Comparison of Return Rate Efficiencies of Forecasting Methods in Stock Market Investment

Um_alkher Saaed Meina

Submitted to the
Institute of Graduate Studies and Research
in partial fulfillment of the requirements for the degree of

Master of Science
in
Computer Engineering

Eastern Mediterranean University
February 2017
Gazimağusa, North Cyprus

Approval of the Institute of Graduate Studies and Research

Prof. Dr. Mustafa Tümer
Director

I certify that this thesis satisfies the requirements as a thesis for the degree of Master of Science in Computer Engineering.

Prof. Dr. Işık Aybay
Chair, Department of Computer Engineering

We certify that we have read this thesis and that in our opinion it is fully adequate in scope and quality as a thesis for the degree of Master of Science in Computer Engineering.

Assoc. Prof. Dr. Mehmet Bodur
Supervisor

Examining Committee

1. Assoc. Prof. Dr. Mehmet Bodur

2. Asst. Prof. Dr. Adnan Acan

3. Asst. Prof. Dr. Ahmet Ünveren

ABSTRACT

Prediction of prices in stock market is an important research topic to direct investments to items with high return rates. This thesis compares available time series prediction methods for predicting of stock market prices. The available methods that have been employed for time series forecasting are support vector regression, autoregressive moving average and k-nearest neighbours. They are applied on four years of stock market data obtained from London Stock Exchange to train each model and to test the performance of the proposed techniques to select the best forecasting method. The result of the tests show that support vector regression gives less forecasting error compared to other methods of forecasting.

Keywords: Stock Market Forecasting, Support Vector Regression, ARMA, k-Nearest Neighbours.

ÖZ

Yatırımları yüksek getiri oranlarına sahip ürünlere yönlendirmek açısından bir borsada fiyatların tahmini, önemli bir araştırma konusudur. Bu tez borsa fiyatlarının tahmini için geliştirilmiş mevcut zaman serileri tahmin yöntemlerinden destek vektör regresyonu, otoregresif hareketli ortalama ve en yakın k komşu yöntemlerini karşılaştırarak en iyi tahmin tekniğini belirlemeyi hedeflemektedir. Her bir model Londra Menkul Kıymetler Borsası'ndan elde edilen dört yıllık borsa verilerinin birinci bölümüyle eğitilmiş ve en iyi tahmin yapabilen yöntemi seçmek için verinin ikinci bölümü önerilen tekniğin performansını test etmek için kullanılmıştır. Testlerin sonucu, SVR yönteminin diğer iki tahmin yöntemine kıyasla tahminde daha az hata verdiğini göstermektedir.

Anahtar Kelimeler: Borsa Tahmini, Destek Vektör Regresyon, ARMA, En Yakın k-Komşu.

To My Family

ACKNOWLEDGMENT

With all the respect and gratitude, I would like to thank my supervisor Assoc. Prof. Dr. Mehmet Bodur for his endless assistance and support in carrying out this thesis.

I would like to extend my thanks to all the staff and instructors of the Department of Computer Engineering at Eastern Mediterranean University who helped me during the study.

I would like to say thanks to all my friends for cooperating with me during my studies.

My special thanks go to my family who has been supporting me at each step of my life.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS

t	Time
R_t	Log-returns of the prices
p_t	Stock price at time *t*
p_{t-1}	Previous value of stock price
x_{k+1}	Next value
σ	Standard deviation
μ	Mean value
γ	Constant parameter of radial basis function
r	Constant parameter of sigmoid function
d	Constant parameter of polynomial function
ε	Control parameter of epsilon insensitive loss function
w	Weight vector
b	Bias term in SVM
$\emptyset(x)$	Nonlinear function
ξ_i	Slack variable
C	Regularization parameter
α_i	Lagrange multipliers
K	Kernel function
θ	Moving average parameter
φ	Autoregressive parameter
ε_t	White noise
q	Order of MA model
P	Order of AR model

R	Real number
m	Embedding dimension of the time series
k	Number of neighbours
SS_R	Regression sum of the square error
SS_T	Total sum square error

LIST OF ABBREVIATIONS

ACF	Autocorrelation Function
AI	Artificial Intelligence
AIC	Akaike Information Criterion
AR	Autoregressive
ARMA	Autoregressive Moving Average
BIC	Bayesian Information Criterion
EMA	Exponential Moving Average
EMH	Efficient Market Hypothesis
FL	Fuzzy Logic
k-NN	k-Nearest Neighbours
LIBSVM	Library of the Support Vector machines
LSE	London Stock Exchange
MA	Moving Average
MATLAB	A software package for mathematical operations
MAE	Mean Absolute Error
NMSE	Normalize Mean Square Error
PACF	Partial Autocorrelation Function
SRM	Structural Risk Minimization
SVM	Support Vector Machine
SVC	Support Vector Classification
SVR	Support Vector Regression

Chapter 1

INTRODUCTION

The importance of time series analysis and forecasting has been emphasized in many areas using positive science, such as engineering, and business. It gathered interest for many researchers in these areas of study.

A time series is an ordered set of data sequence sampled at regular time intervals. Analysis of the time series involves techniques to analyse data in order to gain better understanding of its characteristics and forecast future values based on these characteristics. Time series analysis has been employed in a many organizations such as government organizations in order to forecast the future events. Therefore, many time series forecasting techniques have been proposed in the literature.

Prediction or forecasting of the time series is generally a process that determines the future values using the available information in the data set for making a decision concerning the future. For a time series prediction, understanding the natural structure of the observations is very important.

The primary objective of developing a time series forecasting method is making more accurate prediction of the future, which means reducing the uncertainty inherent in the decision-making process [1]. Successful prediction of the time series depend on many decision processes and proper model fitting in order to guarantee

the accurate prediction of likely outcomes for the future. Prediction of the future values is also important in stock market investment, as the investor would like to make proper decisions to increase their profits. The stock market data set can be treated as a typical time series data and its trend can be analysed accordingly, hence can also be forecasted [2].

The ability to predict the stock market is critical for decision processes in planning, supply management and making the market policy. Therefore prediction of stock market is gaining more attention and has become an important topic where a lot of research efforts have been carried out.

Stock market prediction targets to determine the future value of a company stock or a financial instrument traded on a financial exchange [3]. Prediction of the stock market is gaining more attention due to its financial benefits and its low risk. Numerous investigations gave rise to various decision support systems to provide the investor the optimal prediction of stock. Thus, most of the traders nowadays depend on support trading system which can help them in making right investment decision.

One of the basic theoretical assumptions regarding stock market prediction is the Efficient Market Hypothesis (EMH), which asserts that the price of the stock reflect all information available and everyone has some degree of access to the information. The implication of EMH is that the market reacts instantaneously and no one can outperform the market in the long run. Therefore, a change in daily price is unpredictable, but the trends of prices may be used to predict future value within an uncertainty [4]. In addition, a similar perspective view of the stock market prediction is the random walk hypothesis theory, which believes in an unpredictable price series

and the stock price does not depend on past stock [5]. However the degree of market efficiency is controversial and there are strong evidences to prove that one can beat the market in a short period of time and enable the prediction of price direction based on the current and past data [6].

In order to predict the future stock price, there are two analytical approaches used in the literature to analyse the stock market and make decision. The first method is fundamental analysis and the second method is technical analysis.

Fundamental analysis: It is the approach that investigates the factors which affects supply and demand. The analyst looks at the intrinsic value of the stock, performance of the industry and economy to make a decision whether to invest or not [4]. Fundamental analysts make their decisions by studying sales, profits and earnings or any factors that reflect economy performance.

Technical analysis: It is the approach that involves the evaluation of stock by means of studying statistics generated by market activity, such as past price and volumes [4]. In stock analysis, there are two main approaches; first approach includes analysis of graphs where analysts try to find out certain patterns that are followed by the stock. In second approach analysts make use of quantitative parameters like technical indicators, the daily ups and downs, highest and lowest values of a day, volume of stock, indices, etc., [5]. The analysts make the use of technical indicators as a measurement of the relationship between the current and past stock such as moving average, the rate of the change and exponential moving average.

To accurately predict the stock market, a set of forecasting techniques has been developed that aims to predict the direction and future value of the stock market prices. In the early stage of the stock market prediction, statistical model, especially time series models have been used to construct various prediction models. These models are mostly based on the assumption that the time series is linear and it follows a particular statistical distribution, for example normal distribution [19]. Examples of these models include Exponential Moving Average (EMA), Autoregressive Moving Average (ARMA) and Autoregressive Integrated Moving Average (ARIMA).

However, the financial time series data is understood to be stochastic and also characterized by non-linearity. The data is prone to random fluctuations. Therefore, in recent years, artificial intelligence techniques (AI) have been employed in this field to handle nonlinear relationships that exist between the series, such as an Artificial Neural Network (ANN), Fuzzy Logic (FL) and Support Vector Machine (SVM). These techniques were developed to address the increasing demand for methods that have the ability to learn the patterns of the change which is considered as the main characteristic of stock prices [7].

In this study, both statistical and artificial intelligence techniques have been used. The estimation methods which are used include: autoregressive moving average, support vector regression and k-nearest neighbours. The proposed method is tested in predicting the future price of London Stock Exchange (LSE) and compared to determine the best prediction model depending on the predicted feature values.

1.1 Problem Statement

The stock market prediction is important for making many investment decisions. The factors such as supply and demand of the investment, world events and economic conditions have a greater influence on the stock market investment. All these factors may lead to difficulty in a making an accurate decision about the stock prices as well as when to buy and sell the stock in order to gain more profit. Therefore, it is necessary to develop the model that reflects the structures and patterns of the stock market and enable the forecast movement of the stock prices. The ability to predict the movement of stock market value would be a crucial capability for investors and stakeholders to trade the securities safer and avoid the risk involved when making a decision. The work presented in the study aims to develop three types of predictive models and find out the best prediction model applicable to our data set that can be used to help trading the stock price with the least risk when making an investment decision.

1.2 Research Motivation

Predicting the trend of the financial stock market obviously has a great economical benefit and is considered to be a critical input to many types of planning and investment decision making. Therefore, many strategies were employed that attempt to achieve this, using statistical and artificial intelligence techniques to model stock price. The major motivation for our work includes many benefits including (i) gaining higher outcome from the financial market, (ii) getting a good working prediction model that predicts the trend and future value of the stock market which helps for better trading decision, (iii) increasing the profit of financial communities in stock market which shifts higher amount of investments in successful sectors of stock market.

1.3 Aim and Objective

The main objective of this research is to develop better market trading predictors that can predict the prices and up-down direction of the next trading day of London stock prices accurately, thereby acting as a decision-support tool for the firms, investors and stakeholders. This was accomplished by construction of three different models. Three types of learning algorithms were employed using both artificial intelligence techniques and statistical methods. A comparison between the results of predictions of different techniques was performed in order to find out the best model and method for predicting the direction movement of London stock market prices.

1.4 Significance of the Study

The stock market has been widely studied to extract the useful pattern and predict their future movement. This study would be useful for decision makers, investors and researchers as this would provide the knowledge about the underlying factors behind the forecasting accuracy of the stock market. This will further give more prospective to develop a mechanism for predicting the stock to avoid the risk involved. Predicting the stock market can also provide more benefits to do more business with less risk.

1.5 Thesis Outline

The remaining Chapters of this thesis have the following contents:

Chapter 2, contains a general introduction to the stock market and its data as well as the data pre-processing steps that are used to improve the quality of the time series data. The chapter also describes the methods that are used to conduct our experiments.

Chapter 3, introduces the methodology of the study and our proposed approach to achieve the research objectives. The techniques that are used to build prediction

models are introduced, thus the evaluation criteria and how to select the best method among proposed techniques is explained.

Chapter 4, present the result obtained from our experiments on the data set.

Chapter 5, concludes the research and suggests future research directions.

Chapter 2

DATA SET AND RELATED BACKGROUND

In this chapter, we present the fundamental understanding of subjects related to stock market and its data. In addition, all methods that are used in the study to achieve the research objectives would be described in this chapter. We begin by reviewing the fundamental background studies of the stock market. The next section examines the data related to the stock market. We would describe how it should be pre-processed and transformed before being used for prediction. In the last section we explain the methods that are used in the study.

