Minor Project Report

on

Stock Market Analysis and Prediction using NARIMA

Submitted to the
Indian Institute of Information Technology, Pune
In partial fulfilment for the award of the Degree of
Bachelor of Technology in Computer Science Engineering

By

111615010	Dhornala Bharadwaj
111615013	Dheeraj Jutury

Under the guidance of

Dr. Jayadeep Pati

Department of Computer Science and Engineering
Indian Institute of Information Technology, Pune
Talegaon-Chakan Road, Sudumbre, Pune-412109, Maharashtra

(2018 - 2019)

Indian Institute of Information Technology, Pune

Department of Computer Science and Engineering

CERTIFICATE

This is to certify that

Dhornala Bharadwaj (111615010)
Dheeraj Jutury (111615013)

Have successfully completed Minor Project Work on

Stock Market Analysis and Prediction using NARIMA

Towards the partial fulfillment of Degree of
Bachelor of Technology in Computer Science Engineering
in Semester-V of the academic year 2018-2019.

Dr.Jayadeep Pati
[Minor Project Guide]

Dr. Darshan V. Medhane

[Faculty in Charge-Minor Project Work]

Dr. Sourabh Shrivastava

[Head, Dept. of CSE]

Place: Pune
Date: 30th November 2018

Acknowledgement

We are highly indebted to our guide **Dr. Jayadeep Pati** for his guidance and constant supervision as well as for providing necessary information regarding the report and also for the support in completing the report.

We would also like to express our gratitude towards our parents and friends for their kind co-operation and encouragement which help us in completion of this report. Our thanks and appreciations also go to our colleague in developing the report and people who have willingly helped us out with their abilities.

Dhornala Bharadwaj (111615010)

Dheeraj Jutury (111615013)

Abstract

Nowadays, the stock market is attracting more and more people's notice with its high challenging risks and high return over. A stock exchange market depicts savings and investments that are advantageous to increase the effectiveness of the national economy. The future stock returns have some predictive relationships with the publicly available information of present and historical stock market indices.

ARIMA is a statistical model which is known to be efficient for time series forecasting.In this paper, we propose a model for forecasting the stock market trends based on the technical analysis using historical stock market data and ARIMA model. The results are shown in terms of visualizations using R programming language. The obtained results reveal that the ARIMA model has a strong potential for prediction of stock market trends.

This paper aims to combine the conventional time series analysis technique with information from the SPX dataset to predict daily changes in stock price.Important news/events related to a SPX stock over a two-year span are used to provide a measure of the magnitude of these events.The result of this experiment shows significant correlation between the changes in daily stock prices and the values of important news/events computed from the SPX dataset.

What we are doing??

We intend to predict the stock values using the best model which minimizes the error.

1. Using ARIMA model,we make predictions for the stock values.
2. To minimize the error obtained in ARIMA model,we produce the results of ARIMA into a neural network.
3. Using neural network we try to minimize the error.

Contents

List of Figures

List of Tables

Chapter 1

INTRODUCTION

This chapter gives a brief description about our project, and motivation behind designing this project.

1.1 Basics

Today we live and breathe data. Forecasting the stock exchange data is an important financial subject which involves an assumption that the fundamental information publicly available in the past has some predictive relationships to the future stock returns. Stock market forecasting contains uncovering the market trends, planning investment tactics, identifying the best time to purchase the stocks and which stocks to purchase. A stock exchange or equity business sector is a non-direct, non-parametric framework that is difficult to model with any sensible exactness. It is the mix of speculators who need to purchase or offer or hold a share at a specific time. Prediction will continue to be an exciting locale of research, making scientists in the analytics field always desiring to enhance the existing forecasting models.

R is a programming language and environment for statistical processing and graphics. The R dialect is generally utilized among analysts and data excavators for statistical programming and data analysis.Around the world, a large number of analysts and information researchers use R language to take care of their most difficult issues in the fields going from computational science to quantitative promoting. R-Studio is a free and powerful integrated development environment for R language. R-Studio allows the user to run R scripts in a more user-friendly environment.

1.2 Motivation

Stock market price prediction is a problem that has the potential to be worth billions of dollars and is actively researched by the largest financial corporations in the world. It is a significant problem because it has no clear solution, although attempts can be made at approximation using many different machine learning techniques. The project allows techniques for real-world machine learning applications including acquiring and analyzing a large data set and using a variety of techniques to train the program and predict potential outcomes. The motivated idea is that, if we know all information about todays stock trading, the price is predictable.Thus, if we can obtain just a partial information, we can expect to improve the current prediction lot.

1.3 Objectives of Work

The Objectives of our project are as follows:

1. Predicting future stock values and therefore making informed decisions on that basis.
2. View the current status of the stock market by providing charts, graphs , news feeds and other research tools.
3. To identify factors affecting share market.

1.4 Organization of the Report

The report begins with chapter 1, the brief introduction to project topic followed by the chapter 2, related work and project planning. The report further gives the detailed information about the proposed system with detailed system analysis in chapter 3 and system design with UML diagrams in chapter 4. Chapter 5 briefs about implementation details. The details of experimental setup, testing (test cases) and overall performance evaluation are discussed in chapter 6. Chapter 7 concludes the report.

Chapter 2

RELATED WORK AND PROJECT PLANNING

This chapter gives brief information about project domain and related work.

Predicting the Stock Market has been the bane and goal of investors since its existence. Everyday billions of dollars are traded on the exchange, and behind each dollar is an investor hoping to profit in one way or another. Entire companies rise and fall daily based on the behaviour of the market. Should an investor be able to accurately predict market movements, it offers a tantalizing promises of wealth and influence. It also referred to as Box-Jenkins methodology composed of set of activities for identifying, estimating and diagnosing ARIMA models with time series data. The model is most prominent methods in financial forecasting [1, 12, 9]. ARIMA models have shown efficient capability to generate short-term forecasts. It constantly outperformed complex structural models in short-term prediction [17]. In ARIMA model, the future value of a variable is a linear combination of past values and past errors, expressed as follows:

$$\hat{y}_t = \mu + \phi_1 y_{t-1} + \ldots + \phi_p y_{t-p} - \theta_1 e_{t-1} - \ldots - \theta_q e_{t-q}$$

Figure 2.1: Arima formula

where, Yt is the actual value and t is the random error at t, i and theta j are the coefficients, p and q are integers that are often referred to as autoregressive and moving average, respectively.

Project planning was divided into two parts- Trainig and Testing using ARIMA and Neural network.we take 80/20 percent for train and for test.Estimate values using ARIMA and use it as input to Neural network.Predict the values using Neural network and minimize the error.

