To the Graduate Council:

I am submitting herewith a thesis written by Bukola Titilayo Ojemakinde entitled "Support Vector Regression for Non-Stationary Time Series". I have examined the final electronic copy of this thesis for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Master of Science, with a major in Industrial Engineering.

Charles H. Aikens
Major Professor

We have read this dissertation
and recommend its acceptance:

Adedeji B. Badiru

Myong K. Jeong

Chanaka Edirisinghe

Accepted for the Council:

Anne Mayhew
Vice Chancellor and
Dean of Graduate Studies

(Original signatures are on file with official student records.)

# Support Vector Regression for Non-Stationary Time Series

A Thesis
Presented for the
Master of Science
Degree
The University of Tennessee, Knoxville

Bukola Titilayo Ojemakinde
August 2006

# DEDICATION

This thesis is dedicated to the one man in my life, my husband, David Olatokunbo Ojemakinde, who desires to see me succeed in life. I will forever love and appreciate you baby!

# ACKNOWLEDGEMENTS

I sincerely thank the Almighty God for his steadfast love and for seeing me through, from the start to the completion of my Masters Degree program. It is in Him I live and move and have me being. It is Him that holds all things together and He makes everything beautiful in His time.

I am thankful to Dr. Badiru, Dr. Aikens, Dr. Edirisinghe and Dr. Jeong for serving on my thesis committee. I am especially appreciative of Dr. Badiru for paving the way for me to come to the US to study and for his support and encouragement throughout my program. I am thankful to Dr. Aikens and Dr. Edirisinghe for the valuable time they invested in this study and for providing helpful inputs and feedback throughout the research work. I am also grateful to Dr. Edirisinghe for providing the datasets used for this study. I thank Dr. Jeong for teaching me the method that serves as the highlight of this research. To the other professors in the department and the administrative staff, I say a big “Thank you” for their individual and collective assistance.

I am indebted to my parents for their prayers and encouragement. They have both been there for me since the day I was born and are always longing to see me succeed in all my endeavors. I also appreciate my parents-in-law, Mr. and Mrs. S.O. Ojemakinde, and Mrs. Victoria Omobamiduro, for their prayers and support since I became their daughter. I thank you Mom for your numerous phone calls just to “check up” on me.

I thank all my siblings and my husband siblings for wishing me well and praying for me. I see all of your prayers been answered as I complete my Masters program.

I am indeed very grateful to my husband for standing beside and behind me, and for his sacrifices during my search for knowledge. Sweetheart, you made me reach the heights I cannot get to on my own, calming my fears of what lies ahead. I was so far away from you dear, yet your love for me was always visible in your phone calls, prayers, encouragement and support. Olatokunbo, you are a blessing to me and I truly adore you.

I am thankful to our friends especially Dr. and Mrs. Olufemi Omitaomu for standing as my family here in Knoxville. Dr. Omit, I thank you for your help both in my coursework and in my thesis. May God bless you abundantly and may he take you to greater heights. I am grateful to Ayo Olujide, Busola and Lanre Awoliyi, Akin and Biola Ajayi, Rex and Lola Nwakamma, Funsho and Bola Aduloju, Remi and Deola John, Tokunbo Roluga, Ronke Odeyemi, Bukola Ogunremi, and Muyiwa, Folabi and Deola Oyerokun, Gbemi Popoola, and others too many to mention, for their relationship and phone calls. I thank my Nigerian friends in Knoxville, Dr. and Dr. Mrs. Ijadunola, Gozie Nsofor, Ruby Tasie Kemi Obitayo, and Ibukun Fakoya for their help and support. I say thanks to all the members of the International Class at Cedar Springs Presbyterian church in Knoxville including Paul and Jane Slay, Rachael, and Kelvin and Lydia Zinn for contributing in one way or the other to my physical and spiritual life.

Finally, I say a big thank you to my past professors including Dr. O.E Charles-Owaba, Dr. Oluleye, Dr. Oyawale and Mr. Ofiabulu for their support and encouragement over the years.

# Abstract

The difficulty associated with building forecasting models for non-stationary and volatile data has necessitated the development and application of new sophisticated techniques that can handle such data. Interestingly, there are a lot of real-world phenomena where data that are “difficult to analyze” are generated. One of these is the stock market where data series generated are often hard to forecast because of their peculiar characteristics. In particular, the stock market has been referred to as a complex environment and financial time series forecasting is often tagged as the most challenging application of time series forecasting.

In this study, a novel approach known as Support Vector Regression (SVR) for forecasting non-stationary time series was adopted and the feasibility of applying this method to five financial time series was examined. Prior to implementing the SVR algorithm, three different methods of transformation namely Relative Difference in Percentages (RDP), Z-score and Natural Logarithm transformations were applied to the data series and the best prediction results obtained along with the associated transformation technique was presented. Our study indicated that the Z-score transformation is the best scaling method for financial time series, exhibiting superior

performance than the other two transformations on the basis of five different performance measures.

To determine the optimum values of the SVR parameters, a cross-validation method was implemented. For this purpose, the value of $C$ and $\varepsilon$ was varied from 5 to 100, and 0.001 and 0.1 respectively. The cross-validation method, though computationally expensive, is better than other proposed techniques for determining the values of these parameters.

Another highlight of this study is the comparison of the SVR results to that obtained using 5-day Simple Moving Averages (SMA). The SMA was selected as a comparative method because it has been identified as the most popular quantitative forecasting method used by US corporations. Discussions with financial analysts also suggest that the SMA is one of the widely used in the financial industry. The popularity of the SMA can be explained by the fact that it is easy and cheap to use and it produces forecasts that can be easily interpreted by econometricians and other interested practitioners.

# Table of Contents

# List of Tables

# List of Figures

# Chapter 1

# Introduction

The global nature of financial markets has renewed the interests of financial analysts in advanced techniques for predicting daily closing prices of stocks. Several sophisticated statistical and machine learning techniques have been adapted for financial time series predictions in the last decade. Some of these techniques are Artificial Neural Networks (ANNs), Exponential Smoothing (ES), Autoregressive Integrated Moving average (ARIMA), and Support Vector Regression (SVR). These techniques vary in their accuracy, prediction efficiency, robustness, and transparency.

Financial time series data is especially difficult to analyze because it has inherent noise and non-stationary characteristics. Furthermore, the data is dependent on a lot of quantitative and qualitative factors. Some of the quantitative factors can be incorporated into the model but the qualitative factors are usually unknown and unaccounted for. In some cases, the information provided by the recent data points could provide more valuable information than the preceding data points. Therefore, the analysis of stock prices requires the application of proven techniques.

Both Artificial Neural Network (ANN) and Support Vector Regression (SVR) are new in the financial world but they are techniques often used for linear and nonlinear function approximations. Neural networks have been described as universal approximators because they can map both nonlinear function approximations without any assumptions about the properties of the data. Unlike traditional statistical techniques, ANN is data-driven. However, ANN is known to exhibit inconsistent and unpredictable performance on noisy data (Kim, 2003). In addition, ANN requires a lot of parameter tuning (such as, the number of hidden layers, the number of neurons in each layer, and the type of network architecture to use) that has made its implementation in the real world difficult. ANN also has the tendency of falling into a local minimum.

The Support Vector Machine (SVM) technique was recently developed by Vapnik (1995) for learning classification rules. The technique has been widely applied to problems in pattern recognition (Schmidt, 1996), and regression estimation (Vapnik, 1997; Mukherjee, 1997) due to its remarkable theoretical and practical characteristics. These characteristics include good generalization performance, the absence of local minima, and sparse representation of solutions. When the SVM technique is applied to regression problems it is called Support Vector Regression (SVR). The performance of SVR depends on three training parameters: the type of kernel used, and the respective values of $C$ and $\varepsilon$.

## 1.1 Motivation for the Research

The motivation for this research is the perceived need in the financial world to be able to rapidly predict stock prices without having to expend extensive computation time trying

to fine-tune model parameters. Participants in the financial marketplace are constantly involved in making decisions that will meet their objectives on risks and returns. Those decisions are often based on their predictions or anticipations of stock market phenomena. However, obtaining accurate market projections is difficult because many of the properties common to financial time series data inhibit the ability to apply proven tools of analysis and prediction. According to Cao and Tay (2001) and Abecasis et al (1999), financial time series forecasting is one of the most challenging applications of modern time series forecasting.

Generally, the financial market has been tagged as a very complex environment, involving the interplay of decisions and actions, taken by millions of individual investors and institutions on a global scale (Pantazopoulos et al, 1998). This explains why data accumulated in the financial industry are often noisy, highly volatile (non-stationary), and deterministically chaotic. The data is noisy because there is no complete information from the past behavior of the series to fully explain the relationship between the future and the past data points (Cao, 2002). Furthermore, financial data are known to be non-stationary, and unfortunately, most of the available techniques were developed based on an assumption of stationarity. The term "stationarity" suggests that the distribution for both the patterns and noise in the series remains the same over the model estimation and forecasting period (Cao and Tay, 2001). Not only is a single data series stationary in the sense of the mean and variance of the series, but the relationship of the data series to other related data series is also expected to remain the same over time (Tay and Cao, 2002).

Stock data has also been described as chaotic (Tay and Cao, 2001), which implies that the data is characterized with a long term trend but with small short term fluctuations. The presence of chaos in financial time series also indicates that financial systems are characterized by non-repetitive and non-predictable fluctuations arising from the inter-play of a system's participants, and its relation to other systems (Pantazopoulos et al, 1998). Therefore, it may be inappropriate to use the information deduced from past values to try to fully explain the behavior of future values.

Given the nature of financial time series data and the quest of financial engineers to find the best possible prediction systems, researchers and econometricians have continued over the years to search for forecasting procedures with good generalization/predictive abilities. Several univariate and multivariate procedures have been applied to financial time series forecasting and new methodologies have been developed with the aim of obtaining good predictions of the actual measurements.

Objective of this research is to explore the use of SVR for stock price prediction by performing a comprehensive experimental study, and then comparing the experimental results with one of the conventional methods used for predicting stock prices.

## 1.2 Accomplishments of the Thesis

The following were achieved by conducting this study:

- Models based on Support Vector Regression were developed and the feasibility of applying these models for stock price predictions was examined. Five different stock price data were used.

- The best combination of SVR parameters that could be used for building Support Vector Regression models was identified for each stock data and compared similarities between these values.
- The performance of the best SVR model with the Simple Moving Average (SMA) model, which is one of the popular conventional time-series forecasting techniques, was compared.
- The performance of SVR with SMA was discussed and the best model that can be used to predict each one of the stock price data was identified.

## 1.3 Outline of the Thesis

This thesis is organized as follows. A literature review of techniques used for time series predictions is presented in Chapter 2. Chapter 3 describes a theoretical framework of SVR. In Chapter 4, a description of the data used was included, the methodology followed for this study was outlined, and a discussion of the experimental results was presented. The conclusions of the study are presented in Chapter 5 together with some suggestions for extending the work.