2.1 Stock Market

The stock market is an extremely productive environment where data is reflected rapidly in prices. It can be characterized as an open market for exchanging the organization's stock and derivative at an approved stock price. These are called securities, recorded on a stock exchange as well as an investor trading secretly [9].

The essential goal of stock market is to serve as a stage for organizations to exchange shares of ownership. Rising share prices will expand business investment and development of the organization's profitability [8].

The stock price of an organization is determined by demand and supply of the stock. In stock market extensive volume of stock is exchanged every day. If the stock is purchased more than it is sold, then the stock price will increase. Alternately, if the

selling of the stock is more than buying, then the stock value will decrease. Besides, the factors such as financial condition, political circumstance and unexpected events are exceedingly causing fluctuations in the stock price. Moreover, stock price fluctuations strongly affect trading volume and investment decision making in the stock market.

Stock markets are organized into stock exchanges which are the place where members of the organization gather to trade company stock [10]. The activity on the stock exchange will induce price movement as it's influenced by supply and demand of investors and stakeholders. Furthermore, stock exchanges play essential role in the country's economic strength and development since they allow the companies to raise their capitals for investment through selling their shares.

2.1.1 Stock Market Data

Various types of stock data are accessible for predicting the stock market. The stock data is related to the circumstance and state of the market. They are ranging from open, close, high, low and volume of the prices.

Fundamentally, the stock data is time series data with past observation that gives a visual illustration of nature fluctuations of the market. The time series data typically characterize by its non-linearity, dynamic and non-stationary because of the random walk process behavior of stock market prices. Consequently, it is generally undesirable to utilize them in their raw form for the forecasting.

2.2 Dataset

The stock market data of the London Stock Exchange (LSE) has been used in this research. The data set was downloaded from the website of www.finance.yahoo.com

[11]. The data set is comprised of the daily closing price of LSE over the period of January 1, 2008 to December 31, 2012, resulting in 1827 trading days.

We partition the whole data into two parts. 70% of the data is for training and 30% for testing. The period utilized as a part of training data beginning from 1st January 2008 to 3rd July 2011. The period utilized as a part of testing data set beginning from 4th July 2011 to 31st December 2012.

2.2.1 Data Pre-processing

The raw data is highly susceptible to noise, missing values and inconsistency. Consequently, a special preparation and transformation is necessary. The raw data is pre-processed so as to enhance efficiency and simplicity of mining process [12].

Missing Data is an important factor in forecasting the future values of stock market items. The stock prices comprise of the vast amount of the data which is incomplete. This can result from missing values in a daily stock data because of weekends and holidays since these are normally not a trading day. Therefore, the data is pre-processed to fill missing values. The estimation of the missing value is done by the mean of interpolation. This method replaces the missing values using the mean value between previous and preceding value of the considered attribute.

Linear interpolation was used to calculate an output value y for the input x using two known values, the previous value (x_k , y_k) and succeeding value (x_{k+1}, y_{k+1}) of the missing data. Its equation is defined as follows:

$$y = y_k + (x - x_k) \frac{(y_{k+1} - y_k)}{(x_{k+1} - x_k)} \quad (2.1)$$

where (x_k , y_k) and (x_{k+1}, y_{k+1}) are the closest points to the unknown values.

Logarithmic Return rates of a stock data provides a stationary time series data. The stock data is converted into stock log-returns because there are more convenient to work with algorithm [12]. Besides, time series data are described by non-stationary, the logarithmic return transformation can convert data to stationary time series data.

Another advantage of log-return is its capability to handle the outliers in the data which decrease its impact on prediction model and increase forecasting accuracy.

This approach utilizes the difference between the natural log of the stock price at time t, and the natural log of the price at the previous step in time, as given by equation 2.2.

$$R_t = \log p_t - \log p_{t-1} = \log \frac{p_t}{p_{t-1}} \quad (2.2)$$

where R_t is log-returns of the data, p_t is current value and p_{t-1} is pervious value.

Data Normalization is necessary to resize the data set into the best working range of the algorithms which operates on the data set. The principal goal of normalization is to improve the quality of the data and increase the performance of the algorithm. In this research, the following formula has been used to normalize the stock prices:

$$Y = \frac{x - \mu(x)}{\sigma(x)} \quad (2.3)$$

where x is vector of data values, Y is vector of normalized data values, $\mu(x)$ is the mean of elements of x, and $\sigma(x)$ is standard deviation of elements of x.

Figure 2.1 shows the daily closing price and log-returns of the London Stock Market after the stock data was pre-processed.

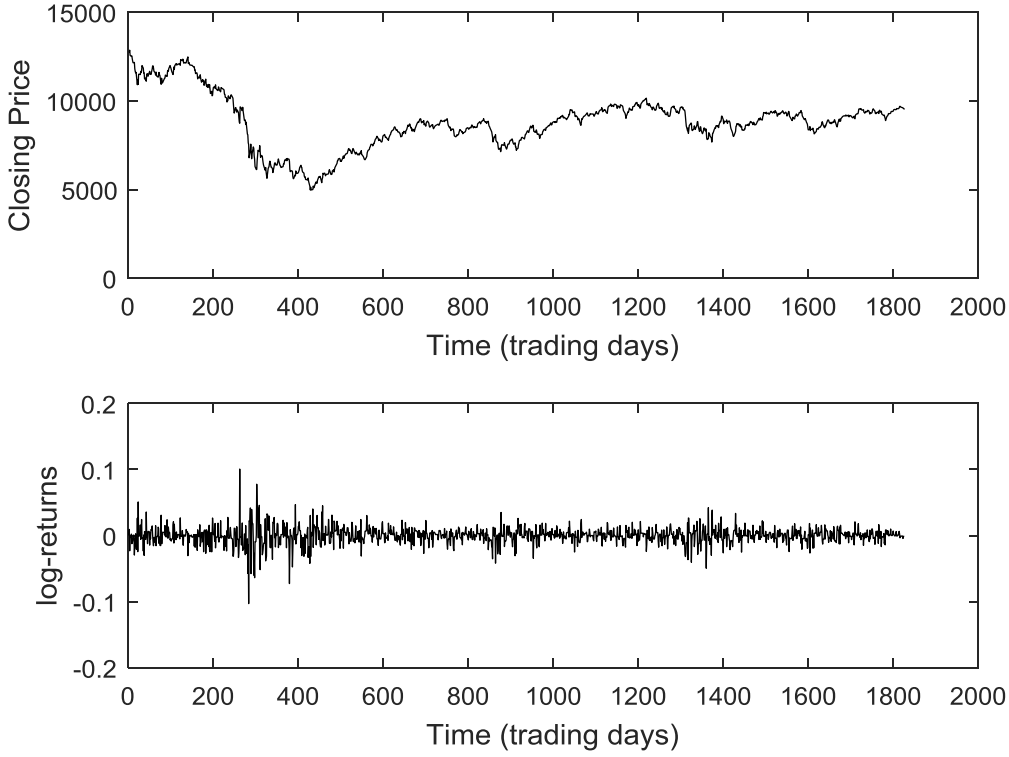

Figure 2.1: Closing Price and Log-Returns of London Stock Market
(top: Closing Prices, bottom: logarithmic return rates)

2.3 Proposed Prediction Techniques

2.3.1 Support Vector Machines

Support Vector Machine (SVM) is a supervised learning technique that can be applied to solve a variety of classification and regression problems. It was proposed by Vapnik and his co-works as implementation of the structural risk minimization principle (SRM) [14].

Basically, the SVM was originally utilized for classifying the input data that are linearly separable. It finds optimal hyper-plane in order to separate the data with maximum margin. Moreover, SVM can be utilized to separate the data that are not linearly separable [13]. In such case, the fundamental thought is to map input data into a high-dimensional feature space by utilizing a nonlinear function, then it separates the data linearly in a higher dimensional space. The separation is done by defining the support vectors with a maximum margin between two separated classes. Figure 2.2 below outlines the thought behind optimal hyper-plane under the assumption of linear separable data.

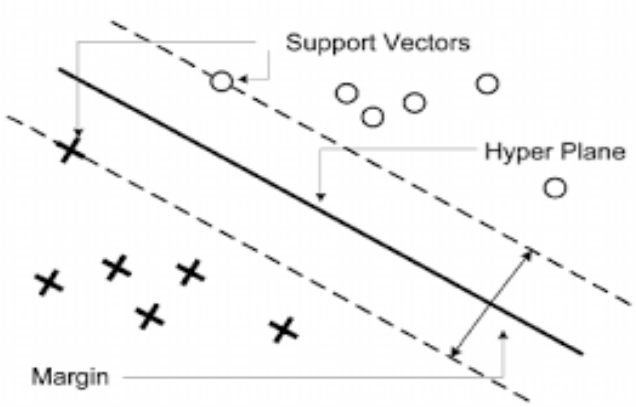

Figure 2.2: SVM in a Separable Case

SVM is specified as a kernel based learning approach. The kernels are functions that play a vital role in the transformation of the data into an appropriate feature space representation in which separating the data is easier. There are many types of the kernel function. However, the most commonly utilized kernels are polynomial, radial basis function and sigmoid which is represented by following equations respectively:

$$K(x_i, x_j) = (1 + (x_i . x_j))^{\mathrm{d}} \qquad (2.4)$$

$$K(x_i, x_j) = \exp(-\gamma \left\| x_i - x_j \right\|^2) \quad (2.5)$$

$$K(x_i, x_j) = \tanh(\gamma(x_i . x_j) + r) \quad (2.6)$$

where γ, r and d are the constant parameters and (x_i, x_j) are vectors in the input space. SVM has become a popular method to learn and analyze data, as a learning algorithm. It has also been shown to yield a good generalization performance in a variety of prediction domains such as financial time series forecasting [10, 13, 15].

Generally, there are two types of the support vector machine which are support vector classification (SVC) and support vector regression (SVR). Its regression form has been applied to solve regression and prediction problems [7] [16]. In this research, we focus our work on SVR.

Support Vector Regression (SVR) is an extension of support vector machine for classification. SVR uses the same principle of the support vector classifier with generalization of SVM for regression estimation. Furthermore, SVR is a nonlinear regression technique that finds the best regression hyper-plane with the lowest risk in a high-dimensional feature space.

In order to perform a regression task, SVR utilizes the loss function to quantify empirical risk and attempt to minimize the regression error [16]. The most commonly utilized function is the ε-insensitive loss function which was proposed by Vapnik.

The ε-insensitive loss function is given by the following equation:

$$l_\varepsilon(y, f(x,w)) = \begin{cases} 0 & if\ |y - f(x,w)| \leq \varepsilon \\ |y - f(x,w)| - \varepsilon & otherwise \end{cases} \quad (2.7)$$

where $\varepsilon > 0$ is a predefined constant parameter that controls the width of ε-insensitive zone, y is actual value and $f(x,w)$ is the predicted value.

Typically, ε-insensitive loss function finds a regression hyper-plane with an ε-insensitive band [16]. It tries to construct a linear hyper-plane in a way that the training data lies within a distance of ε as represented in figure 2.3. Additionally, the ε-insensitive loss function is equal to zero if the data points lie inside the band region. Otherwise, the loss is given by the absolute difference between the actual and predicted values which mean that the data points lie outside the band region.