The technologies used-

1. R Studio :popular open source and enterprise-ready professional software for the R statistical computing environment.

2.1 Detailed Project Plan

2.1.1 Gantt Chart

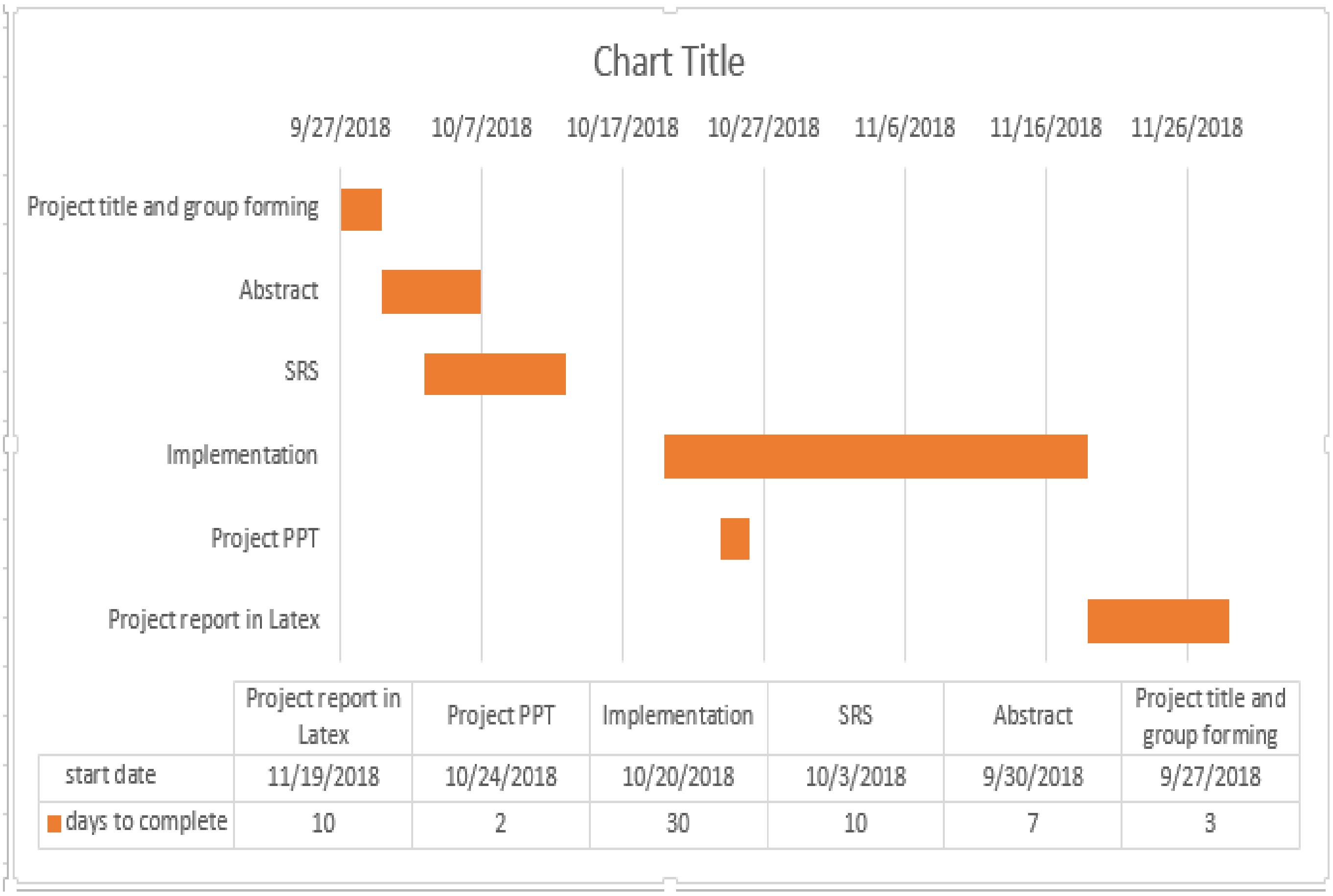

Figure 2.2: Arima formula

Chapter 3

SYSTEM ANALYSIS

This chapter gives detailed information about system, purpose, scope, technologies to be used. It also contains hardware requirements and the documentation of project.

3.1 Introduction

3.1.1 Purpose

Stock market prediction is the act of trying to determine the future value of a company stock or other financial instrument traded on an exchange. The successful prediction of a stock's future price could yield significant profit. The efficient-market hypothesis suggests that stock prices reflect all currently available information and any price changes that are not based on newly revealed information thus are inherently unpredictable. Others disagree and those with this viewpoint possess myriad methods and technologies which purportedly allow them to gain future price information.

3.1.2 Scope

Analysis of stocks using data mining will be useful for new investors to invest in stock market based on the various factors considered by the software.Stock market includes daily activities like sensex calculation, exchange of shares. The exchange provides an efficient and transparent market for trading in equity, debt instruments and derivatives.Our software will be analyzing stocks based on SPXs stock value and predict values which minimize the error between predicted values and actual values.

3.1.3 Definition, Acronyms and Abbreviations

Acronyms

- ARIMA : AutoRegressive Integrated Moving Average model

- AR : AutoRegressive model
- MA : Moving Average model
- ACF : Auto Co-relation Funtion
- PACF : Partial Auto Co-relation Function

Abbreviations

- Feed Forward Neural Network (FFNN)

Definitions

- Stock Market Prediction : Stock market prediction is the act of trying to determine the future value of a company stock or other financial instrument traded on an exchange. The successful prediction of a stock's future price could yield significant profit. The efficient-market hypothesis suggests that stock prices reflect all currently available information and any price changes that are not based on newly revealed information thus are inherently unpredictable.
- Stock : The capital stock (or stock) of an incorporated business constitutes the equity stake of its owners. It represents the residual assets of the company that would be due to stockholders after discharge of all senior claims such as secured and unsecured debt. Stockholders' equity cannot be withdrawn from the company in a way that is intended to be detrimental to the company's creditors.
- Stock Market : A stock market or equity market is a public entity (a loose network of economic transactions, not a physical facility or discrete entity) for the trading of company stock (shares) and derivatives at an agreed price; these are securities listed on a stock exchange as well as those only traded privately.
- ARIMA : ARIMA stands for Autoregressive Integrated Moving Average models. Univariate (single vector) ARIMA is a forecasting technique that projects the future values of a series based entirely on its own inertia. Its main application is in the area of short term forecasting requiring at least 40 historical data points. It works best when your data exhibits a stable or consistent pattern over time with a minimum amount of outliers.
- Activation Function : An activation function is the function in an artificial neuron that delivers an output based on inputs. Activation functions in artificial neurons are an important part of the role that the artificial neurons play in modern artificial neural networks.

- Neural Network : An Artificial Neural Network (ANN) 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 (neurones) working in unison to solve specific problems.

3.1.4 Technologies Used

1. R Studio :popular open source and enterprise-read professional software for the R statistical computing environment.

3.2 Overall Description

3.2.1 User Characteristic

User need to perform following two actions:

- Browse: User will have to browse the dataset from Google Finance if not provided.The getSymbols() function in quantmod package will allow the user to browse the dataset from Google Finance.

- Upload: User will have to upload the dataset browsed earlier in order to upload it to the ARIMA model for testing and prediction.

3.2.2 Hardware Requirements

There is no special hardware requirement for this project. So we considered :

SYSTEM: Intel Core i5 7th Gen

HARD DISK: 1TB

RAM: 8GB

3.2.3 Product Functionality: Use case documentation

Actors :
User
Network
Use case and description:
The user will provide dataset samples for training and testing for part1 (using ARIMA) and part2(Neural network).
Part-1

Training: The ARIMA model will be trained with 80 percent of the dataset provided.

Testing: After training, the ARIMA model will forecast the values.
Part-2

Training: The forecasted and error values of ARIMA are produced in neural network to train.

Testing: After training,the neural network will give the estimated value for stocks.

The use cases shows overall functionality of project. The UML diagrams are explained briefly in the next chapter.