# Chapter 2

# Literature Review

A time series data is a continuous set of observations occurring at equally spaced periods, which may be recorded daily, weekly, monthly, quarterly, or yearly. However, the closing stock prices considered in this thesis are recorded daily except during major business holidays such as weekends. There are various time based data in the real world including daily closing prices of stocks, average daily temperature of a city, and daily records of goods sold in a store. Financial time series is one of the most widely analyzed time based data because of its economic importance; however, it is an unusual data because of its specific characteristics. Some of these characteristics are discussed here:

## Noisy Data

Financial time series data usually possess relatively low signal-to-noise (SNR) ratio which indicates that most of the factors responsible for the actual data cannot be accounted for or explained. SNR is the relative magnitude of the useful information in the data compared to the embedded uncertainty or noise. In order to avoid modeling the noise in the data, some of the noise is usually reduced or removed by using techniques such as smoothing or filtering, but this produces a lag problem. The problem of lagging

occurs when the smoother is tracking the actual series data but does not tell us much about the future of the series. In such scenarios, the model can only make forecasts for very few periods in advance.

In addition, the removal or reduction of the noise cannot guarantee an accurate model because the noise is a part of the overall system environment in which financial transactions take place, and represent qualitative factors that drive that sector. One possible implementation strategy is to develop models that can incorporate an understanding of such noise.

## Non-stationarity

Financial time series data fluctuates with respect to changes in the operating conditions of the systems that generate them. However, the complete definition of such operating conditions is not available because financial data depend to a large extent, on factors that analysts cannot explain or quantify. Therefore, the data will have the propensity toward statistical inconsistencies, possessing different statistical properties (for example mean and variance) at each point in time. This makes the forecasting of such data difficult. A common technique for overcoming this problem is to take the difference of data points. Although good models can be developed by first taking differences of data points in the training data set, there is no assurance of obtaining good performance of such models on the test data set especially for non-stationary time series, since the non-stationarity in the test set has not been captured by the model.

## Uncertainty

The world of finance is about risk and uncertainty. Risk is characterized by randomness whose likelihood of occurrence can be measured precisely whereas uncertainty is present when such randomness is indefinite or incalculable. Uncertainty in financial data arises from several sources and occurs in varying degrees. Several models have been developed that are founded on the concept of using the variance/standard deviation of the series to estimate its uncertainty (Dionísio et al, 2005). Central to an effective implementation strategy would be a fundamental understanding of the sources of uncertainty, and the development of models that have been designed to reduce or eliminate them. Until such models are possible or feasible, available techniques must be used to develop prediction models that will produce some global albeit uncertain solutions.

Time series prediction typically uses observed time-ordered data to predict the future values of the series, such as

$$..., x_{t-3}, \ x_{t-2}, \ x_{t-1}, \ x_t, \ ?, \ ?, ...$$

*where*

$x_t$ is the value of the series at time t

$x_{t-1}$ is the value of the series at time t-1 and so on

The following section gives a brief introduction into how to build a time series forecast system.

## 2.1 Forecasting Framework

Forecasting is an integral part of planning in any system, whether in business or government. Modeling a real-world problem and doing forecasting can offer objective information for future development. The flowchart in Fig 2.1 highlights different phases in modeling a real-world system. This contains several functions:

**Model Estimation:** Understand the underlying machinery that generates the data or controls the system; this includes describing and explaining any variation, seasonality, and trend.

**Forecast Generation:** Predict the future based on the assumption that everything remains constant; that is, business as usual.

**Forecast Updating:** Control the system, which is to perform the "what-if" scenarios.

## 2.2 Data Processing

Fitting a prediction model to time series data is a crucial and expensive task. Depending on the problems, it may be necessary to perform some data preprocesses in order to satisfy the requirements of the techniques being used. For instance, preprocessing can be used to remove seasonal or trend effects, or cyclic oscillations. Without preprocessing, for example, it may be incorrectly inferred that recent increase patterns will continue indefinitely when actually the increase is simply because it is that time of the year. Two methods commonly used for data preprocessing are smoothing and differencing.

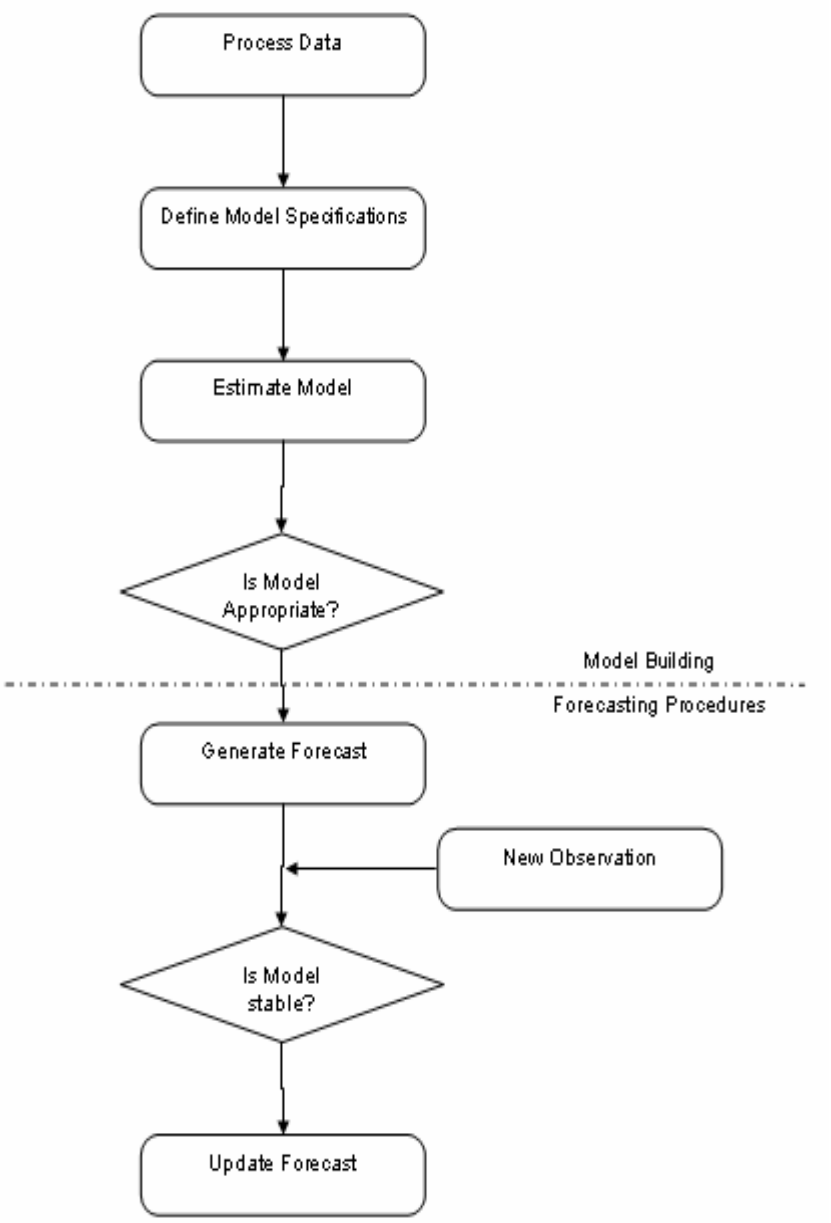



Figure 2.1: Model Building and Forecasting Phases of a Forecasting System.

## Smoothing

One characteristic of time series data is the presence of random variation. One commonly used technique for reducing such random variation is smoothing. This technique dampens the random variation component thereby revealing any underlying trends, or seasonal and/or cyclic components in the original data.

Smoothing methods can be categorized in two ways: Simple Moving Averages (SMA) or Exponential Moving Averages (EMA).

**1. Simple Moving Average (SMA):** a *k*-day SMA takes the average of previous *k* days' values and presents the average as the current day's value.

For a given data series $x_1, x_2, x_3, x_4, ....., x_n$, after calculating the SMA with $k = 3$, the new (SMA) series becomes;

$$\left[\left(x_1 + x_2 + x_3\right)/3\right],\ \left[\left(x_2 + x_3 + x_4\right)/3\right]\ ,.....,\ \left[\left(x_{n-2} + x_{n-1} + x_n\right)/3\right]$$

For the purpose of illustration, consider a particular series $A = 120,\ 124,\ 122,\ 123,\ 125,\ 128,\ 129,\ and\ 127$ that represent the sales of a furniture company for time period $t = 1, 2, ....., 8.$

SMA3 for time periods $t = 4, 5, ....., 9$ is given by;

$SMA3 = -,\ -,\ -,\ 122,\ 123,\ 123.33,\ 125.33,\ 127.33,\ 128$

**2. Exponential Moving Average (EMA):** a *k*-day EMA begins by setting the first point in the series, $EMA_1 = x_1$, and thereafter, the $i$-day's value is computed as

$$EMA_i = EMA_{i-1} * (1-\frac{2}{k}) + x_i * \frac{2}{k} \qquad (2.1)$$

For k=3 the EMA series becomes

$$x_1,\ \frac{1}{3}x_1+\frac{2}{3}x_2,\ \frac{1}{9}x_1+\frac{2}{9}x_2+\frac{2}{3}x_3,\ \frac{1}{27}x_1+\frac{2}{27}x_2+\frac{2}{9}x_3+\frac{2}{3}x_4,\ ....$$

Using series A again for the purpose of illustration, EMA3 can be calculated as follows;

$$EMA3 = 120,\ 122.67,\ 122.22,\ 122.74,\ 124.25,\ 126.75,\ 128.25,\ 127.42$$

Comparing the SMA and EMA series, several observations can be made.

1. Taking $k$-days' SMA will reduce the number of data points in the series by $k$, while EMA still retains the same number of original data. In calculating SMA3 in our example, we lost the first two data points as these points were used to calculate the first SMA occurring at time $t = 4$. This is not the case for EMA3 since the first number in the EMA3 series is the first number in the original series.

2. EMA gives more weight to the latest data than SMA. In SMA, a constant weight of $1/k$ is attributed to all the points used in calculating the moving averages while in EMA, varying weights are attributed to the data points with the most recent data point receiving a weight of $2/k$. In our example a weight of $1/3$ was attributed to all points

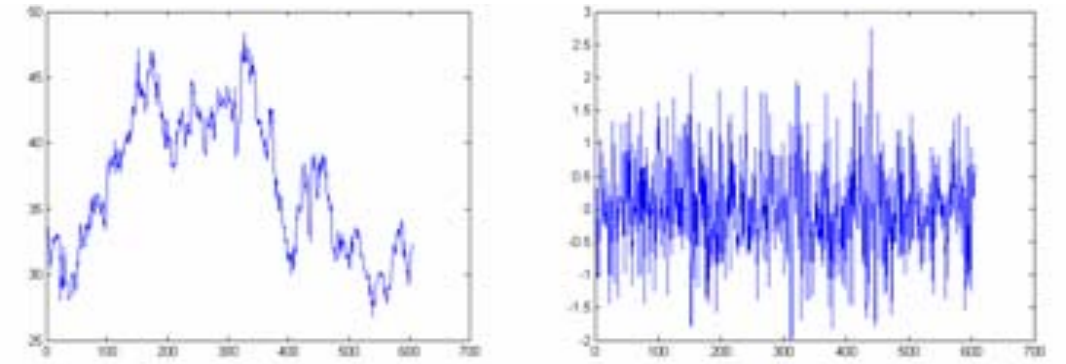
using SMA3 but varying weights were used in EMA3 with the most recent data point receiving a weight of $2/3$.

3. Since recent data points receive more weights than the more distant ones, EMA tends to react faster to recent value changes than SMA.