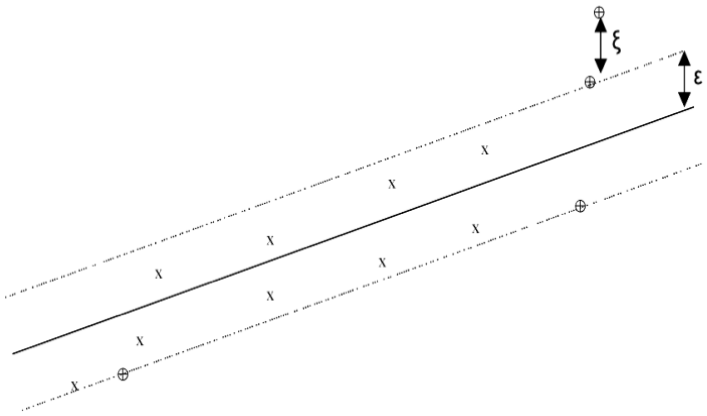

Figure 2.3: SVR using a ε-Insensitive Loss Zone

In SVR model, we are given a set of training data $(x_1, y_1), \ldots, (x_l, y_l)$ where x_i is represented as a set of real input data, y_i is output value and $i = 1, .., l$. The aim is to learn data trend and behavior of the training data vectors, and then use it to predict the target value where the estimated regression function $f(x)$ that used to form a linear regression in a feature space can be expressed as:

$$f(x) = w\,\emptyset(x) + b \tag{2.8}$$

where $\emptyset(x)$ is a nonlinear function that transforms the nonlinear input data into a linear form, w and b are weight vector and constant respectively. In order to accomplish less training error, the slack variables ξ_i, ξ_i^* are introduced in order to measure the deviation of training data that lie outside ε. However, this resulted in the following optimization problem:

$$\text{Minimize } J(w, \xi_i, \xi_i^*) = \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{l}(\xi_i + \xi_i^*) \tag{2.9}$$

$$\text{Subject to, } \begin{cases} y_i - w^T x_i - b \le \varepsilon + \xi_i \\ w^T x_i + b - y_i \le \varepsilon + \xi_i^* \\ \xi_i, \xi_i^* \ge 0 \end{cases} \tag{2.10}$$

After transforming the above optimization problem to dual form the solution is given as follows:

$$F(x) = \sum_{i=1}^{n}(\alpha_i - \alpha_i^*)K(x_i, x) + b \tag{2.11}$$

$$\text{Subject to, } 0 \le \alpha_i \le C, 0 \le \alpha_i^* \le C \tag{2.12}$$

where $C > 0$ is a regularization parameter which controls the trade-off between machine complexity and misclassification errors, α_i and α_i^* are Lagrange multipliers, n is number of support vectors and $K(x_i, x)$ is kernel function that specify as the inner product of two vectors in the feature space [6].

The precision of the support vector machine relies on a suitable determination of the parameter C and kernel parameters. It is critical to decide about the value of these parameters before the experiment. The kernel parameters depend on the kind of kernel function of support vector machine. For example, the radial basis function has the parameter gamma (γ). A good choice of these parameters has a greater impact on the generalization performance of SVM and prediction accuracy [28]. The low estimation of parameter C may increase the misclassification errors where we might have under fitting problem, while a large value of C will prompt to a high penalty for the data points and the over fitting has a tendency to happen. The gamma parameter defines the distance which a single training example can be reached to decision boundary. Thus, it has greater impact on how many samples can be selected by the model as support vectors. The lower value means far training examples and a higher value means closer training examples. In this way, it is imperative to choose these parameters precisely.

One of the commonly used approaches for parameters determination is Cross-Validation [17]. It is an assessment technique that separates the data into two sections, a part for model training and the other for model validation. The fundamental type of cross validation is k-fold cross-validation. In this approach the data is partitioned into k folds of equivalent size, then the cross validation process is repeated k times. At every cycle different partition is utilized for validation and the rest of the folds are utilized for training, and the performance of each fold is measured using mean square error. The average of the errors of all folds is a good estimate of the error for the model trained with the complete training data set.

2.3.2 Autoregressive Moving Average (ARMA)

Autoregressive moving average (ARMA) is a mathematical model based on an integration of AM (p) and MA (q)**.** Mathematically the ARMA (p, q) model is represented as [18]:

$$y_t = c + \varepsilon_t + \sum_{i=1}^{p} \varphi_i y_{t-i} + \sum_{j=1}^{q} \theta_j \varepsilon_{t-j} \quad (2.13)$$

where φ_i and θ_j are coefficients of AR and MA models respectively and c is a constant term which represent the mean of the series.

An AR model is expressed as a combination of one or more previous values and a random error. The AR (p) is defined as follows [2]:

$$y_t = c + \sum_{i=1}^{p} \varphi_i y_{t-i} + \varepsilon_t = c + \varphi_1 y_{t-1} + \varphi_2 y_{t-2} + \ldots\ldots\ldots \varphi_p y_{t-p} + \varepsilon_t \quad (2.14)$$

where y_t and ε_t are respectively the actual value and random error at time period t, c is the mean of the series, ($\varphi_1, \varphi_2, \ldots, \varphi_p$) are model parameters and p is order of the model.

An MA (q) model utilizes past errors as the input variables. The past errors are assumed to follow the typical normal distribution. Thus, a moving average model is a linear regression of the current observations of the time series against the random errors of one or more prior observations. The MA (q) model is given by [2]:

$$y_t = c + \sum_{j=1}^{q} \theta_j \varepsilon_{t-j} + \varepsilon_t = c + \theta_1 \varepsilon_{t-1} + \theta_2 \varepsilon_{t-2} + \ldots\ldots\ldots + \theta_q \varepsilon_{t-q} + \varepsilon_t \quad (2.15)$$

where c is the mean of the series, $(\theta_1, \theta_2, \ldots, \theta_q)$ are the model parameters and q is the order of the model.

An ARMA (*p, q)* is an appropriate technique for modeling stationary procedures where the mean and variance are invariant in a time. However, most of the time series data is non-stationary due to the existence of the trend. The trend is defined as a pattern of deterministic change in a series of the data; it can be eliminated by curve fitting prior to ARMA modelling where a smooth curve describing the growth trend is removed [19]. Elimination is carried out by including smoothing stage while modeling ARMA such as exponential smoothing. This also helps to ensure the stationarity of the time series data. An ARMA model is commonly used to analyze time series in order to gain better understanding and predicting the future value in the series.

An ARMA (*p, q)* is built by a series of well-defined steps in order to select the applicable model that can be used to predict future values of time series.

The Modeling Steps in ARMA (*p, q)* are described as follows:

1- Model Identification

The initial step is to identify the model by selecting the proper order of the model and determine if the series is stationary with the constant mean and variance. There are two strategies for identifying the order of the model; the principal approach is Box-Jenkins technique. This method is based on inspecting the plot of the autocorrelation (ACF) and partial autocorrelation functions (PACF) to choose which AR and MA component ought to be utilized as a part of the model. The second

technique to identify the order of the model is an automated iterative procedure. In this approach different ARMA model is prepared and fit. Then a goodness-of-fit statistic is utilized to choose the best model. The Akaike Information criterion (AIC) and Bayesian information criterion (BIC) are broadly used to measure the goodness-of-fit of statistical models. Both AIC and BIC are utilized for model determination. The best fit model has the minimum of AIC and BIC.

The AIC and BIC defined as follows:

$$\text{AIC} = -2\log(\sigma) + 2k \tag{2.16}$$

$$\text{BIC} = -2\log(\sigma) + k\log(N) \tag{2.17}$$

where σ is a variance of the model residuals, N is the number of the observations and k is the number of estimated parameters.

2- Model Estimation

This step involves estimation of model coefficients. By using computation methods that determine the best fit coefficients of the selected model such as maximum likelihood estimation and least squares estimation.

3- Model Checking

In this step, the selected model is checked to ensure that residual of ARMA model is random, i.e., residuals are independent of each other and have constant mean and variance over the time [19]. The autocorrelation function (ACF) is one of the most normally utilized techniques to test randomness. In order to check the randomness,

the individual residuals of ACF ought to be relatively small, zero or close to zero and generally within $\pm 2/\sqrt{n}$, where n represent number of the observations. This step also ensures that the chosen model fits the data properly or not.

2.3.3 k- Nearest Neighbours

The k-nearest neighbours (k-NN) is a technique for classifying the data points according to nearest training patterns in a feature space. It is a non-parametric approach utilized for classification and regression problems. The k-NN is referred as instance-based learning strategy [20]. The instance-based learning is a supervised learning in which the data set are categorized by comparing it with the already classified data [29].

The k-NN is widely utilized in pattern classification and forecasting. The algorithm predicts the output using a classification approach, according to the result of the majority vote on the most occurrence class within the group of the neighbours.

In k-NN algorithm, all observations are represented as a set of vectors. When the output of a new vector is requested, the algorithm determines the k training data vectors that are nearest to the new vector among the training vector set using a distance metrics. The output for the new vector is predicted from these nearest k vectors by least squares regression, or simply from the average of their future values.

In k-NN forecast, the choice of the size (m) which is known as embedding dimension and the number of neighbours (k) is an essential point of this method; where m and k are predefined parameters by the users. Additionally, performance of the k-NN depends profoundly on these parameters.

Chapter 3

RESEACH METHODOLOGY

This section examines the proposed research methodology. The methods and steps that are utilized to build up the forecast models are clarified. The evaluation methodology that was used for validating the performance of the perdition models is discussed. Then the approach that was used for selecting the best forecast technique is clarified as a last stage of the Research Methodology.

3.1 Problem Formulation

Predicting stock market is described as the procedure of estimation in unknown future events to help decision making. The interest of predicting the stock prices originates from its advantages of having the better knowledge about the future value developments, its financial benefit and avoiding a certain risk in a financial market.

Generally in stock market the financial specialists are often facing the difficulty in deciding the best time for selling or buying their stock to expand their benefit because of the unpredictable behaviour of the stock market. In this manner, having the possibility to predict stock price movements can help decision making procedure in a stock market.

Hence, in this research, we proposed the development of the three types of forecasting models with the purpose of stock market prediction. The forecasting

models include: Support Vector Regression Model, Autoregressive Moving Average Model and k-Nearest Neighbours' Model.

The contribution of the exploration is to identify the best forecasting method regarding the most accurate prediction result.

3.2 Proposed Approach

Three types of techniques have been utilized to construct the forecasting model that examines the stock patterns using the "Support Vector Regression, Autoregressive Moving Average and k-Nearest Neighbours". The proposed research strategy is shown in a figure 3.1. The fundamental stages of our proposed approach are explained in the following sub-sections.

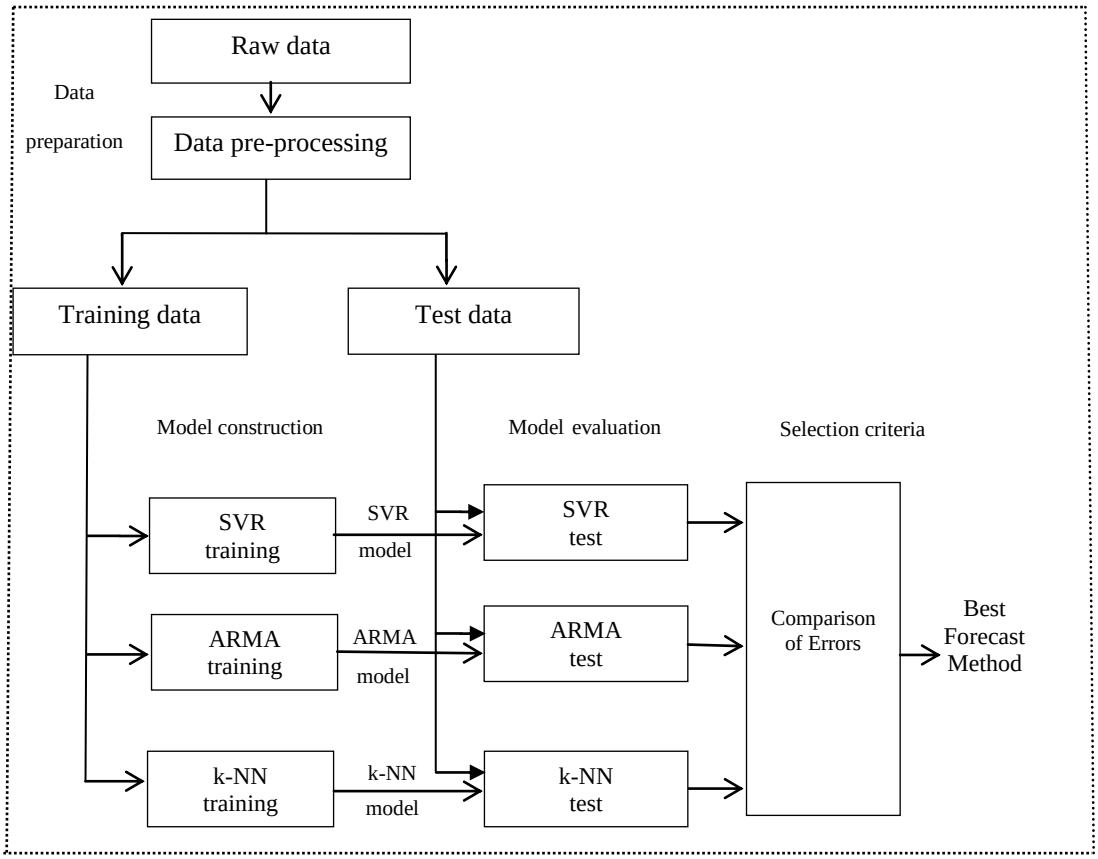

Figure 3.1: Flow Diagram to Decide on the Best Forecasting Method

3.2.1 Data Preparation

In order to predict the future trends and assess our prediction techniques, the historical stock prices of London Stock Exchange has been chosen as the experimental data. Then the raw data have been pre-processed and separated into in-sample and out of sample data set as mentioned in chapter 2, section 2.2 and 2.2.1. The In-sample data is used to construct the forecast models, the out of sample data is used to assess the how well the prediction models preform in a forecasting the new data set.

