmer, R. J., & Ting, K. M. (2000) ”A study of techniques for mining data from the Australian stock exchange”, *Proceedings of the Fourth World Multi-conference on Systemics, Cybernetics and Informatics,* 8(2), pp. 52–57.

Bautista, C. C. (2001) “Predicting the Philippine Stock Price Index using artificial neural networks”, UPCBA Discussion Paper No. 0107.

Branco P.J.C.and Dente J.A.(2000) “Design of an ElectroHydraulic System Using Neuro-Fuzzy Techniques” In: Lakhmi C.J. and Martin N.M (Eds), Fusion of Neural Networks, Fuzzy Systems, and Genetic Algorithms Industrial Application, The CRC Press International Series on Computational Intelligence, New York, pp.69-103.

Brownstone D. (1996) “Using Percentage Accuracy to Measure Neural Network Predictions in Stock Market Movements”, *Neurocomputing*, 10, pp. 237-250.

Bruce V. and Gavin F. (2009) “An empirical methodology for developing stock market trading systems using artificial neural networks”, *International Journal of Expert Systems with Applications*, 36(3), pp. 6668-6680.

Chandra, N. & Reeb, D.M. (1999) “Neural Networks in a Market Efficiency Context”, *American Business Review*, pp. 39-44.

Chang, P.C. and Liu, C.H. (2008) “A TSK type fuzzy rule based system for stock price prediction”, *Expert Systems with Applications*, 34(1), pp.135-144.

Chen, A. S., Leung, M. T., and Daouk, H. (2003) “Application of neural networks to an emerging financial market: forecasting and trading the Taiwan Stock Index”, *Journal of Computers & Operations Research*, 30(6), pp. 901-923.

Chen, Y., Yang, B. and Abraham, A. (2005) “Time-series forecasting using flexible neural tree model”, *Journal of Information Sciences*, 174(4), pp. 219-235.

Chen, Y., Abraham, A., Yang, J., & Yang, B. (2005) “Hybrid methods for stock index Modeling”, *Proceedings of Fuzzy Systems and Knowledge Discovery: Second International Conference*, pp. 1067–1070.

Chen, Y., Dong, X., & Zhao, Y. (2005) “Stock index modelling using EDA based local linear wavelet neural network”, *Proceedings of International Conference on Neural Networks and Brain*, pp. 1646–1650.

Chi F.L (1998) “Forecasting Tourism; A Combined Approach”, *Tourism Management*, 19(6), pp. 515-520.

Choi, J.H., Lee, M.K., & Lee, M.W. (1995) ”Trading S&P 500 stock index futures using a neural network”, *Proceedings of the third annual international conference on artificial intelligence applications*, New York, pp. 63–72.

Chun, S., & Park, Y. (2005) “Dynamic adaptive ensemble case-based reasoning: Application to stock market prediction”, *Expert Systems with Applications*, 28, 435–443.

De Leone R., Marchitto E., Quaranta A. G. (2006) “Autoregression and artificial neural networks for financial market forecast”, *Neural Network World*, 16, pp. 109-128.

Defu Z., Qingshan J. and Xin Li (2005) “Application of Neural Networks in Financial Data Mining”, *Proceedings of World Academy of Science, Engineering and Technology*, 1, pp. 136-139 ISSN 1307-6884.

Doesken, B., Abraham, A., Thomas, J., & Paprzycki, M. (2005) “Real stock trading using soft computing models”, *Proceedings of International Symposium on Information Technology: Coding and Computing ITCC*, 2, 162–167.

Duke, L. S., & Long, J. A. (1993) “Neural network futures trading—feasibility study”. In: Society for worldwide interbank financial telecomunications, adaptive intelligent systems pp. 121–132.

Emdad K. (2000) “Neural Fuzzy Based Intelligent Systems and Applications” In: Lakhmi C.J. and Martin N.M (Eds), Fusion of Neural Networks, Fuzzy Systems, and Genetic Algorithms Industrial Application, The CRC Press International Series on Computational Intelligence, New York, pp.107-139.

Esmaeil, H., Hassan, S. and Arash, G. (2010) “Integration of genetic-fuzzy systems and artificial neural networks for stock price forecasting”, *Knowledge-Based System*, 23(8), pp.800–808.

Gately, E. (1996), “Neural networks for financial forecasting”, 1st Edition, New York: Wiley

George, S.A. and Kimon, P.V. (2009) “Forecasting stock market short-term trends using a neuro-fuzzy methodology”, *Expert Systems with Applications*, 36(7), pp.10696–10707.