## Differencing

Differencing is an alternative method for preprocessing when there is a substantial trend in the data. The differencing technique transforms a data series $x_1, x_{2,} x_3, .....$ into the new series of consecutive differences $(x_2 - x_1), (x_3 - x_2), ....$. In general, the original time series $x_t$ is transformed into a new series $y_t$ where $y_t = x_t - x_{t-1}$ for $t = 2, 3, ....$. This procedure usually achieves stationarity in the mean. If not, differencing can be applied again, creating a third series $z_t$ where $z_t = y_t - y_{t-1}$ for $t = 3, 4, ....$. This procedure may be repeated until the series becomes stationary. Several observations concerning the differencing technique are important: 1) each differencing transformation decreases the total number of data points by one; 2) the data points are no longer independent since each point in the series, after the first one, shares two points in common from the original set; and 3) the random noise in the transformed data has been amplified, because the noise in each difference point represents the cumulative variance present in the two points comprising the difference.

Figure 2.2 illustrates an original price and its result after first differencing.



a) Original prices b) First Differences

Figure 2.2: Prices of McDonalds Stock from July 03, 1995 to Dec. 31 1997

## 2.3 Forecasting Procedure

Forecasting the future values of an observed time series is an important problem in many applications. Forecasting procedures have been classified according to the following taxonomy (Chatfield, 2001; 2004):

**Subjective** forecasts made based on personal judgment, beliefs, commercial knowledge, and any other "unscientific" information.

**Univariate** forecasts entirely based on past observations in a given time series, by fitting a model to the data and extrapolating. For instance, forecasts of future sales of a product would be based entirely on past sales.

**Multivariate** forecasts made by taking other observations or other variables into account. For example, stock price may depend on the political situation in neighboring countries. Regression models are one of these types of models.

More sophisticated and robust forecasting models may involve a combination of the above approaches (Chatfield, 2004). A model is said to be robust if it is unaffected by small variations in its parameters and/or changes in the assumptions used in its construction.

## 2.4 Model Descriptions

The literature and practice norms abound with examples of univariate and multivariate models for time series analysis. All models can be conveniently classified as either linear or nonlinear (Chatfield, 2001) as illustrated in Figure 2.3.

### 2.4.1 Linear Models

Linear models have the following characteristics: simplicity, usefulness, and ease of application. As the classification implies, linear models are best suited for linear stationary time series, but may fail otherwise and especially in cases of non-stationarity as in financial data series. Three types of linear models have been widely applied namely;

#### ARIMA and Its Variations

The Autoregressive Integrated Moving Average (ARIMA) approach, first proposed by Box and Jenkins (Box and Jenkins, 1976; 1994), has evolved over the last 30 years to include several variations of the original model.

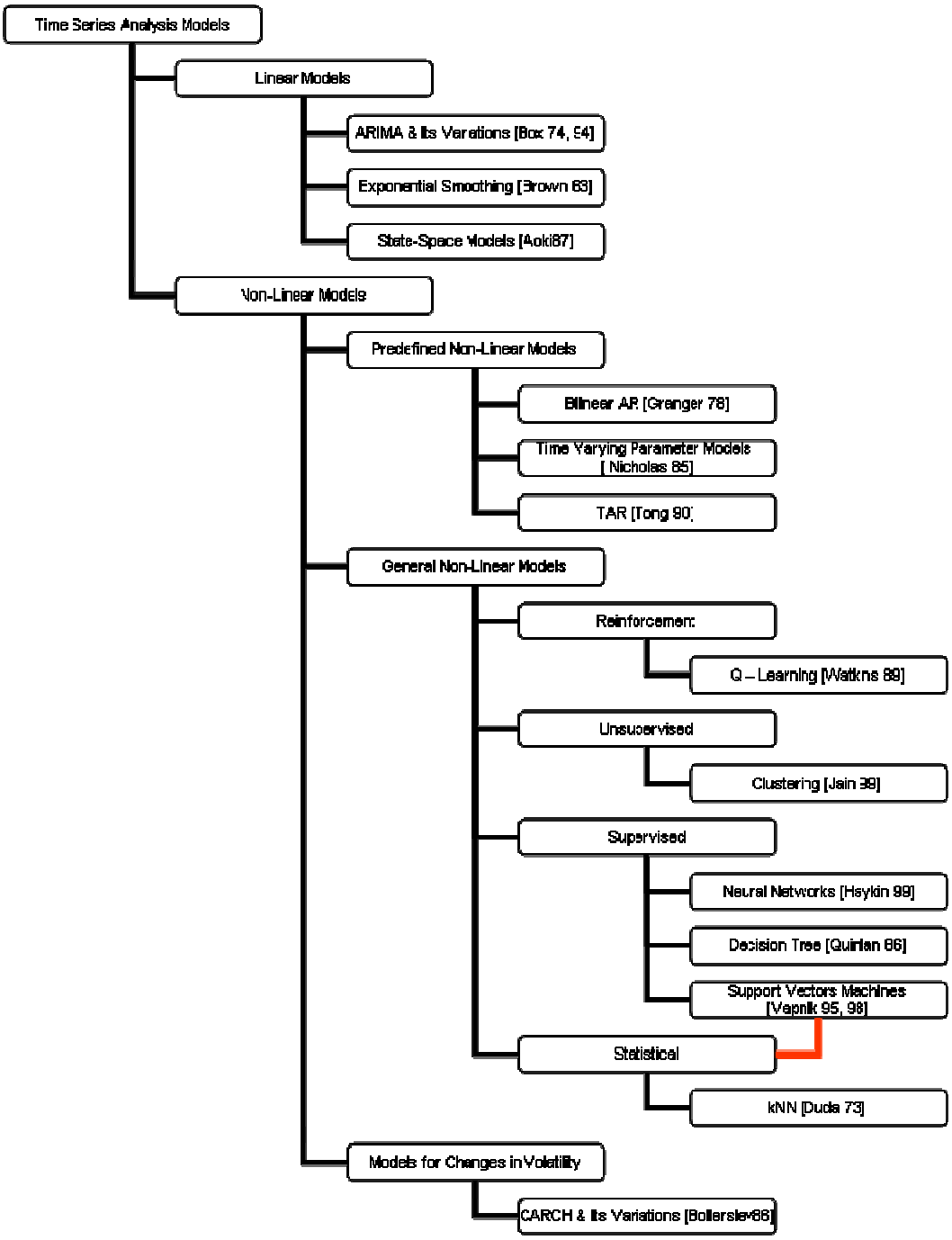



Figure 2.3: Time Series Analysis Models

The idea behind ARIMA is to apply differencing to a non-stationary time series until it becomes stationary, and then to apply a mixture of Autoregressive and Moving Average (ARMA) models. This procedure consists of five stages (Box and Jenkins, 1994):

(a) **Differencing:** If the data is non-stationary, the data will be differenced until it becomes stationary.

(b) **Model Identification:** The essence of this stage is to examine the data to identify the model, i.e., to determine which order of $p$ and $q$ will be most appropriate for the model, where $p$ is the order of auto-regressiveness, and $q$ is the order of moving average. In general, there is no optimal way to do this. Some useful tools are the sample autocorrelation, (ACF) and partial autocorrelation (PACF) functions. ACF measures the correlation between different lags of a time series, while PACF measures the residual correlation after the correlation implied from earlier lags is partialed out.

(c) **Estimation:** In this stage, the parameters of the chosen model are estimated. Least squares method is the method usually used to find the parameters. A detailed description is available in (Box and Jenkins, 1994).

(d) **Diagnostic Checking:** To check whether the model that has been selected is adequate. One method is to examine the residuals from the fitted model.

(e) **Alternative Models Consideration:** If the fitted model appears to be inadequate for any reason, then other ARIMA models may be tried until a satisfactory model is found.

### Exponential Smoothing

Exponential smoothing is another type of linear model (Brown, 1963) that works well for linear time series but fails to model complicated nonlinearity and trends in financial time series. A variant of this model is sometimes applied in the data preprocessing stage.

### State Space Models

State space models (Aoki, 1990) are a class of linear models that represent inputs as a linear combination of a set of state vectors that evolve over time according to some linear equations. Different state space formulations cover a range of models and include the so-called structural models defined in (Harvey, 1989) as well as the dynamic linear models in (West and Harrison, 1997), where the latter uses a Bayesian formulation. Models called unobserved component models by econometricians are also of the state-space form. However, in practice, the state vectors and associated dimensions of these models are hard to choose (Chatfield, 2001).

## 2.4.2 Non-linear Models

Although linear models possess mathematical and practical convenience, there is no reason to assume that real life time series are always linear; therefore the application of non-linear modeling holds promise (Chatfield, 2001). Three classes of models that have been popularized for nonlinear time series data are predefined models, general models, and models for change in volatility.

## Predefined Non-linear Models

In the 1980s, non-linear models were investigated and proposed as modifications of existing linear models, for example ARIMA models, as seen in studies presented by Granger and Joyeux (1980), and Priestley (1981). This type of models includes Bilinear Autoregressive models (Granger and Andersen, 1978), Time Varying Parameter models (Raj and Ullah, 1981; Nicholls and Pagan, 1985), and Threshold Autoregressive (TAR) models (Tong, 1990). These models are similar in the degree of scrutiny given in their development, and the standard statistical considerations of model specification, estimation and diagnosis, but their general parametric nature tends to require significant a' priori knowledge of the form of the relationship being modeled. Therefore, these methods are not effective for modeling financial time series because the underlying nonlinear functions are difficult to choose.

## General Non-linear Models

General non-linear models, also called machine learning form an alternative nonlinear modeling class. These models can "learn" the underlying data structure of a given time series without having to explicitly make non-linear assumptions. Models in this class include Reinforcement Learning, for example, Q-learning (Watkins, 1989), Unsupervised Learning, for example, clustering methods (Jain, Murty, and Flynn, 1999), Supervised Learning, for example, decision trees (Quinlan, 1986), and Neural Networks (NN) (Ripley, 1993; Cheng and Titterington, 1994; Baestaens, 1994; and Haykin, 1999), and Statistical Learning which includes k-nearest neighbors (kNN) (Duda and Hart, 1973). Support Vector Machines (SVMs) are new learning machines that can also model

nonlinear relationships of the data and are based on statistical learning or Vapnik-Chervonenkis (VC), theory. In addition, SVMs are modeled using a training sample with target in order to make predictions of outcomes in a future testing sample. Consequently, SVMs are used for statistical and supervised learning.

## Models for Change in Volatility

The focus of models for changes in volatility is change in variance. The objective of these models is to give better estimates of local data variance so that more reliable prediction intervals can be computed, leading to a better assessment of risk (Chatfield, 2001). Models for change in volatility are not designed to give better point forecasts of future observations in the series. The estimation of local variance is especially important in financial applications, where observed time series often show clear evidence of changing volatility, for example large absolute values tend to be followed by larger (absolute) values, while small absolute values are often followed by smaller values, indicating high or low volatility, respectively. To estimate the local variance, Engle in 1982 proposed Autoregressive Conditionally Heteroskedastic (ARCH) model (Engle, 1982; 1995). According to Tsay (2002), the basic ideas of ARCH models are:

1. The mean-corrected asset return is serially uncorrelated, but dependent, and

2. The dependence of asset return at time t can be described by a simple quadratic function of its lagged values.

An ARCH model with order $p$, in short ARCH ( $p$ ), assumes that the variance $\sigma_t^2$ at time $t$, is linearly dependent on the squares of the last $p$ values of the time series, that is,

$$r_t = \sigma_t \varepsilon_t,\ \sigma_t^2 = \alpha_0 + \alpha_1 r_{t-1}^2 + ... + \alpha_m r_{t-m}^2, \tag{2.2}$$

where $r_t$ is a serial asset return, $\varepsilon_t$ is a sequence of independent and identically distributed $(i.i.d.)$ random variables with mean zero and variance 1, and $\alpha_i$ are coefficients that must satisfy some regular conditions to ensure that the unconditional variance of $r_t$ is finite with $\alpha_0 > 0, \alpha_i > 0$ for $i > 0$. In practice, $\varepsilon_t$ is often assumed to follow the standard normal or a standardized student-$t$ distribution.