3.2.2 Model Construction

After preparing the raw data, the following step was the creation of the prediction models using previously mentioned techniques. The proposed prediction models were created to predict the future daily closing price of the LSE. All prediction models were provided with the same input of the daily closing price. 70% of our data were utilized to build prediction models.

The first prediction model was support vector regression model. SVR with ε-insensitive loss function was utilized to build the model. We choose radial basis function as a kernel function because of its capability to map a non-linear training data and its superior performance [1, 10, 16]. The radial basis function is calculated by the following formula:

$$K(x_i, x_j) = \exp(-\gamma \left\| x_i - x_j \right\|^2) \qquad (3.1)$$

where γ is a constant parameter. The precision of the support vector regression model is exceedingly impacted by the estimation of the parameters. Therefore, we utilized a grid-search on parameters C and γ using k-fold cross-validation technique to choose

the proper estimation of the parameters. In order to form the SVR model, the LIBSVM [21], a library of the support vector machines was utilized to conduct the experiment.

The second prediction model was an autoregressive moving average model. An ARMA model was developed according to the steps that have been described in chapter 2, section 2.3.2. As discussed earlier, when conducting ARMA model, the stationarity of the data must be considered for proper prediction. Consequently, the main stage included testing data for stationarity. Thus, we use logarithmic transformation technique to ensure the stationarity of the data and stabilize attribute.

Identifying the order of ARMA (p, q) model is done by utilizing an automated iterative procedure. The AIC and BIC were utilized as a measurement to choose the best fitted model. The next step involved estimation of the parameters for a tentative model with maximum likelihood estimation. After the parameters have been estimated, in the next step we test the residuals of the model with ACF plot to check for the model suitability. If the residuals are random, then the chosen model can be utilized for forecasting the future price.

The last prediction model was k-nearest neighbours' model. Essentially, the algorithm begins with the determination of the ideal number of neighbours (k). The approach that is used to choose the parameter k in this research has been used as a part of [22-24] by testing the algorithm with various values of k and characterized the best value of k that yield the minimum errors.

Different k values has been attempted to determine the optimum value of k that generates the best prediction result. After the parameter k was identified, the distance between the observations of the training data and test data is calculated using Manhattan distance. The Manhattan distance is defined as:

$$D(\vec{x}, \vec{y}) = \sum_{i=1}^{n} |x_i - y_j| \quad (3.2)$$

The distance metric takes an absolute difference between the coordinates, where x_i and y_j are components of vectors $\vec{x}, \vec{y}$ respectively and n is the number of observations. After sorting the distance from the smallest to the highest values, determination of the k closest neighbours is done based on the minimum distance to the test data. The prediction for the following day is processed as the average of identifying neighbours. The average was computed as follows:

$$y = \frac{1}{k} \sum_{i=1}^{k} x_i \quad (3.3)$$

where k is the number of the closest neighbours of x_i and y is the forecast value of the test data.

3.2.3 Model Evaluation

After the forecast models have been built, the next stage is to assess the models with a new data set. This step is critical to figure out if the model has good generalization ability performance or not. Also this step is very important in a determining whether the model is sufficient or insufficient in a use as a model to predict the future stock price. The trained models are tested on 30% of the selected stock prices.

3.2.4 Selection Criteria

After testing the model, we have assessed the accuracy of the models by utilizing assessment measures. The point is to measure the deviation of predicted stock prices from the actual value. Three kinds of measurement have been used to quantify the precision of the models, thus to be specific we have used Normalize Mean Square Error (NMSE), Mean Absolute Error (MAE) and Coefficient of Determination (R^2). Mathematically, these measures are calculated by following equations:

$$\text{MAE} = \frac{1}{n}\sum_{t=1}^{n} |x_t - y_t| \tag{3.4}$$

$$\text{NMSE} = \frac{1}{n}\sum_{t=1}^{n} \frac{(x_t - y_t)^2}{\overline{x_t}\,\overline{y_t}} \tag{3.5}$$

$$R^2 = \frac{SS_R}{SS_T}\ , SS_R = \sum_{t=1}^{n}(y_t - \overline{x_t})^2,\ SS_T = \sum_{t=1}^{n}(x_t - \overline{x_t})^2 \tag{3.6}$$

where x_t is real values, y_t is the predicted values, n is number of the observations, SS_T is the total sum square error and SS_R is the regression sum of the square error. The criteria to judge for the best model are relatively small of NMSE and MAE, and higher R^2 value.

Chapter 4

IMPLEMENTATION AND RESULTS

This chapter outlines the results of the study. As mentioned earlier, historical data of London Stock Exchange was selected to perform the tests. The principal objective is to generate a one-day forecast of the closing price using the prediction models by examining the historical data. The implementation of all steps was conducted in a MATLAB environment.

4.1 SVR Model Results

We considered support vector machine for regression (SVR) with Gaussian kernel to conduct our experiment. The parameters to be determined are kernel parameter Gamma (γ), C and ε. According to [6, 25, 26], it showed that SVR is insensitive to ε, as long as it is a reasonable value. Therefore in this work, we choose 0.01 for ε. To select values for γ and C, we used a 10-fold cross-validation in combination with a grid-search to determine the appropriate values. Various pairs of (C, γ) values are tried and one with the best cross-validation accuracy is selected. According to the results of the cross-validation, it was found that the best performance can be obtained with C= 8 and γ = 0.0625. Figure 4.1 shows the forecasted result of the developed SVR model. The figure demonstrated the performance of model on the test data set and how the predicted and actual stock prices are close to each other. As it is obvious from the figure, the predictions follow the actual data extremely well which indicates that SVR model is sufficient when it is used as a prediction model.

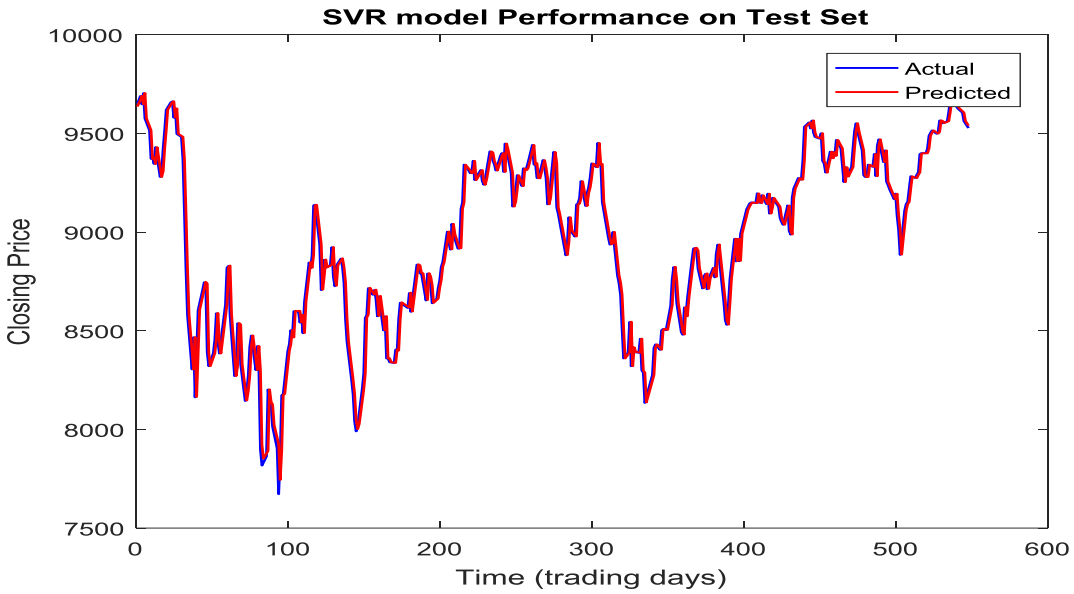

Figure 4.1: Actual vs. Predicted Values of London Stock Market

Figure 4.2 shows prediction errors after applying SVR model on test set which indicates how far away are actual prices from the predicted prices. The prediction errors E_i are calculated for each day by $E_i = x_i - y_i$ where x_i is actual prices and y_i is predicted prices. We obtain for each day the prediction errors that represent a difference between the predicted prices and actual prices. However, the prediction errors are mostly close to each other as it is shown in the figure. Moreover, most of the errors fluctuated around zero indicate that the errors of SVR model is quite acceptable.

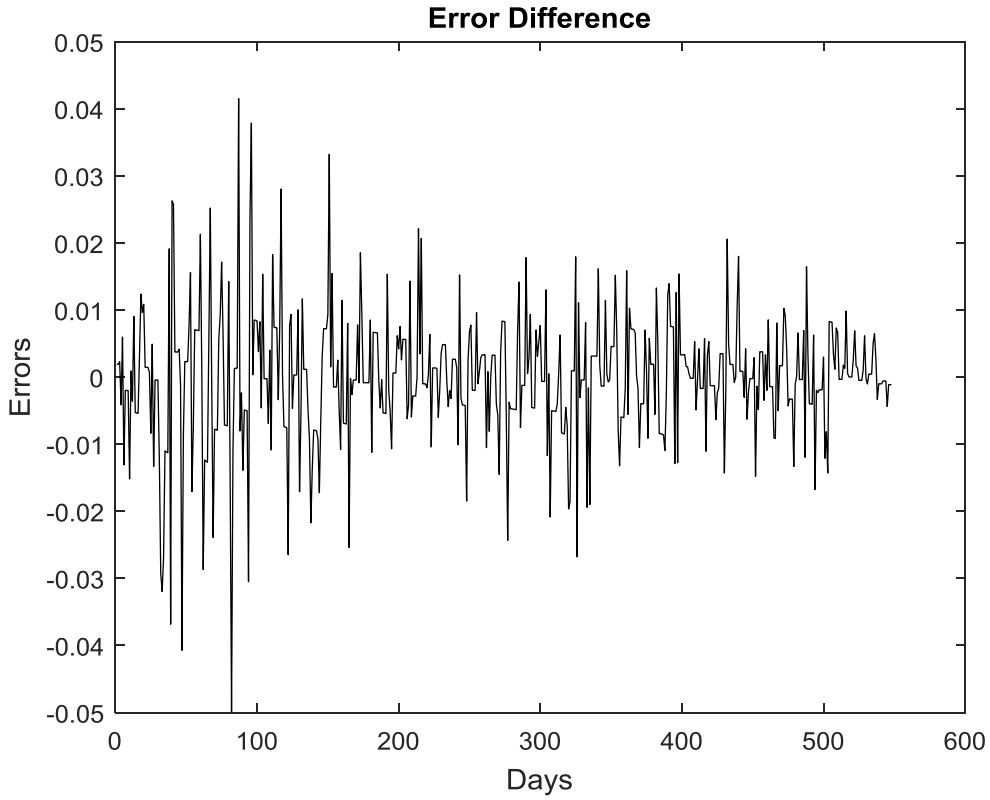

Figure 4.2: Error between Actual and Predicted Stock Prices of SVR Model

In order to evaluate the performance of the model, Normalized Mean Square Error (NMSE), Mean Absolute Error (MAE) and R^2 were calculated. Thus, the performance evaluation results of SVR model is given in the table 4.1.