Gerasimo M., Konstantions P., Yannis T., and Babis T., (2005) "Intelligent Stock Market Assistant using Temporal Data Mining", P*roceedings 10th Panhellenics Conference in Informatics (PCI05), Volos, Greece*

Giordano, F., La Rocca, M., and Perna, C. (2007) "Forecasting nonlinear time series with neural networks sieve bootstrap", *Journal of Computational Statistics and Data Analysis*, 51, pp. 3871-3884.

Gómez, V. and Maravall, A. (1998) "Automatic Modelling Methods for Univariate Series", *Banco de España Working Paper* No. 9808.

Hassan M.D. (2007) "Hybrid HMM and Soft Computing Modeling with application to time series analysis", Ph.d thesis, University of Melbourne.

Haykin, S. (1994) *Neural Networks: A Comprehensive Foundation*, Macmillian College Publishing Company, 1st Edition, New York.

Hiemstra, Y. (1995) "Modeling structured nonlinear knowledge to predict stock market returns". In R. R. Trippi (Ed.), Chaos & nonlinear dynamics in the financial markets: Theory, evidence and applications, pp. 163–175, Chicago, IL: Irwin.

Hornik, K., Stinchcombe, M. and White, H. (1989) "Multilayer feedforeward networks are universal approximators", *Neural Networks*, 2, pp. 359-366.

Huang, W., Nakamori, Y. and Wang, S. (2005) 'Forecasting stock market movement with support vector machine', *Journal of Computers and Operations Research*, Vol. 32, No. 10, pp.2513.

Hui, S. C., Yap, M. T., & Prakash, P. (2000) "A hybrid time lagged network for predicting stock prices", *International Journal of the Computer, the Internet and Management*, 8(3).

Jain, A. and Kumar, A.M. (2007) 'Hybrid neural networks models for hydrologic time series forecasting', *Journal of Applied Soft Computing*, 7, pp. 585-592.

Jang, G. & Lai, F. (1994) "Intelligent Trading of an Emerging Market", In: *Trading on the Edge*, Deboeck, G.J. (Ed.), John Wiley & Sons Inc., pp. 80-101.

Kaastra I. and Boyd M. (1996) "Designing a neural network for forecasting financial and economic time series", *Neurocomputing*, 10(3), pp. 215-236.

Kamijo, K. and Tanigawa, T. (1990) "Stock Price Pattern Recognition: A Recurrent Neural Network Approach", *Proceedings of the IEEE International Joint Conference on Neural Networks*, pp. 215-221.

Karray F.O. and De Silva C. (2004) Soft Computing and Intelligent Systems Design: Theory, Tools and Application, 1st edition, England Pearson Education Limited, pp. 223.

Kate A.S., and Gupta J.N.D. (2000) "Neural networks in business: techniques and applications for the operations researcher", *Computers & Operations Research*, 27, pp. 1023–1044

Khan A. U. Bandopadhyaya T.K. and Sharma S. (2008) "Comparisons of Stock Rates Prediction Accuracy Using Different Technical Indicators with Backpropagation Neural Network and Genetic Algorithm Based Backpropagation Neural Network", *Proceedings of International Conference of Emerging Trends in Engineering and Technology*, Nagpur, Maharashtra, 1, pp. 575-580.

Khashei, M.,. Hejazi S.R., and Bijari, M (2008) "A new hybrid artificial neural networks and fuzzy regression model for time series forecasting", *Fuzzy Sets and System*, 259(7), pp. 769-786.

Khashei, M., Bijari, M., and Ardali, G.A.R (2009) "Improvement of Auto-Regressive Integrated Moving Average models using Fuzzy logic and Artificial Neural Networks (ANNs)", *International Journal of Neurocomputing*, 72, pp. 956-967.

Kim, H. J., and Shin, K. S.(2007) 'A hybrid approach based on neural networks and genetic algorithms for detecting temporal patterns in stock markets', *Applied Soft Computing*, 7, pp. 569–576.

Kim, S.H. & Chung, S.H. (1998) "Graded forecasting using array of bipolar predictions: application of probabilistic neural networks to a stock market index", *International Journal of Forecasting,* 14, pp. 323-337.

Kimoto, T., Asakawa, K., Yoda, M. and Tajeoka, M.(1990) "Stock Market Prediction System with Modular Neural Networks", *Proceedings of the IEEE International Joint Conference, on Neural Networks*, pp. 1-16.

Kohara, K, Ishikawa, T., Fukuhara, Y. & Nakamura, Y. (1997) "Stock price prediction using prior knowledge and neural networks", *Intelligent Systems in Accounting, Finance and Management,* 6, pp. 11-22.