ARCH models were extended by Bollerslev (Bollerslev, 1986) resulting in the generalized ARCH (GARCH) model. Similar to ARMA, a GARCH model can be used to estimate a high order ARCH model with fewer parameters. A GARCH model with order $p$ and $q$, in short $GARCH(p,q)$, assumes that the conditional variance depends on the squares of the last $p$ values of the series and on the last $q$ values of conditional variance, that is,

$$r_t = \sigma_t \varepsilon_t,\ \sigma_t^2 = \alpha_0 + \sum_{i=1}^{p} \alpha_i r_{t-i}^2 + \sum_{j=1}^{q} \beta_j \sigma_{t-j}^2, \tag{2.3}$$

where again $\varepsilon_t$ is a sequence of $i.i.d.$ random variables with mean zero and variance 1, and $\alpha_0 > 0, \alpha_i > 0,\ \beta_j > 0$, and $\sum_{i=1}^{p} \alpha_i + \sum_{j=1}^{q} \beta_j < 1$.

The $GARCH(1,1)$ model has become the `standard' model for describing changing variance for no reason other than relative simplicity. There are also some extensions of the basic GARCH model, such as Quadratic GARCH (QGARCH) and Exponential GARCH (EGARCH). The QGARCH models allow for negative `shocks'(errors) to have more effect on the conditional variance than positive `shocks'. The EGARCH models allow an asymmetric response by modeling $\log\sigma_t^2$, rather than $\sigma_t^2$ . Summaries of this family of models can be seen in (Engle, 1995).

Although GARCH models are applied in a wide range of problems usefully, they do have limitations:

1. GARCH models are only part of a solution. Although GARCH models are usually applied to financial return series, financial decisions are rarely based solely on expected returns and volatilities.

2. GARCH models are parametric specifications that operate best under relatively stable market conditions (Gourieroux, 1997). Although GARCH is explicitly designed to model time-varying conditional variances, GARCH models often fail to capture highly irregular phenomena, including wild market fluctuations (for example crashes and subsequent rebounds), and other highly unanticipated events that can lead to significant structural change.

3. GARCH models often fail to fully capture the fat tails observed in asset return series. Heteroskedasticity explains some of the fat tail behavior, but typically not all of it.

Fat tail distributions, for example, student-$t$, have been applied in GARCH modeling, but often the choice of a specific distribution is a matter of trial and error.

### 2.4.3 ARMA Models

ARMA models are linear models used to capture the linear correlation between any specified lags of a univariate time series and the error term of the model from previous time points. In general, an $ARMA(p,q)$ model can be written as

$$x_t = \mu + \alpha_1 x_{t-1} + \alpha_2 x_{t-2} + ... + \alpha_p x_{t-p} + \varepsilon_t + b_1 \varepsilon_{t-1} + ... + b_q \varepsilon_{t-q,} \qquad (2.4)$$

where $\varepsilon_t$ is a sequence of $i.i.d.$ random variables with mean zero and variance 1, $\mu$ is the mean of the time series and $\alpha's$ and $b's$ are constant coefficients.

Moving Average $(MA)$ models are special cases of $ARMA$ models. In these models, the observation in time $t$ depends on the error term of the model from previous time points, usually these errors are considered as random events (Chatfield, 2004).

Generally, an $MA(q)$ model is

$$x_t = \mu + \varepsilon_t + b_1 \varepsilon_{t-1} + ... + b_q \varepsilon_{t-q} \qquad (2.5)$$

From the above formula, we can see that this $MA$ model is different from the smoothing moving average method described in Subsection 2.2 even though it bears the same name.

Autoregressive ($AR$) models form another special class of ARMA models. In these models, the observation in time $t$ is regressed not on other independent variables but on one or more of the lagged values of the time series (Chatfield, 2004; Box and Jenkins, 1994). A general form of an $AR(p)$ model is

$$x_t = \mu + \alpha_1 x_{t-1} + \alpha_2 x_{t-2} + ... + \alpha_p x_{t-p} + \varepsilon_{t,} \tag{2.6}$$

where $\varepsilon_t$ is a purely random process (also called white noise) with mean zero and variance $\sigma_\varepsilon^2$.

The simplest example of an AR model is the first-order case, i.e. $AR(1)$, which takes the following form;

$$x_t = \mu + \alpha x_{t-1} + \varepsilon_t \tag{2.7}$$

For $\mu = 0$ and $\alpha = 1$, the AR (1) is the random walk model popularized in operations research literatures.

$$x_{t+1} = x_t + \varepsilon_{t+1} \tag{2.8}$$

This model is based on the Efficient Market Hypothesis (EMH) postulated by Fama (Fama, 1965; 1970) and widely accepted throughout the financial community (Malkiel, 1987; Tsibouris and Zeidenberg, 1995). If the EMH is true, then the best estimation of a financial time series is:

$$\hat{x}_{t+1} = x_t \tag{2.9}$$

where $\hat{x}_{t+1}$ is the forecast for time $t+1$ and $x_t$ is the current actual.

That is, if a time series is truly a random walk, then the best estimate for the next time period is equal to the current estimate.

In summary, ARMA models are a combination of AR and MA models. An advantage of ARMA models is the ability to describe a stationary time series using fewer parameters than if an MA or an AR model is used by itself (Chatfield, 2004).

### 2.4.4 Support Vector Machines

Support Vector Machines (SVMs) which first appeared at COLT (Conference of Computational Learning Theory) 1992 (Boser, Guyon and Vapnik, 1992) are grounded in statistical learning or Vapnik Chervonenkis theory (also known as VC theory), which was first developed by Vapnik and his co-workers (Vapnik, 1979; 1995; and 1998). SVMs have the following characteristics.

#### Bounded Generalization Error

SVMs are based on the VC theory, which claims to guarantee generalization, that is, the generalization error is bounded by the sum of the training error (empirical risk) plus a term that depends on the VC dimension of the learning machine (Vapnik, 1979; and 1998).

## Geometric Interpretation

SVMs were initially proposed to solve classification problems where the objective is not only to minimize the empirical risk, but also to maximize the margin (Vapnik, 1995; Bennett and Bredensteiner , 2000).

## Global and Unique Solution

Training SVM requires the solving of a Quadratic Programming (QP) problem over a solution space known to be convex. Therefore, every local optima will also be a global solution. Hence, SVM training always finds a global solution that is usually unique (Burges and Crisp, 2000). This is superior to NN, a technique that often results in the identification of local optima (Burges, 1998).

## Mathematical Tractability

Using a kernel function, SVMs offer an alternative training technique for Polynomial, Radial Basis Function, and Multi-Layer Perceptron classifiers, in which the weights of the network are found by solving a Quadratic Programming (QP) problem with linear inequality and equality constraints. This is generally a technique preferred to the NN training regimen that requires the solution of a non-convex, unconstrained minimization problem (Osuna, Freund, and Girosi, 1997).

The advantages that SVMs have over other methods have considerable interest, and a wide range of SVM applications have produced excellent results (CherKassky and Mulier, 1998; Scholkopf, Burges, and Smola, 1999). These applications have included pattern recognition, such as handwritten digit recognition (Boser et al, 1992; Cortes and

Vapnik, 1995), face detection in images (Osuna et al, 1997), and text classification (Joachims, 1998).

# Chapter 3

# Support Vector Regression

## 3.1 Introduction to Support Vector Machines

The concept of a support vector machine (SVM) was recently developed by Vapnik and his co-workers at AT&T (Vapnik, 1995). SVM is an optimization technique that attempts to find a hyperplane in the original input space to separate a given training set correctly and leave as much distance as possible from the closest instances to the hyperplane on both sides. In regression estimation, the data points that realize the maximal margin are called support vectors. In other words, they are the data points whose approximation errors are equal to or larger than the so-called tube size of SVM. If the training set is not linearly separable, then a nonlinear boundary has to be constructed. In order to achieve the nonlinear boundary, the original input space is mapped into a higher dimensional space called feature space. The feature space is then searched for a hyperplane that can separate the instances in the same feature space. The mapping from the input space to the feature space is defined by a kernel function. The technique also allows for misclassification by introducing a penalty factor $C$ in the optimization model and the total penalty is found by summing up penalties on each misclassification. Therefore, the technique finds a hyperplane that minimizes the sum of the reciprocal of

the margin and the total penalty. The combined penalty function is stated as the objective function in the optimization model.

Since it was first introduced, SVM has been studied extensively and used for several applications such as pattern recognition, hand written character, and text categorization (Joachims, 1997; Scholkpf and Burges, 1995; Schmidt, 1996). As a result of its performance in real world classification problems, the principle of SVM has been extended to regression problems (Smola and Scholkopf, 1998). In SVM literature, when the SVM algorithm is used for classification problems, it is called Support Vector Classification (SVC) and when it is used for regression problems, it is called Support Vector Regression (SVR). Some of the attractive properties of SVR are the use of kernel functions that make the technique applicable to both linear and non-linear approximations, good generalization performance as a result of the use of only the support vectors for prediction, the absence of local minima because of the convexity property of the objective function and its constraints, and the fact that the methodology is based on structural risk minimization that seeks to minimize the generalization rather than the training error.

## 3.2 Support Vector Regression Formulation

In this section, a detailed presentation of the theory behind SVR equations is given, based on the formulation by Vapnik (1995). Considering a data training set, *T*, represented by:

$$T = \{(x_1, y_1), (x_2, y_2), ..., (x_m, y_m)\}, \tag{3.1}$$

where $x \in X \subset \mathbb{R}^n$ are the training inputs and $y \in Y \subset \mathbb{R}$ are the training outputs. Assume a non-linear function, $f(x)$ given by:

$$f(x) = \mathbf{w}^{\mathbf{T}}\mathbf{\Phi}(\mathbf{x}_i) + b. \tag{3.2}$$

where $\mathbf{w}$ is the weight vector, $b$ is the bias, and $\mathbf{\Phi}(\mathbf{x}_i)$ is the high dimensional feature space, which is linearly mapped from the input space $x$. Assume further that the goal is to fit the data $T$ by finding a function $f(x)$ that has a largest deviation $\varepsilon$ from the actual targets $y_i$ for all the training data $T$, and at the same time is as small as possible. Therefore, Eq. (3.2) is transformed into a constrained convex optimization problem as follows:

$$\begin{aligned} &\text{minimize} \quad \frac{1}{2}\mathbf{w}^{\mathbf{T}}\mathbf{w} \\ &\text{subject to:} \quad \begin{cases} y_i - \left(\mathbf{w}^{\mathbf{T}}\mathbf{\Phi}(\mathbf{x}_i) + b\right) \le \varepsilon \\ y_i - \left(\mathbf{w}^{\mathbf{T}}\mathbf{\Phi}(\mathbf{x}_i) + b\right) \ge \varepsilon, \end{cases} \end{aligned} \tag{3.3}$$

where $\varepsilon(\ge 0)$ is user defined and represents the maximum acceptable deviation.