Table 4.1: Evaluation Accuracy of SVR Model

Model	NMSE	MAE	R^2
SVR	0.033996	0.005495	0.965993

4.2 ARMA Model Results

As discussed earlier in chapter 3, section 3.2.2, the first step in ARMA modeling always begins with exploration of the series in order to have an overall view about stationary or non-stationary of the time series data. Figure 4.3 shows the London Stock Market data between the periods 2008 - 2012. From the graph, the time series

over the period show the trends movement and the variance tend to increase over the time. The random movement in the prices indicates that the data are non-stationary.

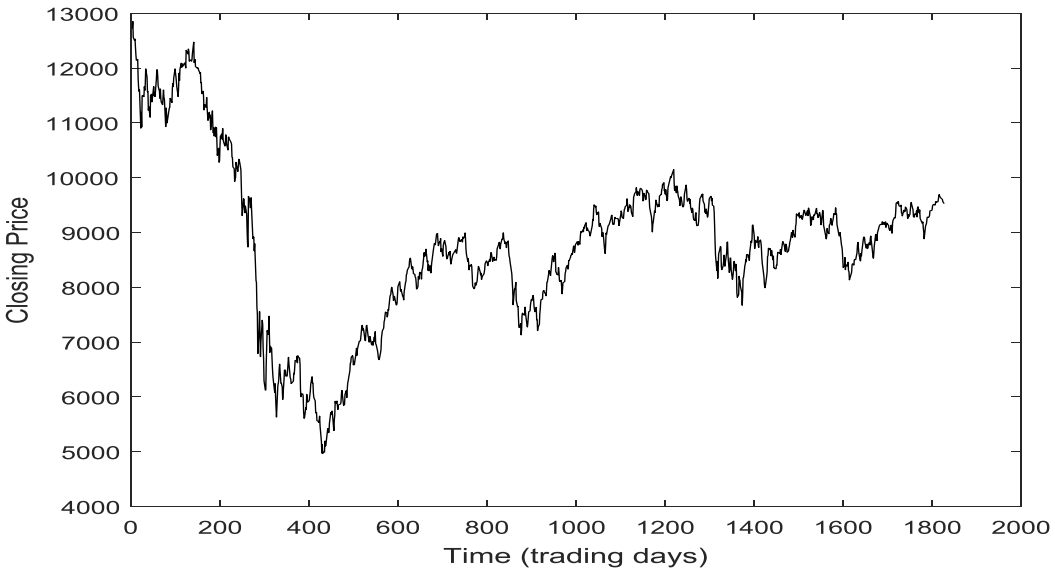

Figure 4.3: The Daily Closing Prices of LSE between the Years 2008-2012

We perform logarithmic return transformation to transform the data to stationary time series data. Figure 4.4 shows the data set after transformation. From the plot it is obvious that the log-returns appear to fluctuate around a constant level where most of the log-returns oscillated around zeros which indicate to the stationarity.

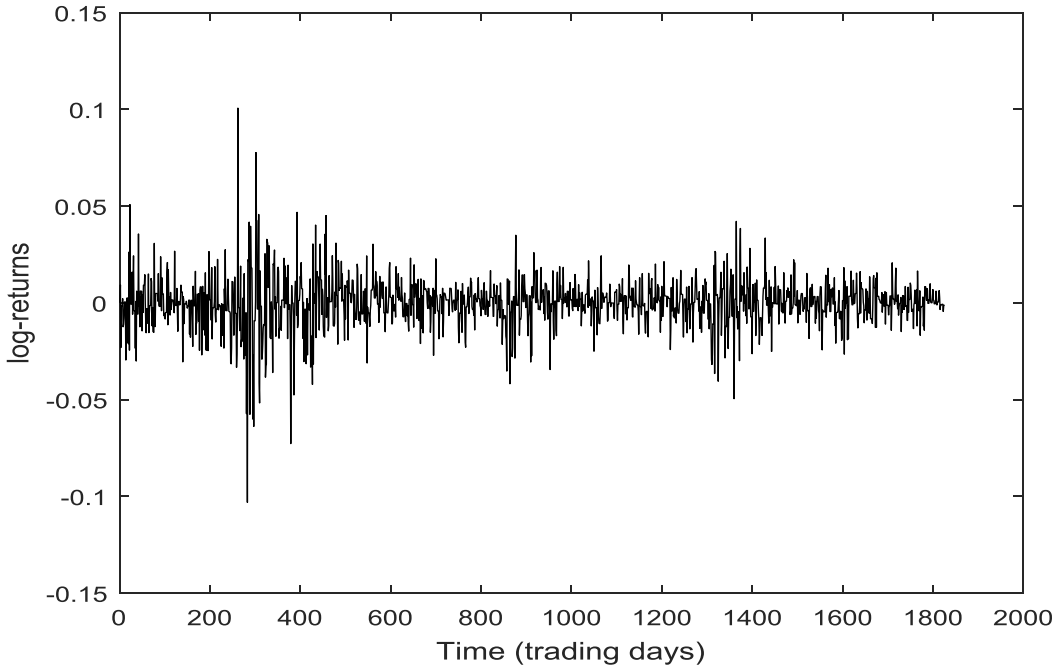

Figure 4.4: Log-Returns of Closing Prices between the Years 2008-2012

After we have obtained the stationary data set, the next step was to identify the best lags p and q of ARMA model based on AIC or BIC criteria. We fit different models with different lags for q and p. In order to perform this experiment the values of q and p were set as follows: p =1 to 10 and q =1 to 10. After that the AIC and BIC criteria was computed for each model to assess the goodness of the fit. The results of AIC and BIC are summarized below:

Lowest AIC = -7511.292262 obtained for p = 9 and q = 9

Lowest BIC = -7469.191524 obtained for p = 1 and q = 2

The selection of the best model is done according to lowest AIC and BIC. ARMA (1,2) is considered the best, since the model return the lowest BIC, hence, was selected as the optimum model.

After selection of the model, the next step involved estimation of the parameters φ_i and θ_j of ARMA(1,2) model where i=1 and j=1, 2. The results of the estimation are summarized below:

AR (1) = 0.995163

MA (1) = 0.164444

MA (2) = 0.091998

In the next step, we checked whether the model fit the data properly by applying a diagnostic checking of model residuals. Figure 4.5 shows the ACF plot of the model errors after applying ARMA(1,2) model on our data set. Note that if the model is good, the residuals should be uncorrelated and ACF of model residuals is expected to be zero or close to zero at the all lags except lag zero. As the plot indicates, most of the lags within acceptable limit indicate sufficiency of the model for forecasting.

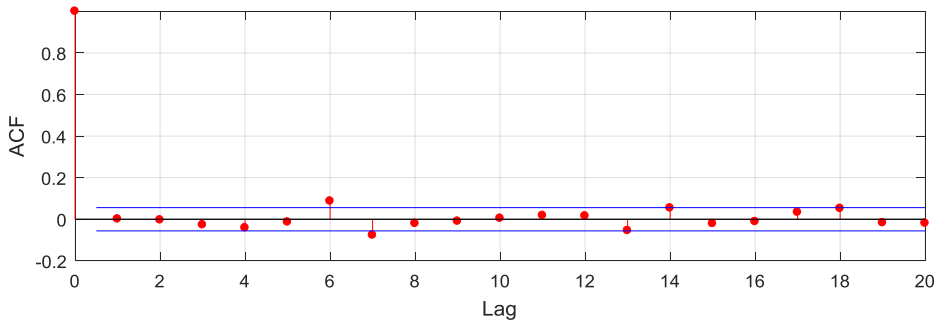

Figure 4.5: ACF of Model Residuals

Figure 4.6 shows the forecasts and actual values of the London stock exchange rates. It is noticeable that the difference between the actual prices and predicted prices is

not distinguishable; since the predicted prices are closely related to the actual prices. The result demonstrates a good performance of the selected model.

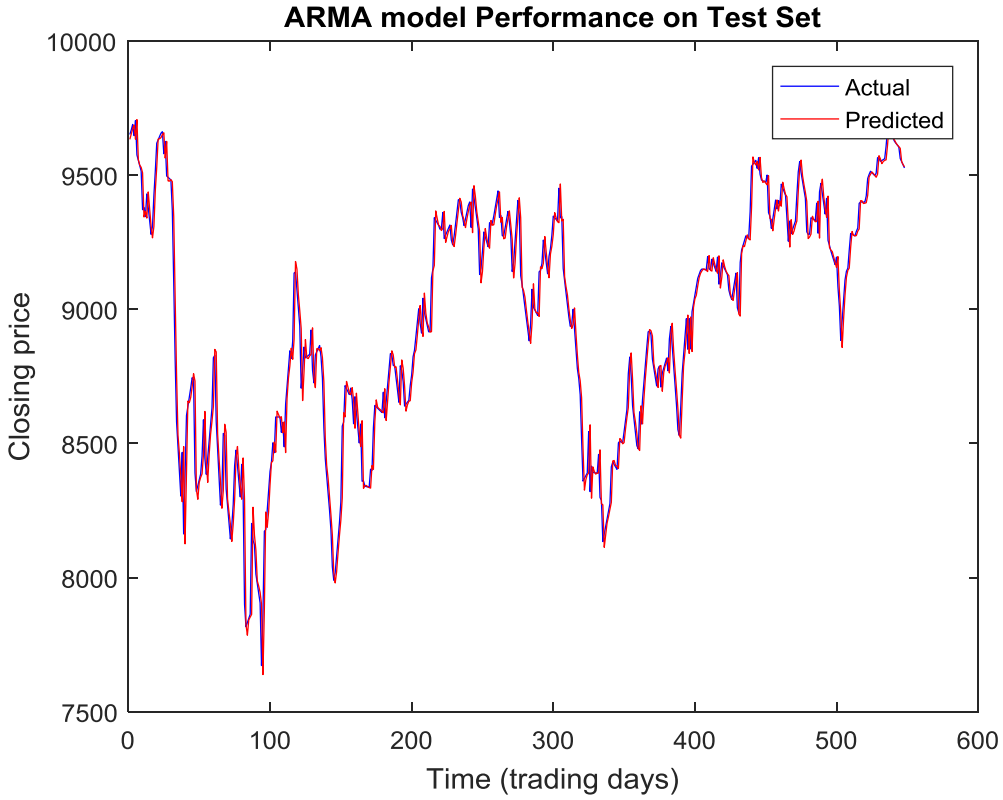

Figure 4.6: Actual vs. Predicted Values of London Stock Market

Figure 4.7 represents the difference between the predicted and actual prices of London Stock Market. From the figure, it is clear that most of the errors are close to zero value. This ensures that the difference between the actual and predicted prices is not very large.