3.2.4 Product Functionality: Use case documentation

Use Case Attribute	Values
ID	01
Name	User
Author	Bharadwaj/Dheeraj
Description	To load the dataset
Actors Involved	User
Precondition	Correct dataset with name is required
Post Conditions	Dataset is loaded into R Studio
Limitations	-
Frequency of Use	1

Table 3.1: Use case: Loading the dataset by User

Use Case Attribute	Values
ID	02
Name	ARIMA model
Author	Bharadwaj/Dheeraj
Description	To find the parameters of ARIMA
Actors Involved	-
Precondition	Dataset should be present
Post Conditions	Predict stock values
Limitations	predicts only for short-term values
Frequency of Use	1

Table 3.2: Use case: ARIMA model Execution

Use Case Attribute	Values
ID	03
Name	Training Neural Network
Author	Bharadwaj/Dheeraj
Description	To train on the output values of ARIMA
Actors Involved	-
Precondition	-
Post Conditions	Forecast the values using NN
Limitations	Contains error after calculation
Frequency of Use	1

Table 3.3: Use case: Training Neural Network for forecast

Use Case Attribute	Values
ID	04
Name	Testing Neural Network
Author	Bharadwaj/Dheeraj
Description	To test the predicted values
Actors Involved	-
Precondition	-
Post Conditions	Generate the minimum error
Limitations	Error is obtained
Frequency of Use	1

Table 3.4: Use case: Testing Neural Network

Chapter 4

SYSTEM DESIGN

4.1 Proposed System

To predict the stock values, firstly user will have to load the dataset . Then user will enter the percentages for training and testing the dataset.Training data is passed through different models and values are predicted.Our main aim is to minimize the error calculated between predicted values and actual values. Overall implementation can be summarized by following steps:

1. Using ARIMA: ARIMA(p,d,q) is used to train the dataset and using arima(dataset,order(p,d,q)) function and predict values using predict() function in ARIMA.

2. Using Neural Network: Neural Network takes predicted values and error values of the ARIMA to minimize the error and predict stock values again.NN gives predicted values for the output of ARIMA.

4.2 UML Diagrams

4.2.1 Use case diagram

To model a system, the most important aspect is to capture the dynamic behavior. Dynamic behavior means the behavior of the system when it is running/operating. The following is the use case diagram for our system. The human actors involved are User, Database Administrator.

The actor Database Administrator will be doing various functions such as inserting values into database,deleting values into database,updating values into database. After the database is completely formed dataset is taken from it.

The User will upload the dataset.Trainig of dataset takes place the testing of it,then predict values.

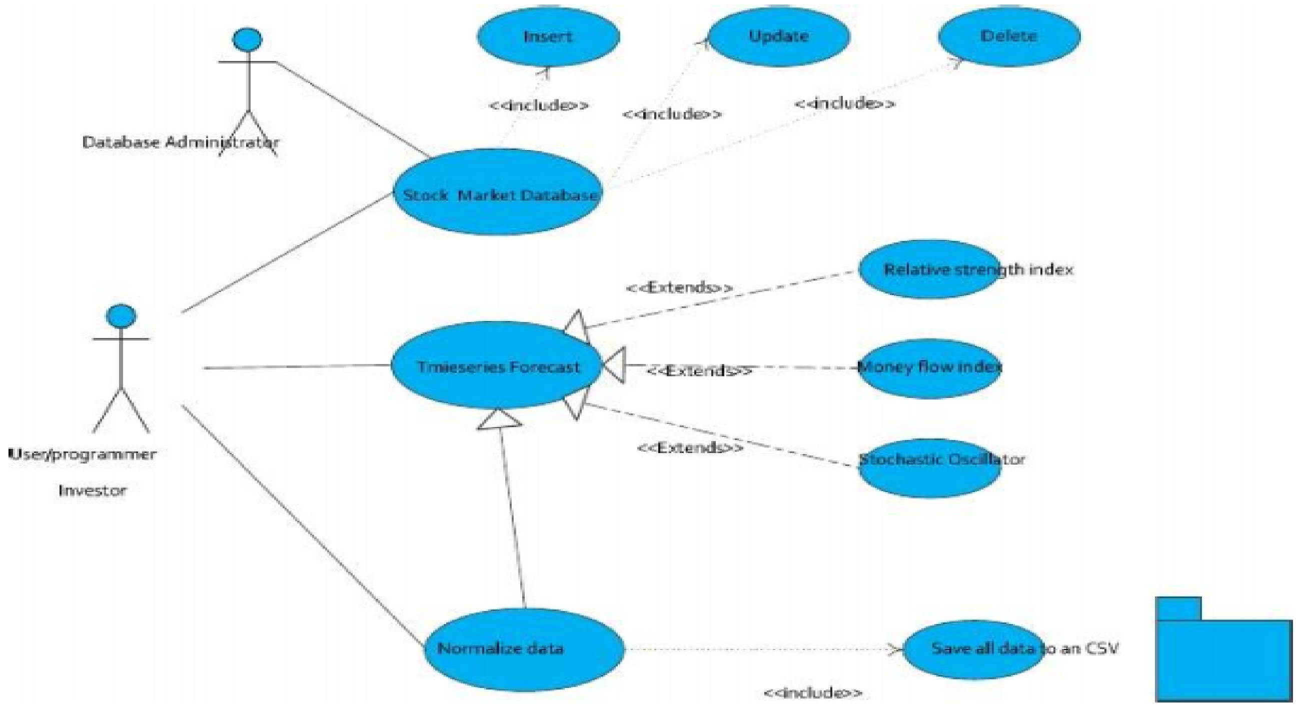

Figure 4.1: Use Case Diagram

4.2.2 Class diagram

Class diagram is a static diagram. It represents the static view of an application. Class diagram describes the attributes and operations of a class and also the constraints imposed on the system. The class diagrams are widely used in the modelling of object oriented systems because they are the only UML diagrams, which can be mapped directly with object-oriented languages. A collection of class diagrams represent the whole system.

Class diagram is also considered as the foundation for component and deployment diagrams. Class diagrams are not only used to visualize the static view of the system but they are also used to construct the executable code for forward and reverse engineering of any system.

It describes how historical stock quotes are taken in technical analysis,Dividends,stocks,assists are taken in fundamental analysis,stock market news and twitter feeds are taken in sentimental analysis.Using technical analysis and fundamental analysis,we train the Artificial Neural Network (ANN) and predicted the stock values.

The following diagram shows class diagram.

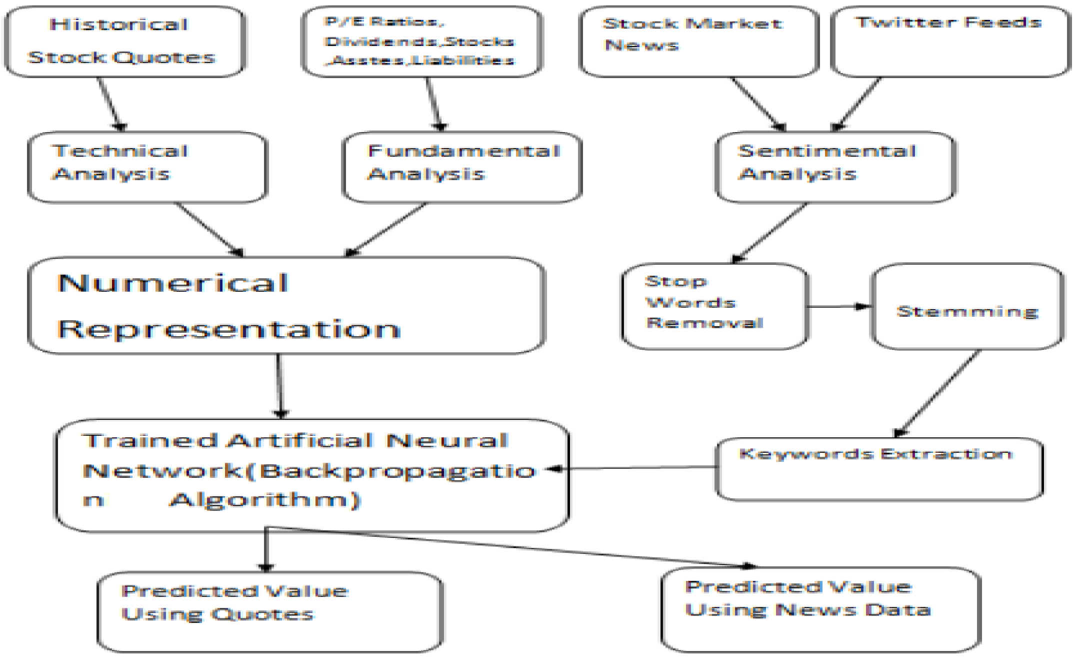

Figure 4.2: Class Diagram

4.2.3 Sequence diagram

Sequence diagram is an interaction diagram that shows how objects interact with one another and in what order. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario.