Eq. (3.3) can be rewritten as:

$$\begin{aligned} &\text{minimize} \quad \frac{1}{2}\mathbf{w}^{\mathbf{T}}\mathbf{w} \\ &\text{subject to:} \quad \begin{cases} y_i - \mathbf{w}^{\mathbf{T}}\mathbf{\Phi}(\mathbf{x}_i) - b \le \varepsilon \\ \mathbf{w}^{\mathbf{T}}\mathbf{\Phi}(\mathbf{x}_i) + b - y_i \le \varepsilon. \end{cases} \end{aligned} \tag{3.4}$$

The goal of the objective function in Eq. (3.4) is to make the function as "flat" as possible; that is, to make $\mathbf{w}$ as "small" as possible while satisfying the constraints. In order to solve Eq. (3.4), slack variables are introduced to cope with possible infeasible optimization problems. One silent assumption here is that $f(x)$ actually exists; in other words, the convex optimization problem is *feasible*. However, this is not always the case; therefore, one might want to trade off errors by flatness of the estimate. This idea leads to the following primal formulations as stated in Vapnik (1995):

$$\begin{aligned} &\text{minimize} && \frac{1}{2}\mathbf{w}^{\mathbf{T}}\mathbf{w} + C\sum_{i=1}^{m}\left(\xi_i^{+} + \xi_i^{-}\right) \\ &\text{subject to:} && \begin{cases} y_i - \mathbf{w}^{\mathbf{T}}\boldsymbol{\Phi}(\mathbf{x}_i) - b \le \varepsilon + \xi_i^{+} \\ \mathbf{w}^{\mathbf{T}}\boldsymbol{\Phi}(\mathbf{x}_i) + b - y_i \le \varepsilon + \xi_i^{-} \\ \xi_i^{+}, \xi_i^{-} \ge 0. \end{cases} \end{aligned} \tag{3.5}$$

where $C$ ( > 0) is a pre-specified regularization constant and represents the weight of the loss function. The first term in the objective function $\left(\mathbf{w}^{\mathbf{T}}\mathbf{w}\right)$ is the regularized term and makes the function as "flat" as possible whereas the second term $\left(C\sum_{i=1}^{m}\left(\xi_i^{+} + \xi_i^{-}\right)\right)$ is called the empirical term and measures the $\varepsilon$-insensitive loss function. According to Eq. (3.5), all data points whose *y*-values differ from $f(x)$ by more than $\varepsilon$, are penalized. The slack variables, $\xi_i^{+}$ and $\xi_i^{-}$ correspond to the size of this excess deviation for upper and lower deviations, respectively, as represented graphically in Fig. 3.1. The $\varepsilon$-tube is the largest deviation and all the data points inside this tube do not contribute to the regression model since their coefficients are equal to zero.

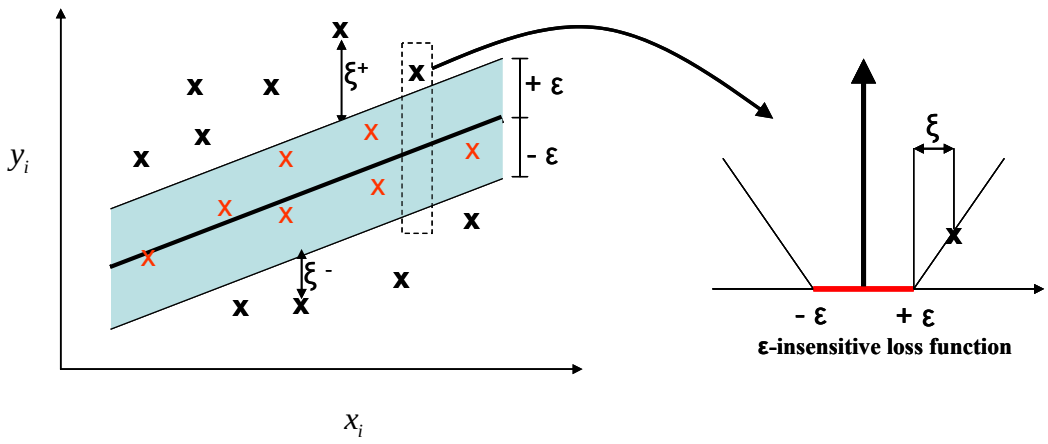



Fig 3.1: Left, the Tube of $\varepsilon$ Accuracy and Points that do Not Meet this Accuracy. The back dots located on or outside the tube are support vectors. Right, the (linear) $\varepsilon$ -insensitive loss function is shown in which the slope is determined by C.

Data points outside this tube or lying on this tube are used in determining the decision function and they are called support vectors and have non-zero coefficients. Eq. (3.5) assumes $\varepsilon$ -insensitive loss function (Vapnik, 1995) as shown in Fig. 3.1 and defined as:

$$|\xi|_\varepsilon = \begin{cases} 0 & \text{if } |\xi| \le \varepsilon \\ |\xi| - \varepsilon & \text{otherwise.} \end{cases} \quad (3.6)$$

To solve Eq. (3.5), some Lagrangian multipliers $\left(\alpha_i^+, \alpha_i^-, \eta_i^+, \eta_i^-\right)$ are introduced in order to eliminate some of the primal variables. Hence, the Lagrangian of Eq. (3.5) is given as:

$$
\begin{aligned}
L_p = \frac{1}{2}\mathbf{w}^{\mathbf{T}}\mathbf{w} + C\sum_{i=1}^{m}\left(\xi_i^+ + \xi_i^-\right) - \sum_{i=1}^{m}\left(\eta_i^+\xi_i^+ + \eta_i^-\xi_i^-\right) \\
- \sum_{i=1}^{m}\alpha_i^+\left(\varepsilon + \xi_i^+ - y_i + \mathbf{w}^{\mathbf{T}}\boldsymbol{\Phi}\left(\mathbf{x}_i\right) + b\right) \\
- \sum_{i=1}^{m}\alpha_i^-\left(\varepsilon + \xi_i^- + y_i - \mathbf{w}^{\mathbf{T}}\boldsymbol{\Phi}\left(\mathbf{x}_i\right) - b\right) \\
s.t. \quad \alpha_i^+, \alpha_i^-, \eta_i^+, \eta_i^- \geq 0.
\end{aligned}
\tag{3.7}
$$

The Eq. (3.7) formulation permits the extension of SVM to nonlinear functions. It follows from the saddle point condition (the point where the primal objective function is minimal and the dual objective function is maximal) that the partial derivatives of $L_p$ with respect to the primal variables $\left(\mathbf{w}, b, \xi_i^+, \xi_i^-\right)$ have to vanish for optimality.

Therefore,

$$\partial_b L_P = \sum_{i=1}^{m}\left(\alpha_i^+ - \alpha_i^-\right) = 0, \tag{3.8}$$

$$\partial_{\mathbf{w}} L_P = \mathbf{w} - \sum_{i=1}^{m}\left(\alpha_i^+ - \alpha_i^-\right)x_i = 0, \tag{3.9}$$

$$\partial_{\xi_i^{(*)}} L_P = C - \alpha_i^{(*)} - \eta_i^{(*)} = 0, \tag{3.10}$$

where (*) denotes variables with + and - superscripts. Substituting (3.8) and (3.10) into (3.7) lets the terms in $b$ and $\xi$ vanish. In addition, Eq. (3.10) can be transformed into $\alpha_i \in \left[0, C\right]$. Therefore, substituting Eqs. (3.8) to (3.10) into (3.7) yields the following dual optimization problem:

$$\text{maximize} \quad \frac{1}{2}\sum_{i,j=1}^{m} K\left(\mathbf{x}_i, \mathbf{x}_j\right)\left(\alpha_i^+ - \alpha_i^-\right)\left(\alpha_j^+ - \alpha_j^-\right) + \varepsilon\sum_{i=1}^{m}\left(\alpha_i^+ + \alpha_i^-\right) - \sum_{i=1}^{m} y_i\left(\alpha_i^+ - \alpha_i^-\right) \tag{3.11}$$

$$\text{subject to} \quad \begin{cases} \sum_{i=1}^{m}\left(\alpha_i^+ - \alpha_i^-\right) = 0 \\ \alpha_i^+, \alpha_i^- \in [0, C] \end{cases}$$

where $K\left(\mathbf{x}_i, \mathbf{x}_j\right)$ is the kernel function. The flexibility of a kernel function allows the technique to search a wide range of the solution space. The kernel function allows non-linear function approximations with the SVM technique, while maintaining the simplicity and computational efficiency of linear SVM approximations. A kernel function must be positive definite in order to guarantee a unique optimal solution to the quadratic optimization problem. Some of the common kernel functions are polynomial kernel and Gaussian radial basis function kernel.

The dual problem in Eq. (3.11) has the following three advantages.

- The optimization problem is now a quadratic programming problem with linear constraints, which is easier to solve than Eq. (3.7) and ensures a unique global optimum.
- The input vector only appears inside the dot product, which ensures that the dimensionality of the input space can be hidden from the remaining computations. That is, even though the input space is transformed into a high dimensional space, the computation does not take place in that space but in the linear space (Gunn, 1998).

- Finally, the dual form does allow the replacement of the dot product of input vectors with a non-linear transformation of the input vector.

In deriving Eq. (3.11), the dual variables $\eta_i^+, \eta_i^-$ were already eliminated through the condition in Eq. (3.10). Therefore, Eq. (3.9) can be rewritten as:

$$\mathbf{w} = \sum_{i=1}^{m} \left( \alpha_i^+ - \alpha_i^- \right) x_i \tag{3.12}$$

Hence, Eq. (3.2) becomes:

$$f(x) = \sum_{i=1}^{m} \left( \alpha_i^+ - \alpha_i^- \right) K\left( \mathbf{x}_i, \mathbf{x}_j \right) + b \tag{3.13}$$

This is the support vector regression expansion. That is, **w** can be completely described as a linear combination of the training patterns $\mathbf{x}_i$. A considerable advantage is that Eq. (3.13) is independent of both the dimensionality of the input space $\chi$ and the sample size *m*.

## 3.3 Types of Kernel Functions used in SVM

$K\left( \mathbf{x}_i, \mathbf{x}_j \right)$ is defined in Eq. (3.11) as the kernel function. Its value is equal to the inner product of two vectors $\mathbf{x}_i$ and $\mathbf{x}_j$ in the feature space $\mathbf{\Phi}\left( \mathbf{x}_i \right)$ and $\mathbf{\Phi}\left( \mathbf{x}_j \right)$. That is,

$$K\left( \mathbf{x}_i, \mathbf{x}_j \right) = \mathbf{\Phi}\left( \mathbf{x}_i \right) \cdot \mathbf{\Phi}\left( \mathbf{x}_j \right) \tag{3.14}$$

$K\left(\mathbf{x}_i, \mathbf{x}_j\right)$ is used in the SVM techniques to map the input space into a high-dimensional feature space through some non-linear mapping chosen *a' priori* and used to construct the optimal separating hyperplane in the feature space. This makes it possible to construct linear decision surfaces in the feature space instead of constructing non-linear decision surfaces in the input space. There are several types of kernel function used in SVM. The type of SVM constructed is a function of the selected kernel function and affects the computation time of implementing the SVM.