Kohavi R. and Provost F. (1998) "Glossary of terms.," *Machine Learning*, 30, pp. 271–274.

Kryzanowski, L., Galler, M. & Wright, D., W. (1993) “Using artificial neural networks to pick stocks”, *Financial Analysts Journal*, 49/4, pp. 21-27.

Kunhuang, H., and Yu, T. H. K. (2006) “The application of neural networks to forecast fuzzy time series”, *Physical A: Statistical Mechanics and Its Applications*, 363(2), pp. 481-491.

Kyoung-Jae K. (2006), “Artificial neural networks with evolutionary instance selection for financial forecasting”, *Expert Systems with Applications*, 30(3), pp. 519-526.

Kyoung-Jae K. and Lee B. (2004), “Stock market prediction using artificial neural networks with optimal feature transformation”, *Neural Computing and Applications*, pp. 255 - 260

Kyungjoo, L., Sehwan, Y., and John, J.J. (2007) ”Neural Network Model vs. SARIMA model in Forecasting Korean Stock Price Index”, *Issues in Information Systems*, 8(2), pp. 372-378.

Lam, M. (2004) “Neural Network techniques for financial performance prediction: integrating fundamental and technical analysis”, *Decision Support Systems*, 37, pp. 565-581.

Lay-Ki S. and Sang H. L. (2007), “Explorative Data Mining on Stock Data – Experimental Results and Findings”, pp. 562-569.

Leigh, W., Paz, M., & Purvis, R. (2002) “An analysis of a hybrid neural network and pattern recognition technique for predicting short-term increases in the NYSE Composite Index”, *Omega*, 30, pp. 69–76.

Li, R.J. (2005) “Forecasting stock market with fuzzy neural networks”, *Proceedings of the Fourth International Conference on Machine Learning and Cybernetics*, Guangzhou, pp. 18-21.

Lipinski P. (2005) “Clustering of large number of stock market trading rules”, *Neural Network World*, 15, pp. 351-357.

Massimiliano V., Rushi B., Oliver H. and Mark S. (2004) “Predicting the exchange traded fund DIA with a combination of genetic algorithms and neural networks”, *International Journal of Expert Systems with Applications*, 27(3), pp. 417-425.

Mehdi K. and Mehdi B. (2010) “An artificial neural network (p,d,q) model for timeseries forecasting”, *Journal of Expert Systems with Applications*, 37, pp.479-489.

Meyler, A., Kenny G. and Quinn, T. (1998). “Forecasting Irish Inflation using ARIMA Models”, Central Bank of Ireland Research Department, Technical Paper, 3/RT/98.

Mills, T., (1993) *The Econometric Modelling of Financial Time Series*, Cambridge University Press: Cambridge.

Mitra S. K. (2009) “Optimal Combination of Trading Rules Using Neural Networks”, *International Business Research*, 2(1), pp. 86-99.

Mohamed, M.M. (2010) “Forecasting stock exchange movements using neural networks: empirical evidence from Kuwait”, *Expert Systems with Applications*, 27(9), pp.6302–6309.

Nikola K. K. (1998) “Foundation of Neural Networks, Fuzzy System Systems and Knowledge Engineering”, The MIT Press, Cambridge, Massachusetts, London, England.

Nikolopoulos, C., & Fellrath, P. (1994) “A hybrid expert system for investment advising”, *Expert Systems with Applications*, 11(4), 245–250.

Nishina, T., & Hagiwara, M. (1997) “Fuzzy interface neural network”, *Neurocomputing*, 14, pp. 223–239.

Nygren K. (2004) “Stock Prediction – A Neural Network Approach”, Master thesis

O'Connor, N. and Maddem, M. G. (2006) ‘A neural network approach to predicting stock exchange movements using external factors, Applications and innovations in intelligent network to investment analysis’, *Financial Analysts Journal*, pp. 78-80.

Pai, P.-F., and Lin, C. –S. (2005) “A Hybrid ARIMA and support vector machines model in stock price forecasting”, *Omega*, 33(6), 497–505.

Pan, H., Tilakarante, C., and Yearwood, J. (2005) “Predicting Australian stock market index using neural networks exploiting dynamical swings and intermarket influences”, *Journal of Research and Practice in Information Technology*, 37(1).

Pantazopoulos, K. N., Tsoukalas, L. H., Bourbakis, N. G., Bruen, M. J., & Houstis, N. (1998) “Financial prediction and trading strategies using neuro-fuzzy approaches”, *IEEE Transactions on Systems Man and Cybernetics – Part B*, 28(4).

Pegal F. (2007), “Stock Trend Prediction Using News Article, A Text Mining Approach”, Master thesis.