Sequence diagram contains the sequnce of how input values move from one layer to other in neural network.Our neural network has input layer ,hidden layer and output layer.

The given figure is example of sequence diagram.

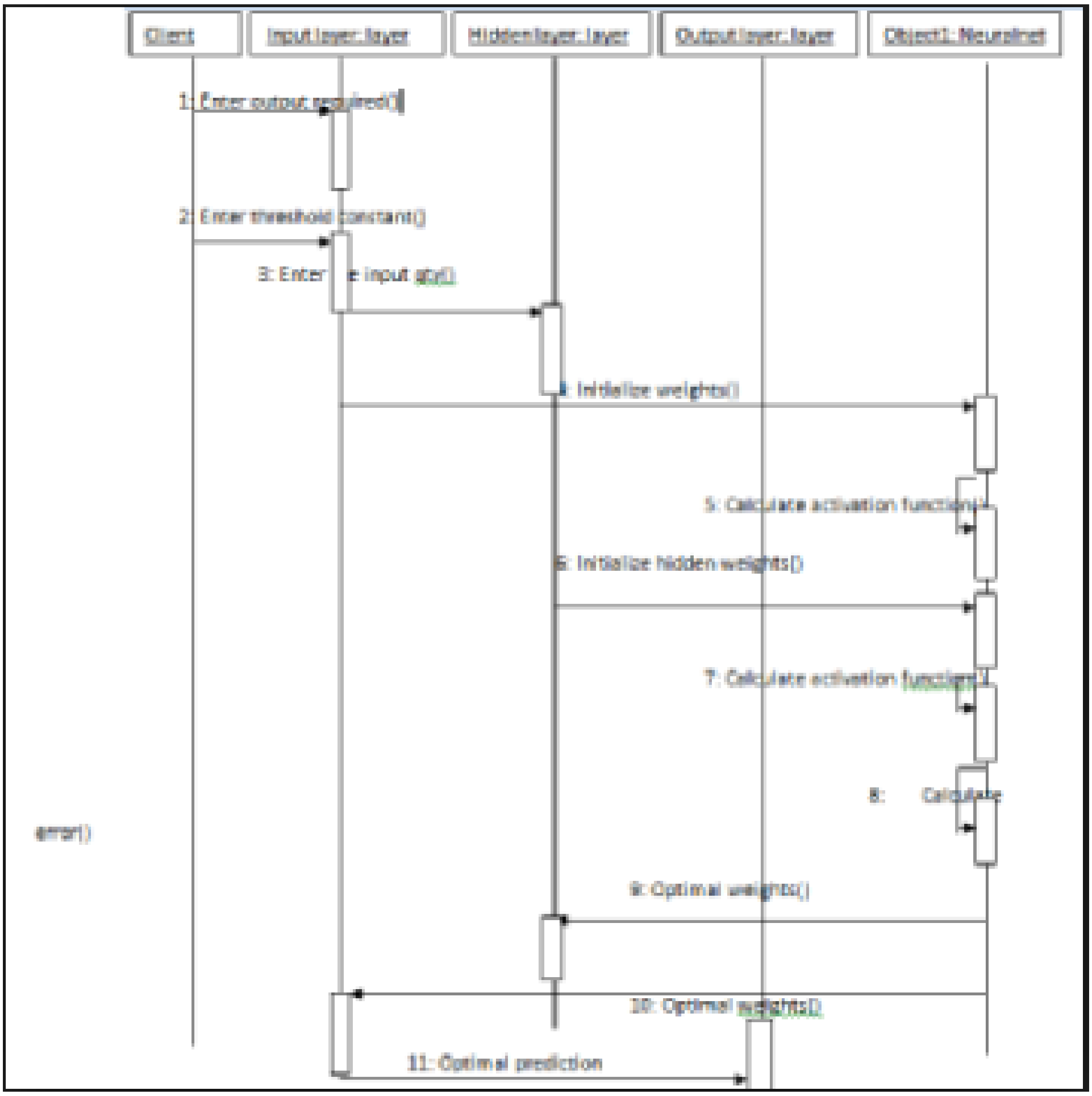

Figure 4.3: Sequence Diagram

4.2.4 Neural Network diagram

An Artificial Neural Network (ANN) 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 (neurones) working in unison to solve specific problems.