Three of the most popular kernel functions used in SVM are:

1) A polynomial kernel function constructed using:

$$K\left(\mathbf{x}_i, \mathbf{x}_j\right) = \left(x_i, x_j\right)^d, \ d = 1, \ldots \tag{3.15}$$

where $d$ is the degree of the polynomial.

An alternative more computationally efficient form of Eq. (3.15) is:

$$K\left(\mathbf{x}_i, \mathbf{x}_j\right) = \left(\left(x_i, x_j\right) + 1\right)^d, \ d = 1, \ldots \tag{3.16}$$

2) A Gaussian radial basis kernel function can be constructed using:

$$K\left(\mathbf{x}_i, \mathbf{x}_j\right) = \exp\left(-\frac{\left(x_i - x_j\right)^2}{\sigma^2}\right), \tag{3.17}$$

where $\sigma$ (> 0) is the kernel width.

3) A sigmoid kernel function constructed using:

$$K\left(x_i, x_j\right) = \tanh\left(b\left(x_i \cdot x_j\right) + c\right) \tag{3.18}$$

where $b$ is the slope and $c$ is the bias associated with the function.

The radial basis function (RBF) kernels are widely used in artificial neural networks (Haykin, 1999), support vector machines (Vapnik, 1998), and approximation theory (Schölkopf & Smola, 2002). The RBF kernel is usually a reasonable first choice because of its outstanding features: it can handle linear and non-linear input-output mapping effectively; it requires less number of hyper-parameters than polynomial kernel, which reduces computation cost in terms of tuning for optimum hyper-parameters; the kernel values for RBF ranges between 0 and 1, hence less numerical difficulties; whereas these values can range between 0 and infinity for polynomial kernel. The sigmoid kernel is often not considered because it does not always fulfill the Mercer Condition (Vapnik, 2000), a requirement for an SVR kernel. In addition, sigmoid kernel is similar to RBF kernel when the kernel width is a small (Lin & Lin, 2003).

## 3.4 Methods of Computing SVR Parameters

The performance of SVR technique depends on the setting of three training parameters (kernel, $C$, and $\varepsilon$) for $\varepsilon$-insensitive loss function. However, for any particular type of kernel the values of $C$ and $\varepsilon$ affect the complexity of the final model. The value of $\varepsilon$ affects the number of support vectors (SV) used for predictions. Intuitively, a larger value of $\varepsilon$ results in a smaller number of support vectors, which leads to less complex

regression estimates. On the other hand, the value of $C$ is the trade off between model complexity and the degree of deviations allowed in the optimization formulation. Therefore, a larger value of $C$ undermines model complexity (Cherkassky and Ma, 2004). The selection of optimum values for these training parameters ($C$ and $\varepsilon$) that will guarantee less complex models is an active area of research. There are several existing approaches for selecting optimum value for these parameters.

The most common approach is based on users' prior knowledge or expertise in applying SVM techniques (Cherkassky and Mulier, 1998; Schölkopf et al., 1999). However, this approach could be subjective and it is not appropriate for new users of SVR. Mattera and Haykin (1999) proposed that the value of $C$ be equal to the range of output values; but this approach is not robust to outliers (Cherkassky and Ma, 2004). Another approach is the use of cross-validation techniques for parameter selection (Cherkassky and Mulier, 1998; Schölkopf et al., 1999). Even though this is a good approach for batch processing, it is data-intensive hence very expensive to implement in terms of computation time, especially for larger datasets. One more approach is to select $\varepsilon$ values in proportion to the variance of the input noise (Smola et al., 1998; Kwok, 2001). This approach is independent of the sample size and is only suitable for batch processing where the entire data set is available. Cherkassky and Ma (2004) propose an alternative approach based on the training data. They propose that $C$ values should be based on the training data without resorting to re-sampling, and by using the following estimation:

$$C = \max\left(\left|\bar{y} + 3\sigma_y\right|, \left|\bar{y} - 3\sigma_y\right|\right), \tag{3.19}$$

where $\bar{y}$ and $\sigma_y$ are the mean and standard deviation of the *y* values of the training data. This approach has the advantage that it is robust to possible outliers. Cherkassky and Ma (2004) also suggest that the value of $\varepsilon$ should be proportional to the standard deviation of the input noise. Using the Central Limit Theorem, they proposed that $\varepsilon$ be given by:

$$\varepsilon = 3\sigma\sqrt{\frac{\ln n}{n}}, \tag{3.20}$$

where $\sigma$ is the standard deviation of the input noise and *n* is the number of training samples. Since the value of $\sigma$ is not known *a priori*, the following equation can be use to estimate $\sigma$ using the idea of the *k*-nearest-neighbor's method:

$$\hat{\sigma} = \sqrt{\frac{n^{1/5}k}{n^{1/5}k-1}\cdot\frac{1}{n}\sum_{i=1}^{n}\left(y_i-\hat{y}_i\right)^2}, \qquad 2 \le k \le 6 \tag{3.21}$$

where *n* is the number of training samples, *k* is the low-bias/high variance estimators, and $\hat{y}$ is the predicted value of *y* by fitting a linear regression to the training data to estimate the noise variance. Cao and Tay (2003) propose ascending regularization constant $\left(C_i\right)$ and descending tube $\left(\varepsilon_i\right)$ for batch SVR applications in financial time series data. They adopt the following definitions:

$$C_i = C\frac{2}{1+\exp\left(a-2a\left(\frac{i}{m}\right)\right)} \tag{3.22}$$

and

$$\varepsilon_i = \varepsilon \frac{1+\exp\left(b - 2b\left(\frac{i}{m}\right)\right)}{2}, \tag{3.23}$$

where $i$ represents the data sequence, $C_i$ is the ascending regularization constant, $\varepsilon_i$ is the descending tube, $a$ is the parameter that controls ascending rate, and $b$ is the parameter that controls descending rate. The computation of the parameter that controls the ascending and the descending rates also requires re-sampling techniques (Cao and Tay, 2003).

## 3.5 Applications of SVR in Financial Time Series Prediction

Since its development in 1995, Support Vector Regression has continued to gain increasing popularity and it is indeed a Herculean task to try to report all the contributions of SVR that have advanced in the area of financial time series forecasting since its inception. In this study, we give a concise overview of some of the reported applications.

Several researchers have attempted to use SVR for making future projections of different types of financial data as found in the literature. For instance, Cao and Tay (2001; 2002; and 2003) examined the feasibility of applying SVM to five financial time series data and compared its performance with models developed using multi-layer back-propagation (BP) neural network and the regularized radial basis function (RBF) neural network. For the SVM model, they proposed adaptive parameters to capture the non-stationarity property of the data. Their results show that SVM performs better than BP neural network in financial forecasting and possesses comparable generalization performance

when compared to the regularized RBF neural network. Their analyses also show that incorporating adaptive parameters into SVM leads to an improved generalization performance and a sparser representation of the solution as compared to the use of standard SVM for financial forecasting.

Cao et al (2002) proposed the $\varepsilon$-Descending support vector machine ($\varepsilon$-DSVMs) for performing regression analyses on financial time series data. The model constructed based on the $\varepsilon$-DSVMs differs from the original SVM model in that it uses an adaptive tube to handle structural changes in the data as opposed to the constant tube employed in the standard SVM. In using the $\varepsilon$-DSVMs to model financial time series, more weight is placed on recent data points than distant data points, on the premise that recent data points provide more information that will be valuable for prediction than the distant data points. The performance of the $\varepsilon$-DSVMs in making accurate forecasts of two time series was investigated and the outcomes in terms of certain performance metrics show that $\varepsilon$-DSVMs has superior generalization ability, and fewer number of support vectors than the standard SVM.

Cao et al. (2003) also proposed another model; $C$-ascending Support Vector Machine (CASVM) which is similar to the $\varepsilon$-DSVMs. CASVM was the result of a simple modification of the regularized risk function associated with SVMs whereby the recent $\varepsilon$-insensitive errors are penalized more heavily than the distant $\varepsilon$-insensitive errors. With a similar motivation for developing the $\varepsilon$-DSVMs, CASVM was proposed based on the prior knowledge that in non-stationary financial time series, the recent past data

could provide more information than the distant past data. Experimental results showed that CASVM also generalizes better than the standard SVM.

Kim (2003) applied SVM to predict the future direction of stock price indices and compared the results obtained with artificial neural networks (ANN) and case-based reasoning (CBR). The effects of the regularization constant C and the radial basis function kernel parameter $\delta^2$ were analyzed, and it was concluded that these two parameters have an impact on the prediction performance of SVM.

As an extension of the standard SVM, Yang et al. (2002) developed an SVR model that uses different types of margins for the forecasting process. The model takes into consideration the volatility that characterizes most if not all financial data and thus uses the standard deviation of the data to calculate a variable margin that can be used for its prediction. The application of the Yang et al. model to financial time series forecasting (specifically Hang Seng Index) was studied with the conclusions that modified SVM models with asymmetrical margins (margin adaptations) lead to the reduction of the downside risks associated with prediction of financial data.

## 3.6 Other Applications of Support Vector Regression

Support Vector Regression has been applied to time series data outside the financial domain. To show the applicability of SVM in process chemo-metrics, Thissen et al. (2003) performed time series prediction of some process parameters using SVM, and compared its performance to the results obtained using Elman recurrent neural networks (also known as recurrent neural networks (RNNs), and autoregressive moving average

(ARMA) models. Experiments were conducted using three different data sets – a simulated dataset generated according to the ARMA principles, the Mackey Glass dataset usually used for benching and a real-life industrial data set. The results showed that SVM outperforms ARMA and the Elman Network in the Mackey Glass data set, outperformed the Elman model but predicted equally well as the ARMA model on the real world data set. For the simulated dataset, the ARMA model was found to perform best while the Elman Network and SVM performed similarly. This is explainable by the fact that the ARMA model was able to build a parametric model similar to the underlying system.

# Chapter 4

# Methodology and Experimental Results

## 4.1 Description of Data

The data set used for our experiments consists of the daily closing prices of five real-life stocks - American Airlines, McDonalds, Sears, Toys 'R us and Microsoft. Stock prices were collated from the New York Stock Exchange and represent the closing prices at which these stocks were traded during daily trading sessions from July 01, 1995 to June 31, 1997.

A total of 485 data points were used for all the experiments. The first 365 data points represented the training set and the remaining 120 data points represented the test set so that genuine out-of- sample forecasts can be made and compared with the actual observations.

## 4.2 Time Plots of the Data Series

When presented with a time series, the first step in the analysis is usually to plot the observations against time to give what is called a time plot. Most analysts agree that the time plot is the most important tool in any time series analysis or forecasting study

(Chatfield, 1997). The time plot of the series serves as a descriptive tool that can guide an analyst in choosing the appropriate techniques to use for forecasting such a series. The time plot may show trend and seasonal variations present in the data, and also reveal discontinuities, turning points, and wild observations (outliers) that do not conform to the rest of the data. Other features to look for in a time plot include sudden or gradual changes in the properties of the series. Generally, the plot is vital both to describe the data and to help in formulating a sensible model.

The graphical representations of the prices of each stock are shown in Figures 4.1 to 4.5. From these plots, we can see that the representation of each stock price is somewhat non-linear. These plots also portray the non-stationarity and volatility associated with each of these selected stock. These patterns are typical of most financial time series.