Phua, P.K.H., Ming, D. & Lin, W. (2001) “Neural network with genetically evolved algorithms for stock prediction”, *Asia-Pacific Journal of Operational Research*, 18, pp. 103-107.

Phua, P. K. H, Zhu, X. T., and Chung, H.K.(2003) “Forecasting Stock Index Increments Using Neural Networks with Trust Region Methods”, *Proceedings of the International Joint Conference on Neural Networks*, 1, pp. 260-265.

Perez-Rodriguez, J. V., Torrab, S., & Andrada-Felixa, J. (2005) “STAR and ANN models: Forecasting performance on the Spanish Ibex-35 stock index”, *Journal of Empirical Finance*, 12(3), pp. 490–509.

Quah, T.-S., and Srinivasan, B. (1999) “Improving returns on stock investment through neural network selection”, *Expert Systems with Applications*, 17, 295–301.

Ramon L.(1997). “Using Neural Networks to Forecast Stock Market Prices”, http://people.ok.ubc.ca./rlawrence/research/papers/nn.pdf

Raposo, R., De, A. J., and Cruz, O. (2002) “Stock market prediction based on fundamentalist analysis with fuzzy-neural networks”, *Proceedings of 3rd WSES International Conference on Fuzzy Sets & Fuzzy Systems (FSFS’02), Neural Networks and Applications (NNA’02), Evolutionary Computation (EC’02).*

Raymond S. T. L. (2004) “iJADE Stock Advisor: An Intelligent Agent Based Stock Prediction System Using Hybrid RBF Recurrent Network” *IEEE Transactions on Systems, Man, and Cybernetics—Part A: Systems and Humans*, 34(3), pp. 421-428.

Rech, G. (2002) Forecasting with artificial neural network models. SSE/EFI Working Paper Series in Economics and Finance, No. 491.

Refenes, A. N., Azeme-Barac, M., & Zapranis, A. D. (1993). “Stock ranking: Neural networks vs. multiple linear regression”, *Proceedings of IEEE ICNN*.

Refenes A.P,(ed.) (1995), “Neural network in the capital markets”, John Wiley& Sons Ltd.

Robert F. (1995) “Neural Fuzzy System”, Abo Akademic University, ISBN 951-650-624-0, ISSN 0358-5654, pp. 153.

Roh, T. H. (2007) “Forecasting the Volatility of Stock Price Index”, *Journal of Expert Systems with Applications*, 33, pp. 916-922.

Rohit and Kumkum (2008) “A Hybrid Machine Learning System for Stock Market Forecasting”, *Journal of World Academy of Science, Engineering and Technology*, 39, pp. 315-318.

Saad, E.W., Prokhorov, D.V. & Wunsch, D.C. (1998) “Comparative study of stock trend prediction using time delay, recurrent and probabilistic neural networks”, *IEEE Transactions on Neural Networks*, 9(6), pp. 1456-1470.

Schoeneburg, E.(1990), “Stock Price Prediction Using Neural Networks: A Project Report”, *Neurocomputing*, 2, pp. 17-27.

Sfetsos A. (2002), “The application of neural logic networks in time series forecasting”, *Neural Network World*, 12, pp. 181-199

Shakih A. H. (2004), “Primer on using Neural Networks for Forecasting Market Variables”, [online] accessed 4th January, 2008.

Stansel, S. and Eakins, S. (2004) ‘Forecasting the direction of change in sector stock indexes: An application of neural networks’, *Journal of Asset Management*, 5(1), pp.37-48.

Stergious C. and Siganos D. (2009) “Neural Networks”, www. Doc.ic.ac.uk~nd/surprise_96/journal/vol4/cs11/report.html [Access online, March 2010]

Sun, Y.F., Liang, Y.C., Zhang, W.L., Lee, H.P., Lin, W.Z. & Cao, L.J. (2005) “Optimal partition algorithm of the RBF neural network and its application to financial time series forecasting”, *Neural Computation & Application*, 14, pp. 36–44

Tabachnick, B.G. and Fidell L.S.(2001) Using multivariate statistics, 4th ed., Person Education Company, USA.

Tae H. R. (2007) “Forecasting the Volatility of Stock Price Index”, *Journal of Expert Systems with Applications*, vol. 33 pp. 916-922.

Tiffany, H. and Kun-Huang, H. (2010) “A neural network-based fuzzy time series model to improve forecasting”, *Expert Systems with Applications*, 37, pp.3366–3372.

Trippi, R. R., & DeSieno, D. (1992) “Trading equity index futures with a neural network”, *Journal of Portfolio Management*, 19, pp. 27–33.