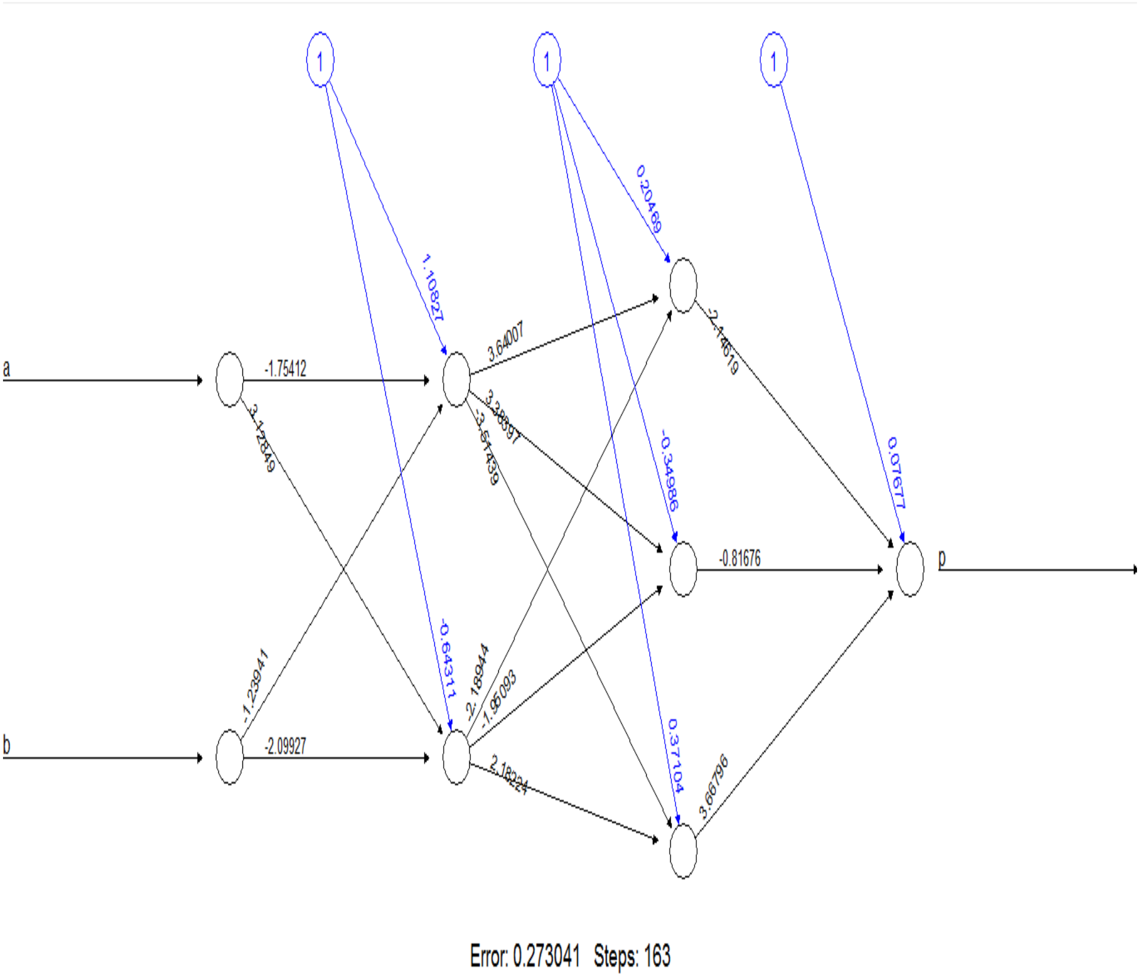

Figure 4.4: Sequence Diagram

Chapter 5

IMPLEMENTATION

Detailed Software Module Description and Lines of code.

```
library(MASS)
library(tseries)
library(forecast)
getwd()
setwd("C:/Users/DHEERAJ/Documents/dataset")
mydata<-read.csv("SPX.csv")
View(mydata)
attach(mydata)
m1<-mydata$Adj.Close[1:500]
m1
logstock=diff(log(m1),lag=1)
logstock
plot(logstock,type='l',main='log returns plot')
difflnstock=diff(logstock)
difflnstock

#ACF,PACF and DIckey Fuller test

adf.test(logstock,alternative="stationary",k=0)
#adf.test(logstock,alternative = "explosive",k=0)

acf(logstock) # #estimate
pacf(logstock,lag.max=20)

acf(difflnstock)
pacf(difflnstock)

arima(m1,order=c(1,0,0))
arima(m1,order=c(2,0,0))
arima(m1,order=c(2,0,2))
#least aic values and ar1 values should not be more or near to
```

```
one
# arima model uing function
fit=arima(m1,order=c(2,0,2))
summary(fit)
tsdiag(fit)
fit
predicted=arimapredit<-predict(fit,n.ahead=100)
predicted
plot(m1)
plot(predicted$pred)
#table1=data.frame(mydata$Close[501:600],predicted
$pred,predicted$se)
#lines(m1,predicted$pred,col="blue")
mdataset=predicted$pred
df=data.frame(mdataset,mydata$Close[501:600])
error=df$mdataset-df$mydata.Close.501.600
mdf=data.frame(mdataset[2:99],mdataset[1:98],error[2:99],error
[1:98],mdataset[3:100])
names(mdf)=c("a","b","c","d","p")
```

```
# Data
getwd()
setwd("C:/Users/DHEERAJ/Documents")
data <- read.csv("mdf.csv", header = TRUE)
data=data[2:6]
data
data1=data[1:2]
data1
data1=data.frame(data1,data$p)
names(data1)=c("a","b","p")
rdata1=data1
rdata=data
rdata
# Min-Max Normalization
data$a <- (data$a - min(data$a))/(max(data$a) - min(data$a))
data$b <- (data$b - min(data$b))/(max(data$b) - min(data$b))
data$c <- (data$c - min(data$c))/(max(data$c) - min(data$c))
data$d <- (data$d - min(data$d))/(max(data$d) -min(data$d))
data$p <- (data$p - min(data$p))/(max(data$p )-min(data$p))
data1$a <- (data1$a - min(data1$a))/(max(data1$a) - min(data1$a))
data1$b <- (data1$b - min(data1$b))/(max(data1$b) - min(data1$b))
data1$p <- (data1$p - min(data1$p))/(max(data1$p )-min(data1$p))

# Data Partition
set.seed(222)
training <- data[1:62,]
testing <- data[63:78,]
train=rdata[1:62,]
test=rdata[63:78,]
```

```
training1 <- data1[1:62,]
testing1<- data1[63:78,]
train1=rdata1[1:62,]
test1=rdata1[63:78,]

# Neural Networks
library(neuralnet)
set.seed(333)
softplus <- function(x) log(1+exp(x))
n <- neuralnet(p~a+b+c+d,act.fct = softplus,
               data = training,
               hidden =c(2,3),

               linear.output = FALSE)
n1 <- neuralnet(p~a+b,
               data = training1,
               hidden =c(2,3),

               linear.output = FALSE)

plot(n)
plot(n1)

# Prediction
pr.nn<- compute(n, testing[,c(1:4)])
pr.nn
pr.nn_=(pr.nn$net.result)*(max(rdata$p)-min(rdata$p))+min(rdata$p)
```

```
pr.nn1<- compute(n1, testing1[,c(1:2)])
pr.nn1
pr.nn1_=(pr.nn1$net.result)*(max(rdata1$p)-min(rdata1$p))+min(rdata1$p)

#summary(foutput)
#testing$p=(testing$p)*(max(rdata$p)-min(rdata$p))+min(rdata$p)
RMSE.nn = (sum((test$p -pr.nn_ )^2) / nrow(testing)) ^ 0.5
MSE.nn <- sum((test$p - pr.nn_)^2)/nrow(testing)
RMSE.nn1 = (sum((test1$p -pr.nn1_ )^2) / nrow(testing1)) ^ 0.5
MSE.nn1 <- sum((test1$p - pr.nn1_)^2)/nrow(testing1)

RMSE.nn_ = (sum((mydata$Close[463:478] -pr.nn_ )^2) / nrow(testing)) ^
0.5
MSE.nn_ <- sum((mydata$Close[463:478] - pr.nn_)^2)/nrow(testing)
RMSE.nn1_ = (sum((mydata$Close[463:478] -pr.nn1_ )^2) / nrow(testing1)) ^
0.5
MSE.nn1_ <- sum((mydata$Close[463:478] - pr.nn1_)^2)/nrow(testing1)

e1=test$p -pr.nn_
e2=test1$p -pr.nn1_
hist(e1,e2,error[80:98])
hist(e1,xlim = c(-3,3))
hist(data$p)
plot(e1)
?hist
plot(e1)

library(tseries)
```

```
error1=data.frame(length=rnorm,mean(e1),sd(e1))
error1=data.frame(e1,e2)
hist(error1)
```

Chapter 6

TESTING AND PERFORMANCE EVALUATION

Software testing is the process of evaluation of a software item to detect differences between expected output and actual output. Testing assesses the quality of the product. Software testing is a process that should be done during the development process. In other words software testing is a verification and validation process.

6.1 Validation Testing

The process of evaluating software during the development process or at the end of the development process to determine whether it satisfies specified business requirements. Validation Testing ensures that the product actually meets the client's needs. It can also be defined as to demonstrate that the product fulfils its intended use when deployed on appropriate environment.

It answers to the question, Are we building the right product?

Validation Testing - Workflow:

Validation testing can be best demonstrated using V-Model. The Software/product under test is evaluated during this type of testing.

Testing is performed at different levels involving the complete system or parts of it throughout the life cycle of a software product. A software system goes through four stages of testing before it is actually validated and deployed. These four stages of Validation Testing as:

- Unit Testing
- Integration Testing
- System Testing

- Acceptance testing

The first three levels of testing are performed by a number of different stakeholders in the development organization, whereas acceptance testing is performed by the customers.

6.2 Unit Testing

Unit testing refers to testing program units in isolation. Some examples of commonly understood units are functions, procedures, or methods. Moreover, a program unit is assumed to implement a well-defined function providing a certain level of abstraction to the implementation of higher level functions. Thus, a program unit is tested in isolation, that is, in a stand-alone manner. During unit testing it is desirable to verify that each distinct execution of a program unit produces the expected result.

Unit testing has a limited scope. A programmer will need to verify whether or not a code works correctly by performing unit-level testing. Intuitively, a programmer needs to test a unit as follows:

- Execute every line of code. This is desirable because the programmer needs to know what happens when a line of code is executed. In the absence of such basic observations, surprises at a later stage can be expensive.