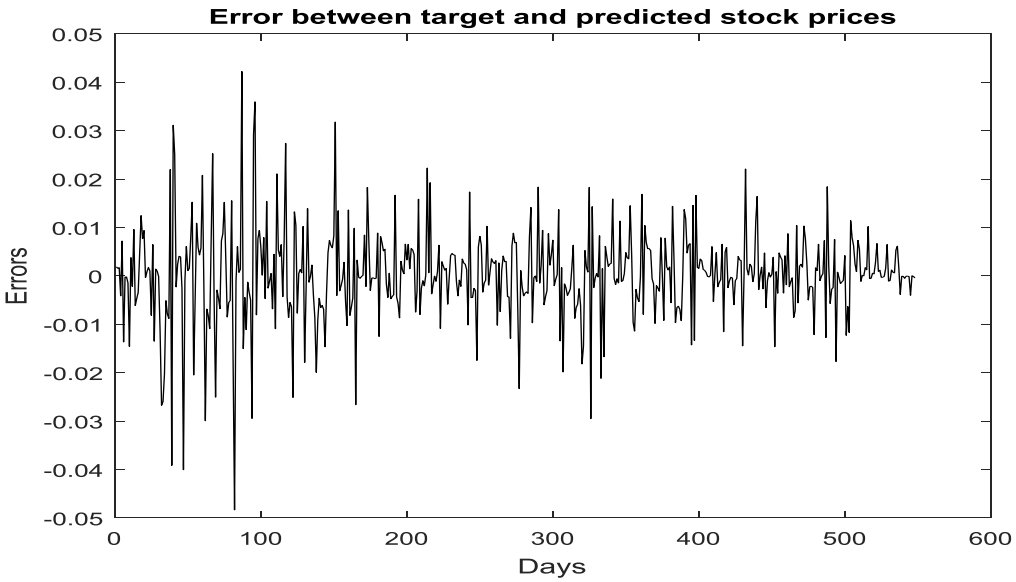

Figure 4.7: Error between Actual and Predicted Stock Prices of ARMA (1,2)

Table 4.2 below represents performance evaluation of ARMA model using evaluation techniques.

Table 4.2: Evaluation Accuracy of ARMA (1, 2) Model

Model	NMSE	MAE	R^2
ARMA(1,2)	0.034745	0.006239	0.965238

4.3 k-NN Model Results

Since our prediction model depend on the number of nearest *k* points in a feature space, we have considered the use of 10 to 50 neighbours with m=2, 3. The selection of the best *k* was done based on the *k* value that yields the smallest mean square error. The best result was obtained with $k = 50$ and m=2 corresponding to smallest mean square error. Figure 4.8 shows the model result for one day a head prediction of London stock market closing prices using Manhattan distance with $k = 50$. It is evident from the graph that the selected parameters are quite acceptable and k-NN performed well as it gives a good performance on the test data set.

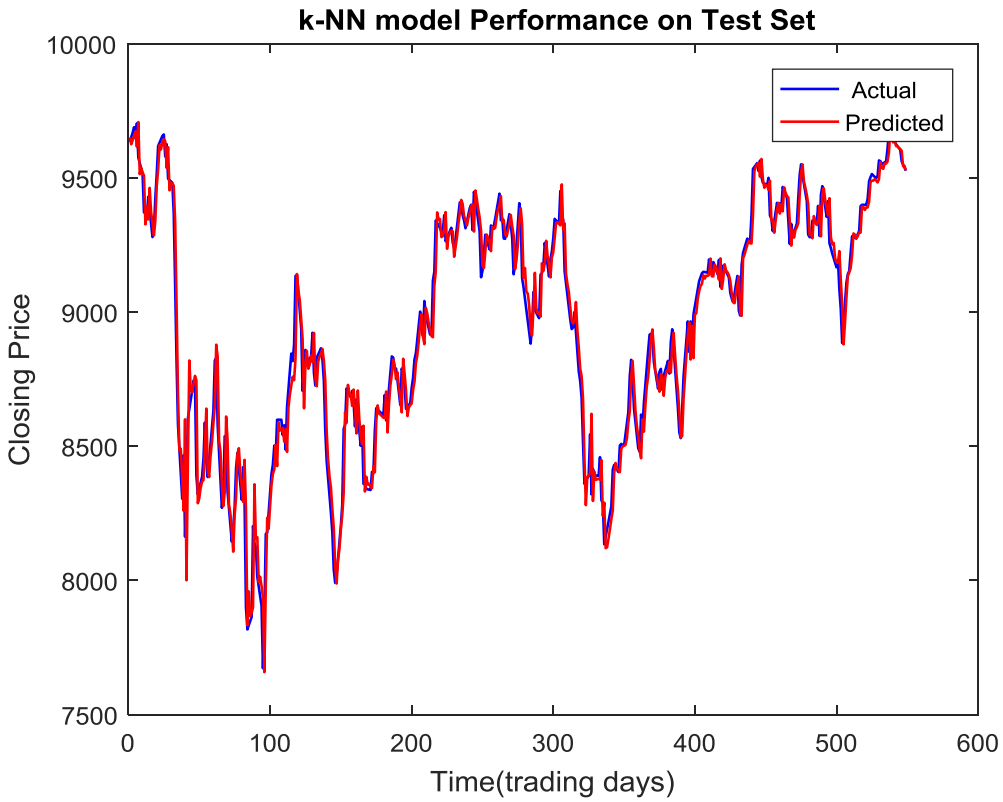

Figure 4.8: Actual vs. Predicted Values of London Stock Market

We estimated the errors between the predicted and actual prices of London Stock Market. As the figure 4.9 shows the errors of k-NN on the *y* axis are ranging from -0.3 to 0.3 values which indicates that k-NN produce the highest errors compared to SVR and ARMA models.

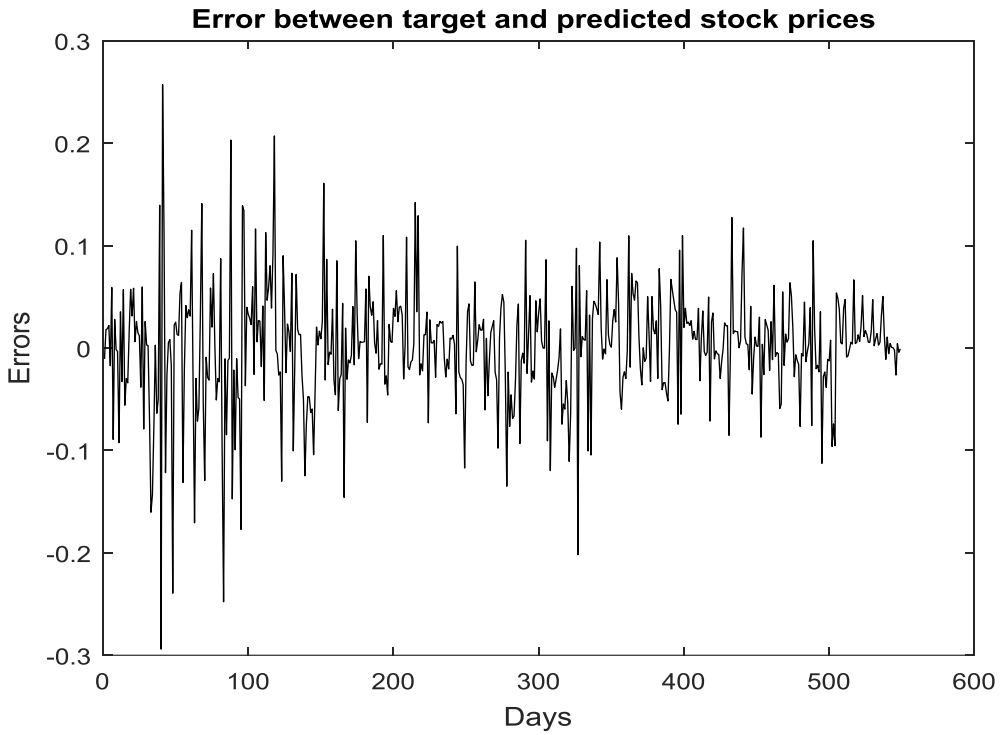

Figure 4.9: Error between Actual and Predicted Stock Prices of k-NN Model

Table 4.3 below represents the performance evaluation of proposed model using evaluation techniques.

Table 4.3: Evaluation Accuracy of k-NN Model

Model	NMSE	MAE	R^2
k-NN	0.036878	0.039978	0.700428

4.4 Comparison of Forecasting Methods

The error figures of the three tested methods are compared in Table 4.4.

Table 4.4: Comparison of Evaluation Accuracies for All Three Methods

Model	NMSE	MAE	R^2
SVR	0.033996	0.005495	0.965993
ARMA(1,2)	0.034745	0.006239	0.965238
k-NN	0.036878	0.039978	0.700428

According to the table of error figures, the method that has the smallest overall error value for both NMSE and MAE scales is the SVR technique. SVR outperforms the other prediction models due to its many advantage compared to ARMA and k-NN methods such as using the various kernels which allows the algorithm to be suits to many prediction problems, also SVM adopt the structural risk minimization principle to minimize the test errors, which eventually leads to better generalization capability of SVM.

However, the figures of all three methods are very close to each other, and therefore advanced methods to combine all three results may be useful in decision making for stock market investments.

The NMSE and MAE are used as estimators of overall deviation between the actual and predicted values, the smaller value of them indicates to better forecast result. The MAE measure the average of the errors in a set of the predictions, the NMSE normalizes the obtained MSE after dividing it by the test variance; it measures the difference between the actual and predicted values.

Overall NMSE value for ARMA is quiet close to value for SVR; however, their MAE values are very different from each other. It means that although the average error of SVR is very low, it gives comparably large errors for closing prices of some days along the test period. The result also indicates that the MAE value of SVR and ARMA models give lower error than NMSE, however, k-NN model produce quiet large value of MAE in comparison to other prediction models.

The R^2, is squared measure of correlation between the model output and actual output. R^2 getting closer to unity correspond to outputs fully correlated, in other words they are explained by a straight line. The value of R^2 rang from 0 to 1, the higher value indicate more useful model and strong correlation between the actual values and predicted values. The R^2 with the small value close to the zero corresponds to weak correlation of the model prediction to the actual values. Both SVR and ARMA methods predicted the actual future price with very high correlation, and their R^2 measures are nearly 0.965, while k-NN shows very low R^2, compared to the other models, indicating that its prediction correlates to actual values quite weak comparing to SVR and ARMA models.

In summary, to improve the prediction technique, ARMA and SVM methods may be combined by some methods such as applied by [23].

Chapter 5

CONCLUSION

Stock market prediction has become one of the most essential tasks for decision making units to determine their trading strategies. In this research, three forecasting techniques were applied to predict the future price of London stock market namely, support vector regression, autoregressive moving average and k-nearest neighbours.

The outcomes we got demonstrated that each of the three methods has ability to predict the future price of the market. In addition, the research result illustrates that past stock price contain data that can be utilized to forecast the future price, if it is predicted within a certain level of accuracy it can provide more information about the future behaviour of the stock market price. This implies that the proposed prediction techniques can be used for forecasting in the market. Furthermore, our results has found that, support vector regression provide the better prediction result than other proposed techniques.

5.1 Recommendation for Future Studies

There are several directions to extend this work for future improvement. In this thesis, only daily closing prices were used as input to the prediction models. The technical indicators such as moving average and exponential moving average may be combined with historical data for better prediction result.

This research has shown that SVM gives better performance in comparison with other prediction techniques. Further research may compare the standard SVM to the least squares SVM, which is an improved version of the SVM algorithm.

REFERENCES

[1] Doego, E., & Cesar, D. (2013). Prediction of Assets Behavior in financial Time Series Using Machine Learning Algorithms. *International Journal of Advanced Research in Artificial Intelligence*, vol.3, No.11, pp.64-52.

[2] Mondal, P., Shit, L., & Goswami, S. (2014). Study of Effectiveness of Time Series Modeling (ARIMA) in Forecasting Stock Prices. *International Journal of Computer Science, Engineering and Applications*, vol.14, No.2, pp.13-29.

[3] Olaniyi, S., Adewole, S., & Jimoh, G.(2011). Stock Trend Prediction Using Regression Analysis Data Mining Approaches. *ARPN Journal of systems and software*, vol.1, No.11, pp.154-157.

[4] Patel, J., Shah, S., Thakar, P., & Kotrcha, K. (2015). Predicting Stock and Stock Price Index Movement using Deterministic Data Preparation and Machine Learning Techniques. Gujarat*: Elsevier*, No.42, pp.259-268.