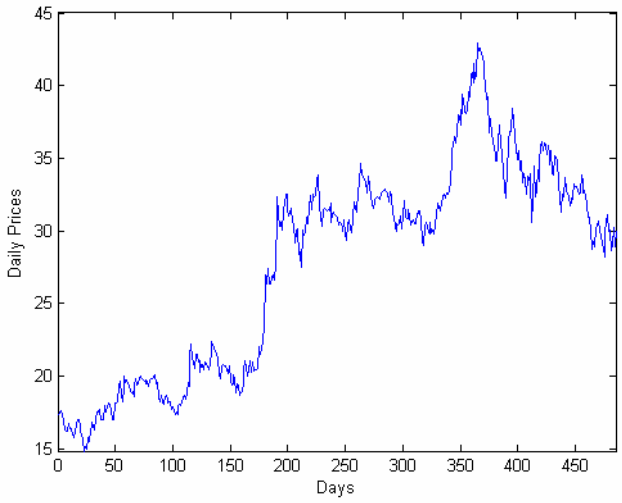



Figure 4.1: Time Plot of Stock Price Traded by American Airlines

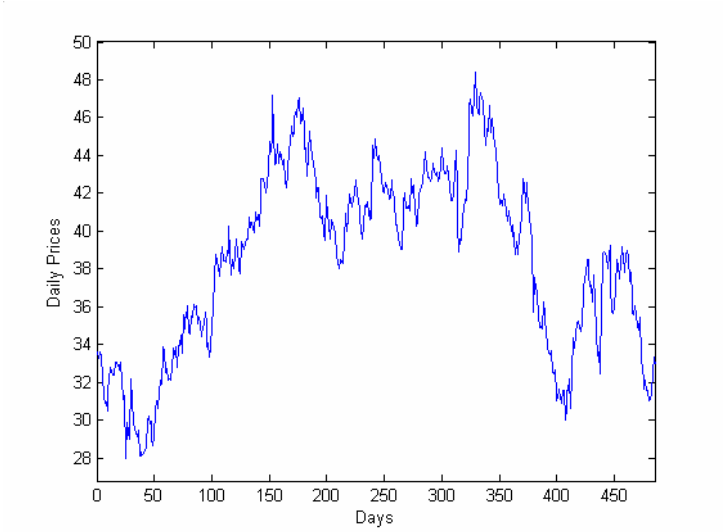



Figure 4.2: Time Plot of Stock Price Traded by McDonalds

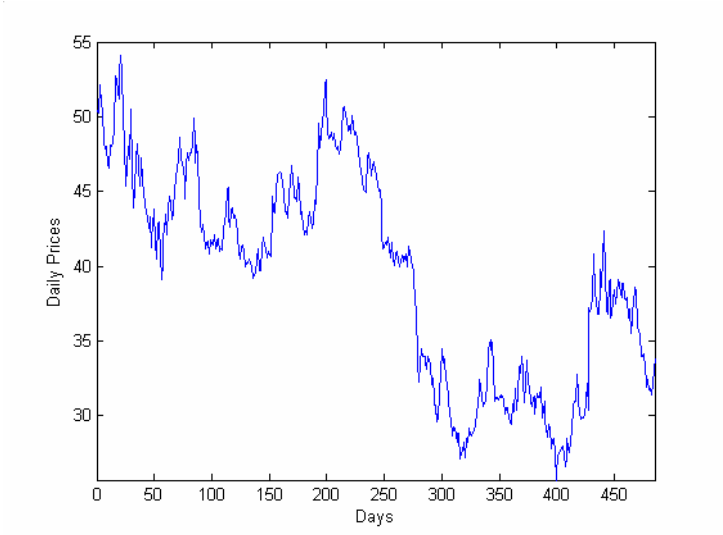



Figure 4.3: Time Plot of Stock Price Traded by Sears

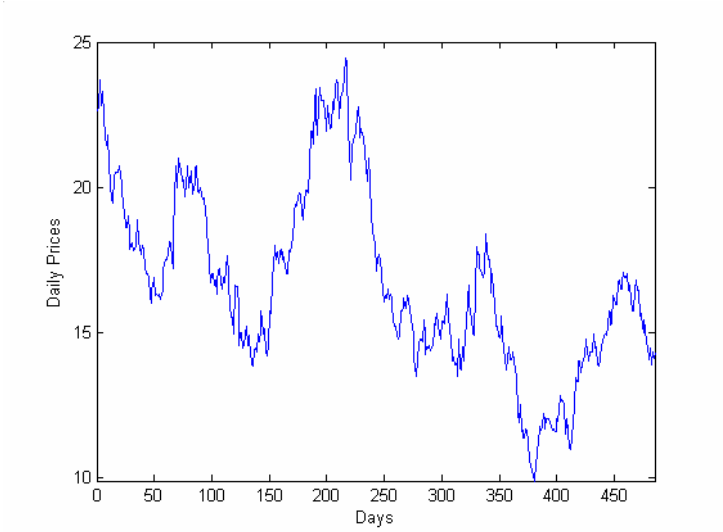



Figure 4.4: Time Plot of Stock Price Traded by Toys ‘R us

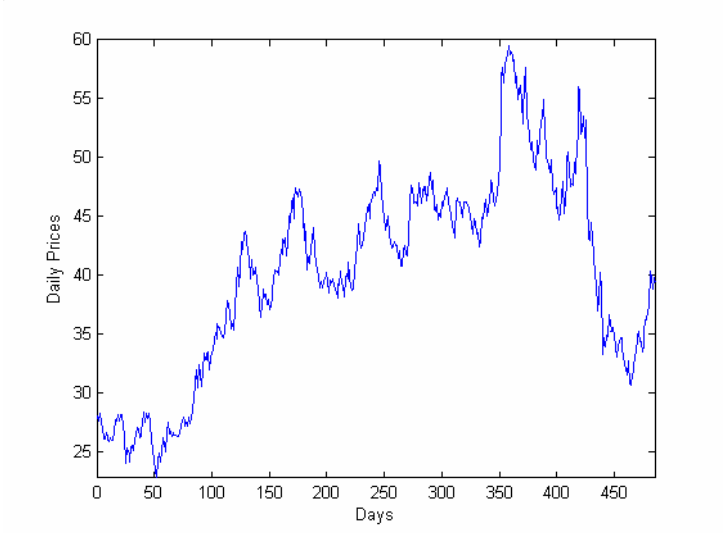



Figure 4.5: Time Plot of Stock Price Traded by Microsoft

## 4.3 Data Preprocessing

Data pre-processing is a vital procedure that includes tasks that must be performed prior to building the forecasting models. Two tasks that fall into this category are data cleaning (identification of missing values, outliers and their elimination or adjustments) and, data scaling and/or transformations.

### 4.3.1 Data Cleaning and Identification of Missing Values

An initial examination of the data is necessary to access its quality, check for errors, missing values, and outliers. The process of checking through data is often called data cleaning or data editing. It is an essential precursor to attempts at modeling data. Data cleaning could include modifying outliers, identifying and correcting obvious errors, and filling in (or imputing) any missing observations. It is important to implement some form of data cleansing/pre-processing function prior to developing the SVR-based prediction model. If the data applied to the model development is incorrect, the resulting model will be incapable of making valid predictions (Thomason, 1999). Therefore, it is imperative to ensure that the data is void of missing values, outliers, errors and so on. One of the ways to accomplish this is to perform a simple visual inspection of the graphical representations of the data. The time plots of the stock prices as shown in Figures 4.1 to 4.5 does not suggest the presence of missing values in the data and the data patterns are consistent with the non-stationarity and volatility that one expects in similar financial time series. However, the plots of the five stock price series suggest that some data values are extremely larger than the rest. To ascertain that there are indeed unusual

observations in each of the series, other diagnostics were employed as discussed in the next section.

### 4.3.2 Outlier Identification and Adjustments

Outliers are defined as those data values that are outside a previously defined normal range of the data. For this study, "the normal range" of data was defined as $\pm 2$ standard deviations. For each of the five stocks, any data value smaller than -2 standard deviations or bigger than +2 standard deviations was considered an outlier. These data values once identified were then processed as extreme values by replacing them with values equal to those limits according to their sign. This procedure for treating outliers was proposed by Thomason, 1999 and was used by Cao and Tay (2003); Tay and Cao (2001, 2002). We have also employed this method for identifying outliers.

### 4.3.3 Scaling and Transformation

The objective of our study is three-fold. First, we seek to make forecasts that are close to the actual values of the five series in spite of the unique features associated with the data sets. Second, we wish to be able to correctly predict the direction of movement of the stocks and finally, we seek to find the type of data transformation that will result in the best performance in terms of the other two afore-mentioned objectives. In order to achieve these objectives using the SVR model, different pre-processing procedures were implemented to enhance the forecasting ability of the model.

The high-volatility components associated with financial data are often very difficult to model successfully; hence, a scaling and/or transformation process is usually performed on the series prior to implementing the actual experiments. These transformation

methods are often employed to eliminate the non-stationarity of the mean and/or variance of the time series. However, there are many other reasons for scaling or transforming a series including the following;

- Time plots of transformed financial data are often more useful in detecting outliers than the time plots of the absolute or original data.
- Generally, transformations enhance the performance of the forecasting models than using the original series (Thomason, 1999)
- Transformation helps to remove undesirable biases in the data such as long-term trends, inflation, and so on.
- Transformations often result in data that possess attractive statistical properties.

Several forms of scaling and transformations have been applied to financial time series as found in previous studies. For the purpose of comparing results, three different transformations were used on our datasets prior to developing the SVR Model. The three transformations used in our study are described below.

## Relative Difference in Percentage

Relative Difference in Percentage (RDP) also referred to as the one-day simple return in certain literatures is undoubtedly the most popular transformation used in financial time series studies and is defined by the following equation:

$$RDP = \left[ \frac{P_t - P_{t-1}}{P_{t-1}} \right] * 100 \qquad (4.1)$$

where $RDP$ is the Relative Difference in Percentage, $P_t$ and $P_{t-1}$ are the original stock prices at period $t$ and period $t-1$ respectively.

According to Tsay (2002), most studies on financial time series are based on Relative Differences in Percentages rather than the actual prices of assets. Transforming to RDPs has been found to produce several benefits. Firstly, the relative magnitudes of stock prices tend to produce more meaningful values than the precise values of the daily prices. This is due to the fact that the RDP represents the return of an asset and is a complete and scale-free summary of the investment opportunity (Campbell, Lo and Mackinlay, 1997). Secondly, data series transformed into RDPs are easier to handle than price series in their original scales because the former have more attractive statistical properties in that it will become more symmetrical and will follow more closely to a normal distribution (Thomason, 1999). Thirdly, Thomason suggested that the plots of data transformed into Relative Differences in Percentage tend to reveal outliers better than the plots of the actual prices (Thomason, 1999).

One of the past studies in which RDP was used is the work of Cao and Tay (2003) where original closing prices of each of five real futures contracts collated from the Chicago Mercantile Market were transformed into a five-day relative percentage difference of price prior to developing an SVR model.

## Z-score Normalization

The Z-score transformation has also been considered as a potentially useful transformation that produces similar benefits as the Relative Percentage Difference. According to Thomason (1998), this procedure is particularly suitable for normalizing time series with outliers. Several researchers have employed the Z-score transformation. For example, Cao and Tay (2003) performed the Z-score normalization on the time series used in their study. Cheadle et al (2003) analyzed Microarray data using Z-score transformation, and Ozgur (2004) developed a neural network model for predicting financial performance of publicly traded Turkish firms by preprocessing the input data using Z-score method of scaling.