Tsanga, P.M., Kwoka, P., Choya, S.O., Kwanb, R., Nga, S.C., Maka, J., Tsangc, J., Koongd K. and Wong, T.L. (2007) "Design and Implementation of NN5 for Hong Stock Price Forecasting", *Journal of Engineering Applications of Artificial Intelligence*, 20, pp. 453-461.

Tseng F.M., Yu H.C., and Tzeng G.H. (2002) "Combining neural network model with seasonal time series ARIMA model", *Technological Forecasting and Social Change*, 69, pp. 71-87.

Valenzuela O., Rojas I., Rojas F., Pomares H., Herrera L.J., Guillen A., Marquez L., and Pasadas M. (2008), **"**Hybridization of intelligent techniques and ARIMA models for time series prediction**",** *Fuzzy Sets and Systems*, 157(7), pp. 821-845.

Wang, Y. H. (2007) "Nonlinear neural network forecasting model for stock index option price: Hybrid GJR–GARCH approach", *Journal of Expert Systems with Applications*, 36(1), pp. 564-570.

Wang, J.H., and Leu, J.Y. (1996) "Stock market trend prediction using ARIMA-based neural networks", *Proceedings of IEEE International Conference on Neural Networks*, 4, pp. 2160–2165.

Weckman, G.R., Lakshminarayanan, S., Marvel, J.H., Snow, A. (2008) "An integrated stock market forecasting model using neural networks", *Int. J. Business Forecasting and Marketing Intelligence*, 1(1), pp.30–49.

Wei C. H. (2005) "Hybrid Learning Fuzzy Neural Models in Stock Forecasting", *Journal of Information and Optimization Sciences*, 26(3), pp. 495-508.

White, H. (1988) "Economic prediction using neural networks: The case of IBM daily stock returns", *Proceedings of the IEEE International Conference on Neural Networks*, pp. 451-458.

Widrow, B. Rumelhart, D.E. and Lehr, M.A.(1994) "Neural networks: applications in industry, business and science", *Communications of the ACM,* 37(3), pp. 93–105.

Wong, F.S., Wang, P.Z., Goh, T.H. & Quek, B.K., (1992) "Fuzzy neural systems for stock selection", *Financial Analysts Journal*, 48, pp. 47-52.

Xiaodan W., Ming F. and Andrew F. (2001) "Forecasting Stock Performance Using Intelligent Hybrid Systems" *Springerlink*, 2, pp. 447-456.

Yan, S. (2008) “A Novel Prediction Method for Stock Index Applying Grey Theory and Neural Networks”, *The 7th International Symposium on Operations Research and Its Applications (ISORA’08),* pp.104-111.

Yang, Q. and Wu, X. (2006) “10 Challenging Problems in Data Mining Research”, *International Journal of Information Technology and Decision Making*, 5(4), pp. 1-8.

Yao J. T., and Poh H.-L. (1995) “Equity Forecasting: A Case Study on the KLSE Index”, *Proceedings 3rd of International Conference on Neural Networks in the Capital Markets, London.*

Yao J. T., Tan C. L. and Poh H.-L.(1999) “Neural networks for technical Analysis: A study on KLCI”, *International Journal of Theoretical and Applied Finance,* 2(2), pp. 221–241

Yoda. M, (1994) “Predicting the Tokyo Stock Market”. In: *Trading on the Edge*, Deboeck, G.J. (Ed.), John Wiley & Sons Inc., pp. 66-79.

Zhang G.P. (2004) ‘Business Forecasting with Artificial Neural Networks: An Overview”. *Neural networks in Business Forecasting*, pp. 1-22, Georgia State University, USA

Zhang, Y.Q., Akkaladevi, S., Vachtsevanos, G., and Lin, T. Y. (2002) “Granular neural web agents for stock prediction”, *Soft Computing*, 6, pp. 406–431.

Zhang, D., Jiang, Q., and Li, X. (2004) “Application of neural networks in financial data mining”, *Proceedings of International Conference on Computational Intelligence*, pp. 392–395.

Zhang, G., Patuwo B., and Hu M.Y. (1998) “Forecasting with artificial neural networks: The state of the art”, *International Journal of Forecasting*, 14, pp. 35-62.

Zhongxing, Y., and Liting, G. (1993) “A hybrid cognition system: Application to stock market analysis”, *Proceedings of IEEE International Joint Conference on Neural Networks*, pp. 3000–3003.

Zhu, X., Wang, H., Xu, L. and Li, H. (2007) “Predicting stock index increments by neural networks: The role of trading volume under different horizons”, *Journal of Expert Systems with Applications*, 34(4), pp. 3043-3054.