[5] Argiddi, R., Apte, S., & Kale, B. (2014). AN Analysis on Stock Market Intelligence and Research Approach. Solapur: *International Journal of Application and Innovation in Engineering& Management*, vol.3, No.1, pp. 297-300.

[6] Jensen, M. (1978). Some Anomalous Evidence Regarding Market Efficiency. *Journal of Financial Economics*, vol.6, pp.95-110.

[7] Ivana, M., Milos, B., & Jelena, M. (2014). Stock Market Trend Prediction Using Support Vector Machine. Serbia: Faculty of economics, vol.13, No.3, pp.147-158.

[8] Ou, P., & Wang, H. (2009). Prediction of Stock Market Movement by Ten Data Mining Techniques. China*: Modern Applied Science*, Vol.3, No.12, pp. 28-42.

[9] Preethi, G., & Santhi, B. (2012). Stock market forecasting techniques: A survey. Thanjavur: *Journal of Theoretical Information Technology*, vol.46, No.1, pp. 1817-3195.

[10] Patil, S., Patidar, K., & Jain, M.(2016). Stock market prediction using support vector machines. Prades: *International Journal of Current Trends in Engineering & Technology*, vol.2, No.1, pp.18-25.

[11] Yahoo Finance [Online]. Available: finance.yahoo.com/. 01/05/2016

[12] Nayak, H., Misra, B., & Behera, B. (2014). Impact of data normalization on stock Index forecasting. *International Journal of Computer Information Systems and Industrial Management Applications*, vol.6, pp. 257- 269.

[13] Hua, W., & Shih, J. (2006). Comparison of Support Vector Machines and Back Propagation Neural networks in Forecasting The Six Major Asian Stock Markets. Taiwan: *International Journal of Electronic Finance*,vol.1, pp.49- 67.

[14] Vapnik, V.N. (1999). An overview of statistical learning theory. *IEEE Transactions on Neural networks*, pp.988-999.

[15] Sheta, A., Ahmed, S., & Faris, H. (2015). A Comparison between Regression, Artificial neural Network and Support Vector Machine for Predicting Stock Market Index. *International Journal of Advanced research in Artificial intelligence*, vol.4, No. 7, pp.55-63.

[16] Yeh, C., Huang, C., & Lee, S.(2011). Multiple-kernel Support Vector Regression Approach for Stock market Price Forecasting. Taiwan: *Elsevier*, pp. 2177–2186.

[17] Hsu, C., Chang, C., & Chang, C. (2016). A Practical Guide to Support Vector Classification. Taiwan: National Taiwan University.

[18] Cortez, P., Rocha, M. (2010). Evolving Time Series Forecasting ARMA Models. Netherlands*: Kluwer Academic Publishers*, pp. 2-25.

[19] Cochrane, J. (1997). Time Series for Macroeconomics and Finance. Woodlawn: University of Chicago.

[20] Subha, M., Nambi, S. (2012). Classification of Stock Index movement using k-Nearest Neighbours (k-NN) algorithm. *WSEAS transaction on information Science and application*, vol.9, pp. 2224-3402.

[21] Chang, C., & Lin, C. (2011). LIBSVM: a library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, pp. 1-27. Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm.

[22] Mizrach, B. (1992). Multivariate Nearest-Neighbours Forecasts of EMS Exchange Rates. *Journal of Applied Econometrics*, pp.151-163.

[23] Andrada, J., Rodriguez, F., Rivero, S., & Artiles, M. (2001). An empirical evaluation of non-linear trading. *FEDEA*, pp.1-20.

[24] Rodriguez, F., Rivero, S., & Artiles, M. (1997).Using Nearest Neighbours predictors to forecast the Spanish stock market, vol.1, pp.95-91.

[25] Thomason, M. (1999). The practitioner method and tools: a basic neural network-Based trading system project revisited (parts 1 and 2), *Journal Computational Intelligence in Finance*, vol.7, No.3, pp.36–45.

[26] Thomason, M. (1999). The practitioner method and tools: a basic neural network-Based trading system project revisited (parts 3 and 4), *Journal Computational Intelligence in Finance*, vol.7, No.4, pp.35–48.

[27] Ibrahim, A. (2014). Model Based Multi Criteria Decision Making Methods for Prediction of Time Series Data (Master's thesis). North Cyprus: Eastern Mediterranean University.

[28] Cherkassky, V., & Ma, M. (2004). Practical Selection of SVM Parameters and noise estimation for SVM regression. *Elsevier*, No.17, pp. 113–126.

[29] Sharma, S., & Sharma, V. (2012). Time Series prediction Using KNN algorithm via Euclidean distance function: A case for Foreign Exchange Rate prediction. India: *Asian journal of computer science and technology*, vol.2, pp. 219- 221.

APPENDICES

Appendix A: Table of Prediction Results of Forecasting Models

Actual values		Predicted Values		
Sample period	Test data set	SVR	ARMA	k-NN
04/07/2011	9651.48	9173.1	9173.9	9650.7
05/07/2011	9668.31	9174.7	9175.4	9625.1
06/07/2011	9689.01	9176.4	9177.3	9640.2
07/07/2011	9647.26	9178.6	9178.5	9656.3
08/07/2011	9703.54	9174.3	9173.5	9673.2
09/07/2011	9575.37	9180.1	9806.8	9615.7
10/07/2011	9555.31	9164.7	9165.1	9708.3
11/07/2011	9535.2	9162.7	9163.1	9513.7
12/07/2011	9515.18	9160.6	9162.1	9537.7
13/07/2011	9371.03	9164.7	9169.7	9520.4
14/07/2011	9379.363	9145.4	9142.5	9508.5
15/07/2011	9345.26	9146.2	9144.8	9326.9
16/07/2011	9429.97	9142.6	9142.1	9394.1
17/07/2011	9379.1	9151.1	9152.4	9345.2
18/07/2011	9329.54	9160.6	9145.6	9462.9
19/07/2011	9279.26	9171.3	9138.9	9373.9
20/07/2011	9309.12	9172.6	9134.1	9330.1
21/07/2011	9425.20	9173.9	9138.7	9285.2
22/07/2011	9515.94	9175.2	9153.5	9340.1
23/07/2011	9618.85	9175.8	9162.1	9469.6
24/07/2011	9631.39	9167.3	9173.1	9532.1
25/07/2011	9643.92	9172.1	9173.2	9622.3
26/07/2011	9656.41	9158.6	9173.7	9605.8
27/07/2011	9661.83	9158.1	9174.9	9634.3
28/07/2011	9579.64	9157.6	9175.5	9650.7
29/07/2011	9625.587	9157.1	9165.7	9625.1
30/07/2011	9496.83	9145.1	9172.2	9640.2
31/07/2011	9491.9	9115.9	9156.8	9656.3
01/08/2011	9487.02	9084.1	9156.7	9673.2
02/08/2011	9482.12	9057.4	9157.3	9615.7
03/08/2011	9368.5	9046.7	9156.5	9708.3
04/08/2011	9098.7	9035.8	9142.6	9513.7
05/08/2011	8812.8	9024.8	9109.9	9537.7
06/08/2011	9651.48	9044.2	9077.4	9520.4
07/08/2011	9668.37	9007.7	9051.5	9508.5
08/08/2011	9689.005	9034.2	9043.4	9326.9
09/08/2011	9647.2	9060.2	9033.3	9394.1
10/08/2011	9703.57	9064.2	9021.9	9345.2
11/08/2011	9575.35	9068.2	9046.5	9462.9
12/08/2011	9555.31	9072.1	9002.8	9373.9
13/08/2011	9535.25	9076.5	9034.9	9330.1
14/08/2011	9515.18	9075.5	9066.2	9285.2
15/08/2011	9371.02	9035.1	9065.5	9644.3
16/08/2011	8579.89	9029.4	9067.8	9636.7

17/08/2011	8487.94	9032.3	9072.3	9537.1
18/08/2011	8395.9	9034.6	9078.5	9614.5
19/08/2011	8304.03	9042.6	9074.8	9453.6
20/08/2011	8466.82	9058.4	9027.5	9482.8
21/08/2011	8162.73	9041.6	9023.1	9478.8
22/08/2011	8382.83	9034.4	9030.6	9471.3
23/08/2011	8604.03	9173.1	9032.8	9336.9
24/08/2011	8638.36	9174.7	9034.4	9025.9
25/08/2011	8672.7	9176.4	9042.9	8694.4
26/08/2011	8707.03	9178.6	9061.8	8483.9
27/08/2011	8745.2	9174.3	9039.3	8490.8
28/08/2011	8736.83	9180.1	9030.6	8378.8
29/08/2011	8389.99	9164.7	9042.8	8259.5
30/08/2011	8320.89	9162.7	9051.7	8600.1
31/08/2011	8342.54	9160.6	9057.4	8000.4
01/09/2011	8364.19	9164.7	9063.9	8438.3
02/09/2011	8385.85	9145.4	9088.2	8819.5
03/09/2011	8453.21	9146.2	9087.2	8712.1
04/09/2011	8588.13	9142.6	9052.9	8699.5
05/09/2011	8444.74	9151.1	9041.1	8733.1
06/09/2011	8384.82	9160.6	9031.3	8762.3
07/09/2011	8446.54	9171.3	9019.9	8746.1
08/09/2011	8508.66	9172.6	9027.7	8287.4
09/09/2011	8569.99	9173.9	9056.2	8305.1
10/09/2011	8631.71	9175.2	9052.8	8344.6
11/09/2011	8819.48	9175.8	9023.2	8367.6
12/09/2011	8828.55	9167.3	9017.2	8373.5
13/09/2011	8580.49	9172.1	9011.8	8492.9
14/09/2011	8477.04	9158.6	9003.9	8640.3
15/09/2011	8373.69	9158.1	9011.9	8415.7
16/09/2011	8270.24	9157.6	9022.9	8384.9
17/09/2011	8322.56	9157.1	9041.3	8464.4
18/09/2011	8537.47	9145.1	9046.5	8514.5
19/09/2011	8530.21	9115.9	9036.6	8585.5
20/09/2011	8330.85	9084.1	9029.1	8648.5
21/09/2011	8422.02	9057.4	9023.1	8878.4
22/09/2011	8301.2	9046.7	9041.4	8834.4
23/09/2011	7899.63	9035.8	9022.9	8521.2
24/09/2011	7817.05	9024.8	8965.2	8489.1
25/09/2011	7832.53	9044.2	8969.1	8357.6
26/09/2011	7848.108	9007.7	8967.3	8276.8
27/09/2011	7863.63	9034.2	8968.6	8327.8
28/09/2011	8201.99	9060.2	8969.9	8610.7
29/09/2011	8138.93	9064.2	9019.4	8523.4
30/09/2011	9336.78	9068.2	9006.8	8282.1