- Execute every predicate in the unit to evaluate them to true and false separately.

- Observe that the unit performs its intended function and ensure that it contains no known errors.

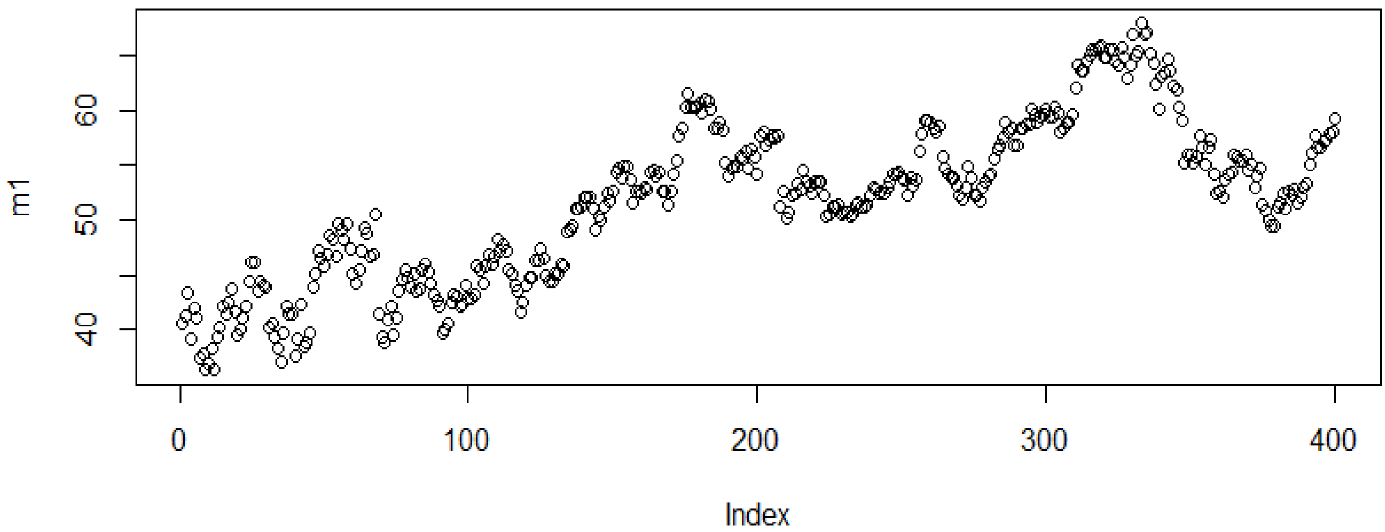

Figure 6.1: Graph of the given dataset

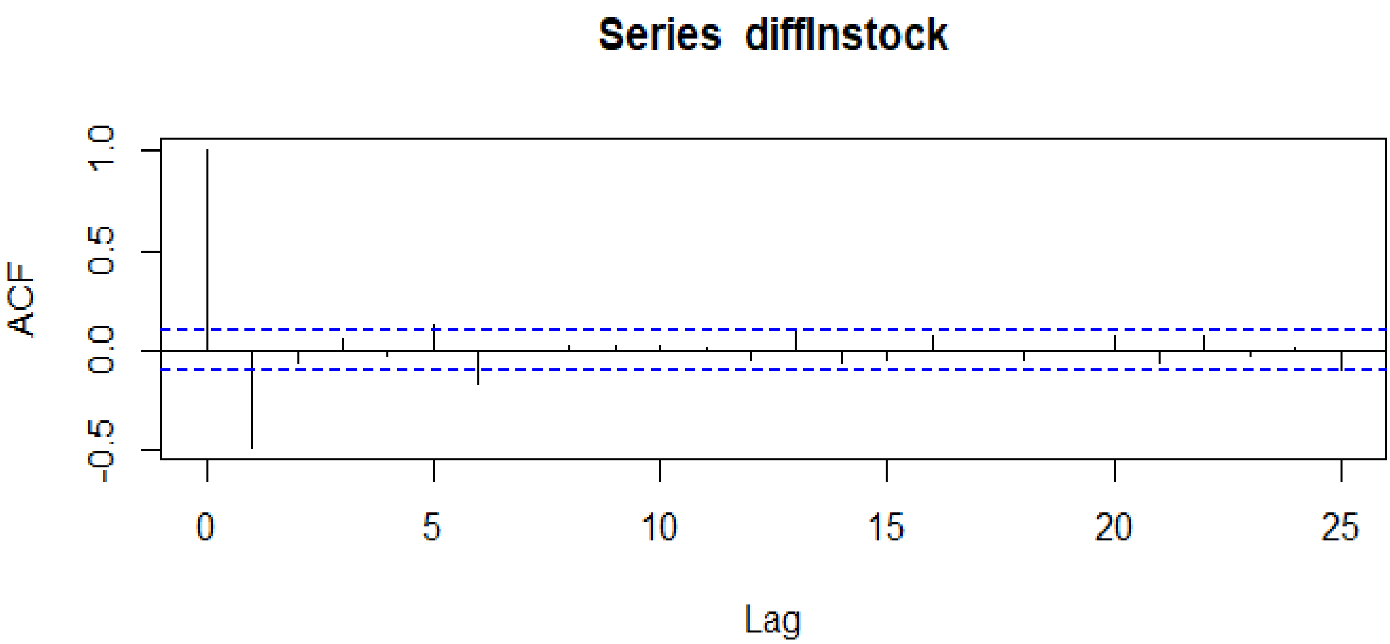

Figure 6.2: Graph of ACF

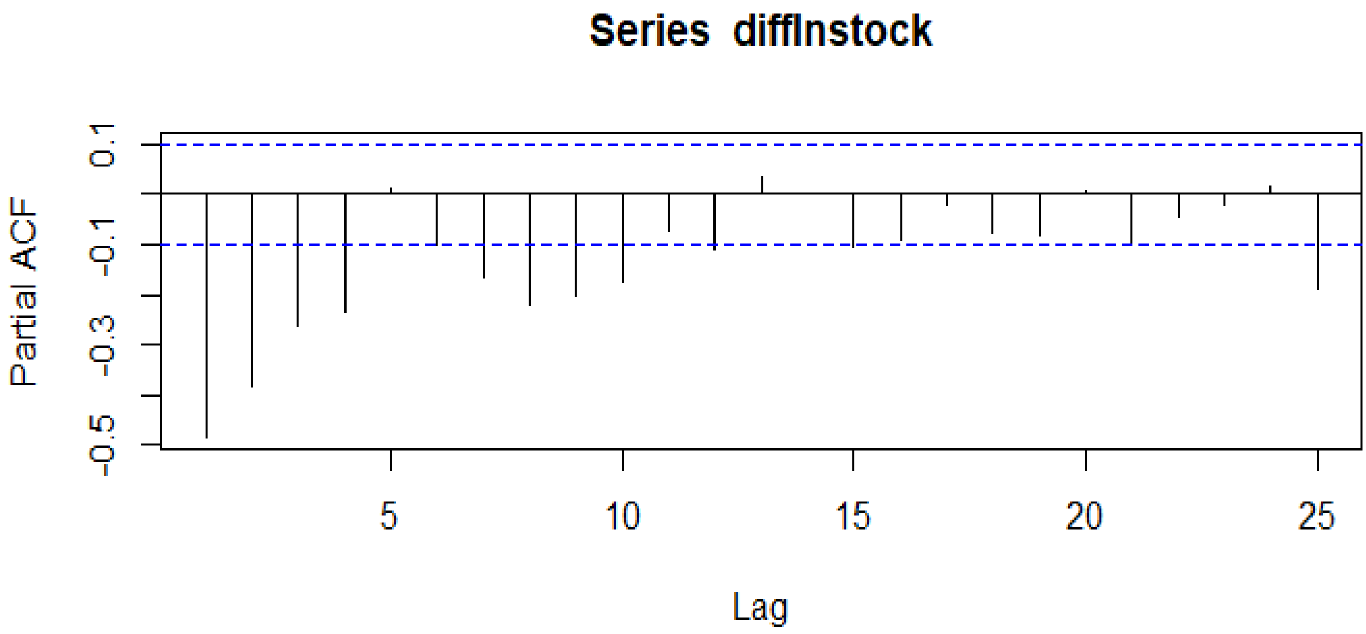

Figure 6.3: Graph of PACF

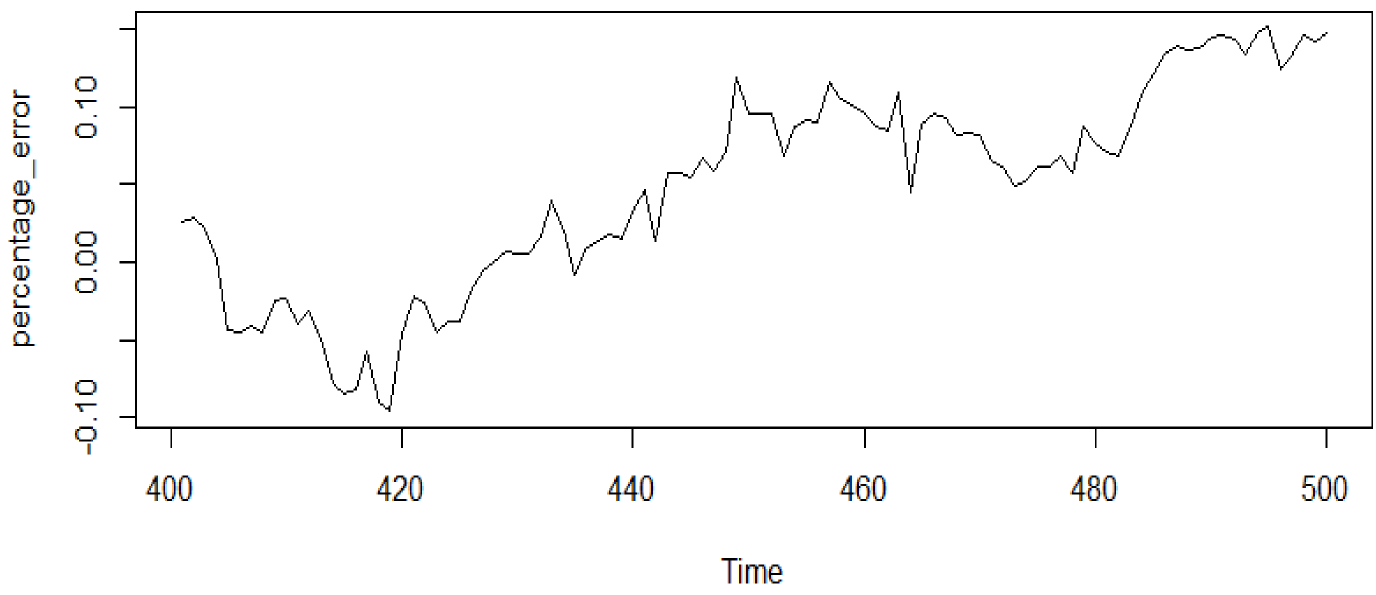

Figure 6.4: Graph of error obtained using ARIMA

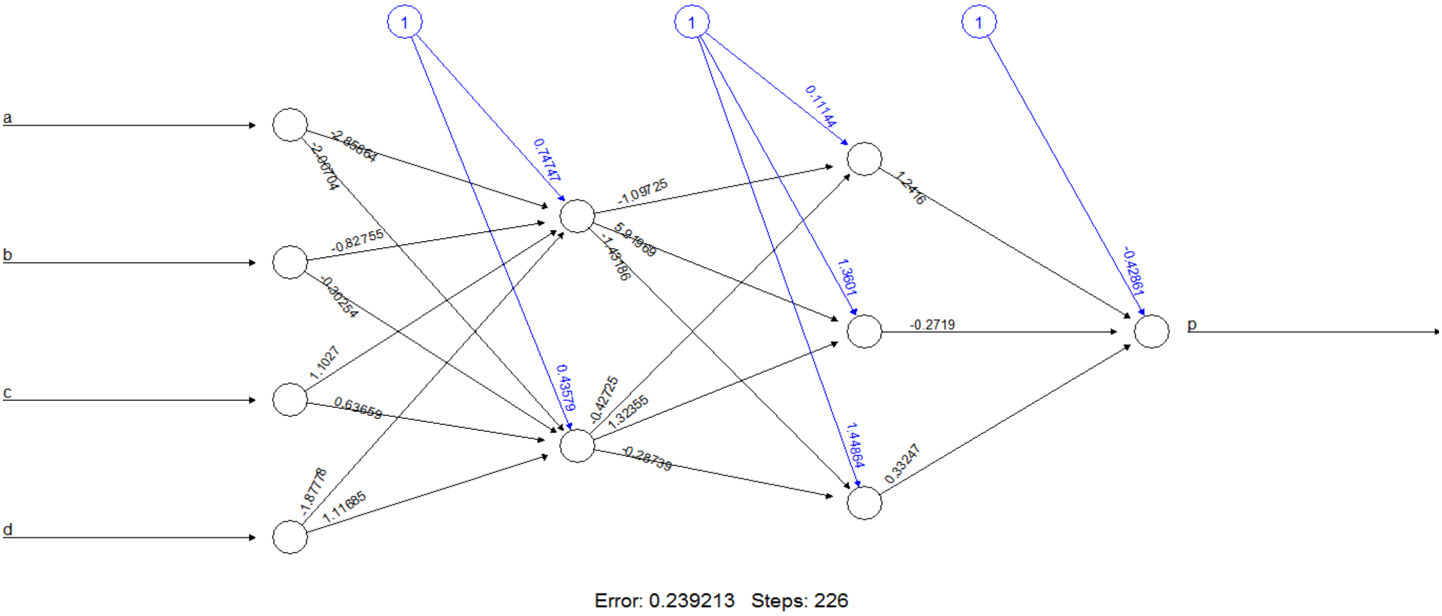

Figure 6.5: Neural Network model 1

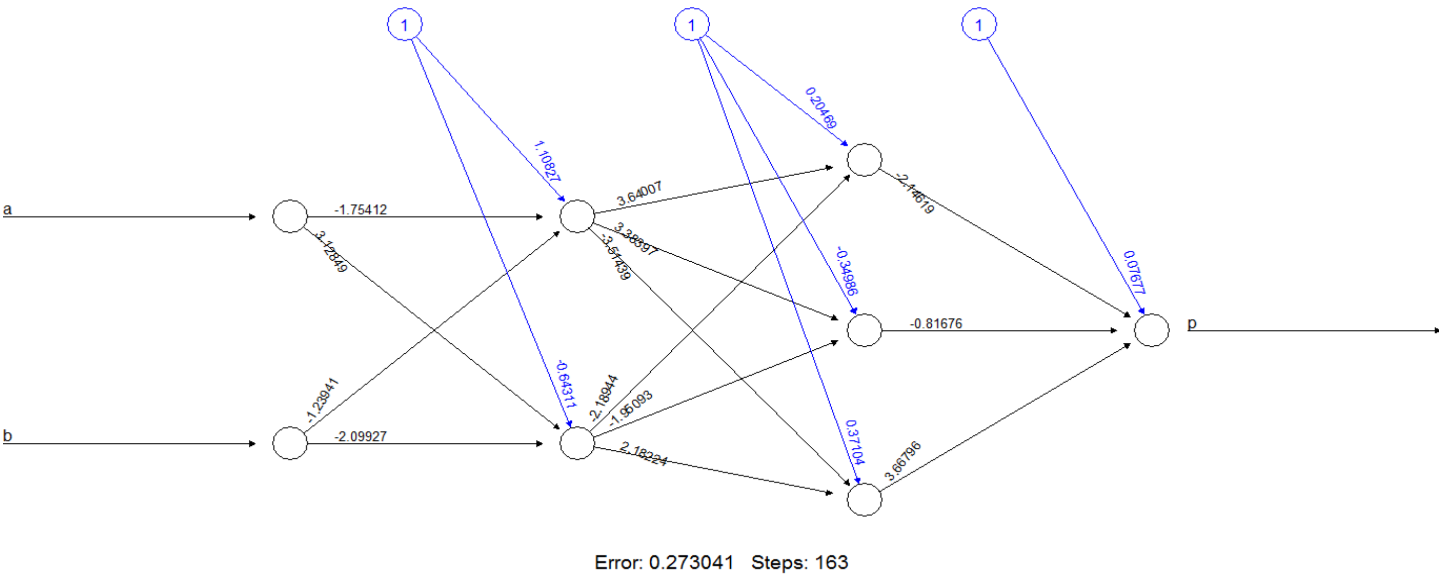

Figure 6.6: Neural Network model 2

error.png

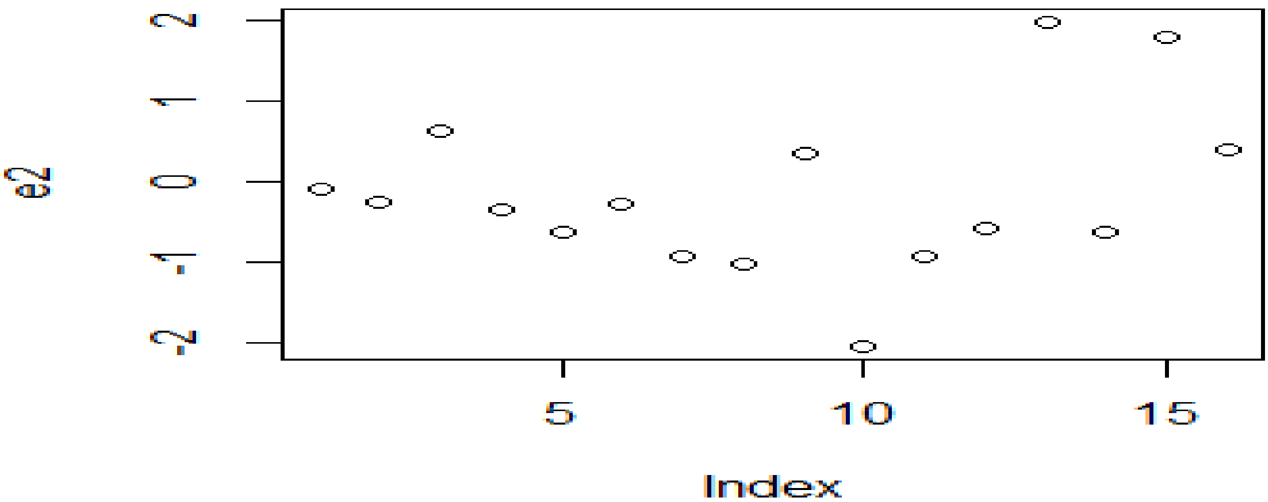

Figure 6.7: Error obtained by neural network

6.3 Integration Testing

Integration testing is a software testing methodology used to test individual software components or units of code to verify interaction between various software components and detect interface defects. Components are tested as a single group or organized in an iterative manner.

6.4 System Testing

SYSTEM TESTING is a level of software testing where a complete and integrated software is tested. The purpose of this test is to evaluate the system's compliance with the specified requirements. system testing: The process of testing an integrated system to verify that it meets specified requirements.

6.5 Acceptance Testing

It is also known as user acceptance testing (UAT), operational acceptance testing (OAT), and end-user testing. It is one of the final stages of the software's testing cycle and often occurs before a client or customer accepts the new application.

Chapter 7

CONCLUSION AND FUTURE SCOPE

7.0.1 Conclusion

This paper presents extensive process of building ARIMA model for stock price prediction. The experimental results obtained with best ARIMA model demonstrated the potential of ARIMA models to predict stock prices satisfactory on short-term basis. This could guide investors in stock market to make profitable investment decisions. With the results obtained ARIMA models can compete reasonably well with emerging forecasting techniques in short-term prediction.

7.0.2 Future directions

The use of the Box-Jenkins Approach or ARIMA to forecast stock prices could be made better by incorporating covariates into the models such as; the introduction of product by an outside company, events that may be occurring in politics, natural disasters, and speculations that are being made about the company in the market. A covariate is a 137 secondary variable that can affect the relationship between the dependent variable and other independent variables of primary interest, and our primary variable is time. By addressing the fact that there are outside components that influence the price per share, we can modify the model to try and anticipate the effect these events will have on the forecasts. In addition to incorporating covariates we could also expand on the topic by looking at more stocks within each industry to determine if the behavior that we observed was the norm, or an exception. Also examine whether or not the behavior in different industries are codependent

Bibliography

[1] Armstrong, J. S. (2001d), Evaluating forecasting methods, in J. S. Armstrong (ed.), Principles of Forecasting. Norwell, MA: Kluwer Academic Press

[2] P. Pai and C. Lin, A hybrid ARIMA and support vector machines model in stock price prediction, Omega vol.33 pp. 497-505, 2005