Appendix B: MATLAB Code

```
%:::::::::support vector machine:::::::%
% read the financial data of stock price
DataSet= xlsread('LSEdataset.xlsx');
[m,n] = size(DataSet);
%traget / input
t = DataSet(2:m,:);
X=DataSet(1:m-1,:);
% Split data into training and test sets:
N = length(t);
split = 0.70;
Train = round(split * N);
nTest = N - Train;
tTrain = t(1:Train);
tTest  = t(Train+1:N);
XTrain = X(1:Train,:);
xTest  = X(Train+1:N,:);
% Normalization :
mu_xTr = mean(XTrain); sig_xTr = std(XTrain);
mu_tTr = mean(tTrain); sig_tTr = std(tTrain);
XTrain = (XTrain - repmat(mu_xTr,Train,1)) ./ ...
   repmat(sig_xTr,Train,1);
tTrain = (tTrain - repmat(mu_tTr,Train,1)) ./ ...
   repmat(sig_tTr,Train,1);
mu_xTe = mean(xTest); sig_xTe = std(xTest);
mu_tTe = mean(tTest); sig_tTe = std(tTest);
xTest = (xTest - repmat(mu_xTe,nTest,1)) ./ repmat(sig_xTe,nTest,1);
tTest = (tTest - repmat(mu_tTe,nTest,1)) ./ repmat(sig_tTe,nTest,1);
% Cross validation:
method       = 3;   % SVM type: 3 = epsilon-SVR
kernel       = 2;   % kernel type:  2 = rbf.
nFoldCV      = 10;  %no. of folds in cross-validation
pars.epsilon = 0.01; % epsilon parameter in epsilon-SVR
display    = true;   % true = display result . false = don't.
%find optimal parameters C and gamma:
if nFoldCV ~= false
log2c_list = -5:2:8;
log2g_list = -8:1:4;
numLog2c = length(log2c_list);
numLog2g = length(log2g_list);
cvMatrix = zeros(numLog2c,numLog2g);
bestcv = 10^9;  % initialize best CV MSE
for i = 1:numLog2c
log2c = log2c_list(i);
for j = 1:numLog2g
log2g = log2g_list(j);
if method == 3  % epsilon-SVR.
svm_params = ['-q -s ',num2str(method), ' -t ',num2str(kernel), ...
   ' -c ',num2str(2^log2c), ' -g ',num2str(2^log2g), ' -p ' ...
   ,num2str(pars.epsilon), ...
' -v ', num2str(nFoldCV)];
 end
 cv = svmtrain(tTrain, XTrain, svm_params); % compute cv MSE
 cvMatrix(i,j) = cv;
 if cv <= bestcv
 bestcv = cv; bestLog2c = log2c; bestLog2g = log2g;
 end
```

```
 end
 end
 bestC = 2^bestLog2c;
 bestg = 2^bestLog2g;
% Print  cross validation results:
if display == true
fprintf('\n Cross Validation Results:\n');
fprintf(['Best parameters:\n  log2C  = %3.1f \t  C = %4.4f\n' ...
'log2gamma = %3.1f  \t  g = %4.4f\n'], ...
bestLog2c, 2^bestLog2c, bestLog2g, 2^bestLog2g);
fprintf('CV MSE = %g \n\n', bestcv);
end
% Train SVM model using optimal  parameters:
% or train model by using specified C and gamma
if method == 3
    svm_params = ['-q -s ', num2str(method),  ' -t ',
   num2str(kernel),...
        ' -c ', num2str(bestC), ' -g ', num2str(bestg),...
        ' -p ', num2str(pars.epsilon), '-b 1'];
end
%Do training by using svmtrain of libsvm
model = svmtrain(tTrain, XTrain, svm_params);
%Do predicting by using svmpredict of libsvm
if display == true
[yTrain, perfTrain, probEstTrain] = svmpredict(tTrain, XTrain,
   model);
else
yTrain = svmpredict(tTrain, XTrain, model);
end
if display == true
fprintf('Prediction on TEST data:\n')
[yTest, perfTest, probEstTest] = svmpredict(tTest, xTest, model);
else
yTest = svmpredict(tTest, xTest, model);
end
disp(yTest);
% De-normalize:  using POSTSTD function to convert the data
% back into unnormalized units.
tTrain=poststd(tTrain,mu_tTr,sig_tTr);
yTrain=poststd(yTrain,mu_tTr,sig_tTr);
tTest=poststd(tTest,mu_tTe,sig_tTe);
yTest=poststd(yTest,mu_tTe,sig_tTe);
%------- Computing  the performance measures of model -----------%
%          Display the result of SVM Regression          %
%Error rate
error_test=tTest-yTest;
%mean absolute error
mAETe=mae(error_test);
%NMSE
NMSETe=var(error_test)/var(tTest);
%computing the  R-Square value
rSqTe=1-sse(error_test)/sum((tTest-mean(tTest)).^2);
fprintf('test result:\n');
fprintf('  - NMSE=%f\n',NMSETe);
fprintf('   MAE     = %g %%\n', mAETe);
fprintf('   R-Square     = %g %%\n', rSqTe);
% plot the result
figure, plot(tTest,'-b','LineWidth',1), hold on, plot(yTest,'-r'...
   ,'LineWidth',1)
set(gcf,'color',[1 1 1])
title(' SVR model Performance on Test Set')
```

```
xlabel('Time (trading days)'), ylabel('Closing Price')
legend('Actual','Predicted')
figure,plot(error_test,'-k')
xlabel('Days')
ylabel('Errors')
title('Error Difference')
%%%%%%%% ARMA model%%%%%%%%%%
timeseriesdata=xlsread('LSEdataset.xlsx');
% Split into training and test data
t=(timeseriesdata);
y= log(timeseriesdata);
N= length(y);
N_train = round( 0.70 * N );% training data = 70% of total.
N_test = N - N_train;% test data = remaining 30%
tTrain = y(1:N_train);
tTest  = y(N_train+1:N);
xTest  = y(N_train+1:N,:);
% Choose best ARMA lags p and q using AIC and BIC:
% Find optimal lags p and q by fitting several models:
maxLags = 10;
LOGL = zeros(maxLags,maxLags);% initialize matrix for loglikelihood
   % values.
PQ = zeros(maxLags,maxLags);% initialize matrix for no. of
   % coefficients.
for p = 1:maxLags
for q = 1:maxLags
mod = arima(p,0,q); % create ARMA(p,q) model.
[fit,~,logL] = estimate(mod,tTrain,'print',false);
% fit model and estimate parameter values.
LOGL(p,q) = logL;    % store loglikelihood.
PQ(p,q) = p+q;       % store no. of coefficients.
end
end
figure
subplot(2,1,1)
autocorr(y)
subplot(2,1,2)
parcorr(y);
% Calculate and display AIC and BIC for each fitted model:
LOGL = reshape(LOGL,maxLags^2,1);
PQ = reshape(PQ,maxLags^2,1);
[aic,bic] = aicbic(LOGL,PQ+1,N_train);
aic = reshape(aic,maxLags,maxLags);
bic = reshape(bic,maxLags,maxLags);
% Find lowest AIC and BIC and best p and q:
[bestAIC, index] = min(reshape(aic, numel(aic), 1));
[bestP_AIC,bestQ_AIC] = ind2sub(size(aic), index);
[bestBIC, index] = min(reshape(bic, numel(bic), 1));
[bestP_BIC,bestQ_BIC] = ind2sub(size(bic), index);
fprintf('Lowest AIC = %f obtained for p = %d, q = %d. \n', ...
bestAIC, bestP_AIC, bestQ_AIC);
fprintf('Lowest BIC = %f obtained for p = %d, q = %d. \n', ...
bestBIC, bestP_BIC, bestQ_BIC);
%BIC(1,2) % - Optimal model: ARMA(1,2)
model=arima(1,0,2);
fit1=estimate(model,y);
model = arima('AR',fit1.AR,'MA',fit1.MA,'Constant',fit1.Constant ...
   ,'Variance',fit1.Variance);
% Fit model to training data:
fit2 = estimate(model,tTrain,'print',false);
%Check the residuals for autocorrelation.
```

```
figure
subplot(2,1,1)
autocorr(res./sqrt(Var))
set(gcf, 'Color', [1 1 1]);
[E0,V0] = infer(fit2,tTrain);
% Forecast:
Y = zeros(N_test,1); YMSE = zeros(N_test,1); V = zeros(N_test,1);
presampleData = tTrain;
for i = 1:N_test
[E0,V0] = infer(fit,presampleData);
[y,ymse,v] = forecast(fit2,1,'Y0',presampleData,'E0',E0,'V0',V0);
presampleData = [tTrain; tTest(1:i)];
Y(i) = y;
YMSE(i) = ymse;
V(i) = v;
end
%------- Computing  the performance measures of model -----------%
%          Display the result of ARMA         %
error_test=tTest-Y;
%mean absolute error
mAETe=mae(error_test);
%NMSE
NMSETe=var(error_test)/var(tTest);
%computing the  R-Square value
rSqTe=1-sse(error_test)/sum((tTest-mean(tTest)).^2);
fprintf('test result:\n');
fprintf('   NMSE=%f\n',NMSETe);
fprintf('   MAE     = %g %%\n', mAETe);
fprintf('   R-Square     = %g %%\n', rSqTe);
Y=exp(Y);
tTest=exp(tTest);
figure
plot( tTest,'b')
hold on
plot( Y,'-r')
title('ARMA model Performance on Test Set')
xlabel('Time (trading days)'), ylabel('Closing price')
legend('Actual','Predicted')
hold off
figure,plot(error_test,'k');
xlabel('Days')
ylabel('Errors')
title('Error between target and predicted stock prices');
 %%%%%k-Nearest Neighbours(k-NN) %%%%%
[x] = xlsread('LSEdataset.xlsx');
[x,mu,sigms]=featureNormalize(x); %normalizatiion
d=1278;
m=2;       % embeding dimension
k=50;       % Number of nearest neighbors
distance='manhattan distance';  % type of distance
[OutSample_For_Corr,InSample_For_Corr,InSample_Res_Corr]= ...
   nn1(x,d,m,k,distance);
disp(InSample_For_Corr)
%Apply De-normalization / convert the data  back into
   % unnormalized units.
X=x(d+1:end);
X=poststd(X,mu,sigms);
InSample_For_Corr=poststd(InSample_For_Corr,mu,sigms);
%plot the result
figure,
set(gcf,'color',[1 1 1]);
```

```
plot(X,'b','linewidth',1)
hold on
plot(InSample_For_Corr,'r','linewidth',1);
%disp(InSample_For_Corr);
xlabel('Time(trading days)');
ylabel('Closing Price');
title(' k-NN model Performance on Test Set'   );
legend(' Actual','Predicted');
figure,
plot(InSample_Res_Corr,'k')
xlabel('Days')
ylabel('Errors')
title( 'Error between target and predicted stock prices ')
%%calclute performance of K-NN model %%
MSE=mse(InSample_Res_Corr);
% calclute normalised mean square error
NMSE=var(InSample_Res_Corr)/var(x(d+1:end));
% Computing mean absolute error
mAb= mae(InSample_Res_Corr);
%compute R-squared
RSq2=1-sse(InSample_Res_Corr)/sum(x(d+1:end)-mean(x(d+1:end).^2));
r=corrcoef(InSample_Res_Corr);
fprintf(' Evaluation of the K-NN model:\n');
fprintf('  - NMSE=%f\n',NMSE);
fprintf('  - MSE=%f\n',MSE);
fprintf('  - MAE=%f\n',mAb);
fprintf('  - R-Square     = %f %%\n', RSq2);
function [x_norm, mu, sigma] = featureNormalize(X)
x_norm = X;
mu = zeros(1, size(X, 2));
sigma = zeros(1, size(X, 2));
mu = mean(X,1);
sigma = std(X,1);
n = length(mu);
for i = 1:n
    x_norm(:,i) = (x_norm(:,i) - mu(i)) / sigma(i);
end
function [OutSample_For,InSample_For,InSample_Res]= nn(x,d,m,k, ...
   distance,n)
if (nargin<4)
    error('Its missing arguments.')
end
if d>=length(x)
error('The value of d must be between 1 and length(x)-1')
end
if (nargin==5)
n=0;
OutSample_For=[];
end
% Main Loop.
for v=0:length(x)-d-1;
Series=x(1:d+v);
[For]=nn_core(Series,m,k,distance);
InSample_For(v+1,1)=For;
fprintf(1,['\nCalculating NN Forecast #',num2str(v+1)]);
end
disp(' ');
InSample_Res=x(d+1:length(x))-InSample_For;
if n~=0
x2=x;
for z=1:n
```

```
[Out_For]=nn_core1(x2,m,k,distance);
OutSample_For(z,1)=Out_For;
x2=[x2;OutSample_For(z)];
end
end
function [For_x]=nn_core(x,m,k,distance);
[n1,n2]=size(x);
chunk = x(n1-m+1:n1,1);
for i=0:n1-m-1;
distance=sum(abs(chunk-x(n1-m-i:n1-i-1,1)));
sum_distance(n1-m-i,1)=distance;
end
[sorted idx] = sort(sum_distance,'descend');
fullIdx = repmat( idx((end-k+1):end),1,m+1) + repmat([0:m],k,1);
s = x(fullIdx);
% Calculate the forecast
For_x(1,1)=mean(s(:,m+1));
end
```