[3] L.C. Kyungjoo, Y. Sehwan and J. John, Neural Network Model vs. SARIMA Model In Forecasting Korean Stock Price Index (KOSPI), Issues in Information System, vol. 8 no. 2, pp. 372-378, 2007.

[4] N. Merh, V.P. Saxena, and K.R. Pardasani, A Comparison Between Hybrid Approaches of ANN and ARIMA For Indian Stock Trend Forecasting, Journal of Business Intelligence, vol. 3, no.2, pp. 23-43, 2010.

[5] J. Sterba and Hilovska, The Implementation of Hybrid ARIMA Neural Network Prediction Model for Aggregate Water Consumption Prediction, Aplimat-Journal of Applied Mathematics, vol.3, no.3, pp.123-131, 2010.

[6] C. Javier, E. Rosario, J.N. Francisco and J.C. Antonio, ARIMA Models to Predict Next Electricity Price, IEEE Transactions on Power Systems vol. 18 no.3, pp.1014-1020, 2003.

[7] M. Khasel, M. Bijari, and G.A.R Ardali, , Improvement of AutoRegressive Integrated Moving Average models using Fuzzy logic and pp. 956-967, 2009.

[8] C. Lee, C. Ho, Short-term load forecasting using lifting scheme and ARIMA model, Expert System with Applications, vol.38, no.5, pp.5902-5911, 2011.

[9] M. Khashei, M. Bijari, G. A. R. Ardal, Hybridization of autoregressive integrated moving average (ARIMA) with probabilistic neural networks, Computers and Industrial Engineering, vol. 63, no.1, pp.37-45, 2012

[10] C. Wang, A comparison study of between fuzzy time series model and ARIMA model for forecasting Taiwan Export, Expert System with Applications, vol.38, no.8, pp.9296-9304, 2011.

[11] A. Meyler, G. Kenny and T. Quinn, Forecasting Irish Inflation using ARIMA Models, Central Bank of Ireland Research Department, Technical Paper, 3/RT/1998.