The Z-score transformation is defined by the following equation.

$$Z = \frac{p_t - \mu_p}{\sigma_{\mathrm{p}}} \tag{4.2}$$

where $p_t$ is the stock price at period $t$, $\mu_p$ and $\sigma_{\mathrm{p}}$ is the mean and standard deviation of the series respectively.

## Logarithmic Transformation

Frequently, econometricians use the logarithmic transformation because of the general belief that the change in logarithm of variables tends to approximate percentage changes, or rate of returns. It is also believed that the variability of a series appears to be related to the overall mean, so that using logarithms may produce relationships with more homogeneous residuals (Nelson and Granger, 1979). On this premise, logarithmic

transformations are often applied to time series data in order to achieve a more homogeneous variance in the residuals. Another advantage of logarithmic transformation is that taking natural logarithms of return series often produce a more stationary series. This is particularly useful for financial time series which are generally believed to be non-stationary (Tsay, 2002).

Applications of logarithmic transformation abound in the literatures. For example, Abecasis and Lapenta, (1996) modeled and forecast the behavior of non-stationary, univariate, real-world time series representing daily prices of Corn, Soya and Wheat by first mapping each of these series to stationary ones using a logarithmic transformation as a benchmark. Also, prior to developing time series models, Nogales et al, (2002) applied logarithmic transformation to data representing the prices and demand of electricity in order to achieve homogeneity of variance. Nelson and Granger, (1979) also applied logarithmic transformations to 21 different time series and showed that using such transformations results in better forecast performance than using the untransformed series.

To achieve a logarithmic transformation with our stock price data, the following equation was applied;

$$y_t = \ln(p_t) \tag{4.3}$$

where $y_t$ is the transformed price and $p_t$ is the original price of the stocks.

## 4.4 Forecasting Models: SVR and SMA

For the purpose of comparing results, two different models were employed for making forecasts of stock prices on yet-to-be seen samples. The first is Support Vector Regression (SVR) which is the focus of this study and the second is Simple Moving Average (SMA). Simple Moving Average is one of the simplest forecasting methods that work on the assumption that the future value will equal to the average of past values. Although it is a very good data preprocessing tool for smoothing out random variations as discussed in Section 2.2, it also serves as a very useful technique for modeling random series (that is, one without trend or seasonality) because it averages out the most recent actual values to remove the unwanted randomness (DeLurgio, 1998). An N-Period moving average denotes that each new forecast moves ahead one period by adding the newest actual data point and dropping the oldest actual data point.

We decided to compare results obtained from SVR with that of SMA because SMA is one of the most popular method used in time series forecasting. Sanders and Manrodt (1994) conducted a survey of forecasting practices in US corporations and found Moving Averages to be the most familiar and most used quantitative technique for short range and medium range forecasts.

## 4.5 Model Parameters for SVR and SMA

There are five major parameters that must be defined prior to constructing the SVR and SMA models. These parameters are the embedding dimension, the forecast horizon, the type of kernel function, the regularization constant $C$, and the maximum allowable

deviation $\varepsilon$. The first two parameters are required in building both the SVR and SMA models, while the last three parameters are required only for developing the SVR model. In the next two sections, we discuss each of these five parameters briefly and explained how they were defined in our experiments.

### 4.5.1 Embedding Dimension and Forecast Horizon

Choosing a suitable embedding dimension and forecast horizon is the first step in modeling. These two parameters are very important in developing our SVR and SMA models because they form the basis for which the input-output patterns are generated.

We define "Embedding" as the number of back-lagged observations (relative to the present time instance) of a series that can be used to construct the input pattern with which predictions can be made for a current data point. In time series prediction that involves using either SVR or SMA, an input pattern $x_i$ to the model, is a re-constructed series that consists of a finite set of consecutive measurements of the original series. In the case of our daily stock price data, the re-constructed series can be represented by the following:

$$x_i = \{p_{(t_i)}, p_{(t_i - s)}, \ldots, p_{(t_i - \tau s)}\} \tag{4.4}$$

where $p_{(t_i)}$ is the stock price at the most recent time instance $t$ for the input pattern $i$, $s$ is the sampling time step and $\tau$ is an embedding size. The embedding size is usually an integer factor that determines the time window and hence the number of elements of the input pattern.

Virtually all forecasting models require the definition of the forecast horizon. Generally, forecast horizon is the number of periods into the future for which values are to be estimated. Since SVR and SMAs require the use of several input patterns, we will redefine the forecast horizon as the number of periods into the future for which forecasts are to be made given a particular input pattern. Mathematically, we can represent the output of the models for each input pattern given;

$$y_i = p_{(t_i+h)} = f\{p_{(t_i)}, p_{(t_i-s)}, \ldots, p_{(t_i-\tau s)}\} \tag{4.5}$$

where $h$ is the forecast horizon representing the period into the future for which a value $p_{t_i+h}$ of the price series is to be estimated for each input pattern $i$. Usually, the forecast horizon is the same for all the input patterns generated in the experimentation process.

The forecast horizon can be set to short-term (one-period or two-periods ahead), medium-term, or long term (multi-period ahead). Although there are no universally accepted rules that governs a forecaster's choice of forecast horizon, more often than not, the choice of a suitable horizon depends on the interests of the analyst, nature of the data being analyzed, associated transaction costs and how closely the prediction system is to be monitored (Thomason, 1999). For instance, for highly volatile data, it may be advisable to use short-term forecast horizon. Although using a short-term forecast horizon has limitations that include greater transaction costs and inability to generate multiple period ahead forecasts that can make planning easier, analysts dealing with financial time series are forced to employ this form of horizon for their prediction systems because of the high-

volatility and non-stationarity associated with most of the financial data they handle. Most authors/researchers do also agree that short-horizon forecasts are typically more accurate than longer-horizon forecasts (Delurgio, 1998).

When performing time series prediction, the embedding dimension becomes an additional tunable parameter. In the literature, several authors have used different embedding sizes. For example**,** Cao and Gu (2002) examined the feasibility of SVR to three different time series data using embedding values of 8, 12 and 20 respectively; Tay and Cao (2001), Cao et al (2003) constructed the input pattern for their series using a combination of four (4) lagged values based on 5-day time steps and another variable that was obtained by subtracting a 15-day exponential average from each of the data points. Yang et al (2002,) used an input window of four (4), modeling their prediction system as $x_t = f\left(x_{t-4}, x_{t-3}, x_{t-2}, x_{t-1}\right)$; Lee and Billings (2002) applied a variant of SVR on three different datasets using embedding dimensions of 4, 6 and 4 respectively; Fernandez, (1999) applied the $v$-SV regression algorithm to the Santa Fe data set using embedding sizes of 15, 20 and 25.

As we can see, there is also no general rule for selecting the embedding dimension. For this study, we drew inspiration from the work of Tashman, (2000) by choosing an embedding dimension of five (5) in 1-day time steps and setting the forecast horizon to 1. This makes sense since there are 5 trading days in a week and we believe that stock prices recorded over a given week - say Monday to Friday - should be appropriate for making forecast for the next Monday. The justification for making 1-day ahead

forecasts is that financial time series are usually influenced by the interplay of several qualitative and quantitative factors both on the local and international fronts, which explains why such financial data are dynamic and volatile. Therefore using the input vector to make forecasts beyond the next time instance may be inappropriate as much could have happened between the most recent time period represented in the input object and the period into the future for which forecast is to be made. In other words, given the type of data we have, we seek to minimize the errors associated with our prediction models by limiting our forecast horizon.

### 4.5.2 Choice of Kernel, C, and $\varepsilon$

There are three user-defined parameters that are peculiar to SVR; these are the type of kernel function, the regularization constant $C$ and the maximum allowable devation $\varepsilon$. In Section 3.4, we discussed several methods for obtaining $C$ and $\varepsilon$. To obtain the optimal values of these two parameters, we followed the cross-validation procedure employed by Muller et al (1997). Although this method is computationally expensive, we chose to use it because other methods of obtaining the parameters are even more complex and involve defining many more parameters.

For the choice of kernel, the Radial Basis Function (RBF) was selected as it has been found to be superior to other kernel types as explained in Section 3.3.

## 4.6 Measures of Prediction Accuracy

There are several measures of accuracy that are commonly used to access the generalization/forecasting ability of forecasting models. However, there are certain

metrics that are peculiar to financial time series forecasting. Abecasis et al (1999) presented an excellent article that summarized the state of the art performance metrics for financial time series forecasting. Most of these metrics measure the prediction accuracy that is often defined in terms of the difference between the actual and predicted values while others measure the ability of the models to correctly predict the trend/direction in the series. It is not in all cases that prediction systems are constructed for the purpose of obtaining forecast of the next element in the series that are closest to the actual, in some cases, what is required is a prediction of the movement of the series.

In reality, forecasting models are developed to achieve varying objectives, thus different metrics are required to ascertain that those objectives are accomplished. Each of the performance metrics has its advantages and limitations; hence, it is often desirable to use multiple performance measures rather than a single measure for a particular forecasting problem.

As mentioned earlier, there are several performance measures that are commonly used to assess the predictive ability of forecasting models in the financial world. In our study, five (5) different performance metrics namely Directional Symmetry, Mean Squared Error, Mean Absolute Percentage Error, Downside Risk, and Upside Risk were considered. These metrics are defined in the next section and their calculations are presented in Table 4.1.

It is worthy to note that the performance metrics used in this thesis are not exhaustive; there are many more performance measures that have been used for financial forecasting.

Table 4.1: Performance Metrics and their Calculations

| **Performance Metrics** | | |
|---|---|---|
| **Metric** | **Acronym** | **Calculation** |
| Directional Symmetry | DS | $DS = \frac{100}{n}\sum_{i=1}^{n} d_i,$ <br> $where\ d_i = \begin{cases} 1\ if\ (a_i - a_{i-1})(p_i - p_{i-1}) > o \\ o\ otherwise \end{cases}$ <br> $n = number\ of\ samples$ <br> $a_i = actual\ price\ at\ period\ i$ <br> $p_i = predicted\ price\ at\ period\ i$ |
| Upside Risk | UPR | $UPR = \frac{1}{n}\sum_{i=1,\ a_i > p_i}^{n} (a_i - p_i)$ |
| Downside Risk | DSR | $DSR = \frac{1}{n}\sum_{i=1,\ a_i < p_i}^{n} (p_i - a_i)$ |
| Mean Squared Error | MSE | $MSE = \frac{1}{n}\sum_{i=1}^{n} (a_i - p_i)^2$ |
| Mean Absolute Percentage Error | MAPE | $MAPE = \frac{1}{n}\sum_{i=1}^{n} \lvert PE_i \rvert$ <br> $where\ PE_i = Percentage\ \text{Error}\ for\ period\ i$ |

For instance, the Normalized Mean Square Error (NMSE) had been previously used by Dorsey and Randall, (1998), Tay and Cao, (2001, 2002), Cao and Gu, (2002), Cao and Tay, (2003), and Cao et al, (2003). Other accuracy measures include the Mean Absolute Error (MAE) employed by Tay and Cao, (2001), and Cao and Tay, (2003); and the Root Mean Square Error (RMSE) used by Cao and Gu (2002), Muller et al, (1999), Fernandez (1999), Pawelzil et al, (1996), Thomason and Caldwell, (1998), and Thissen et al, (2003).

### 4.6.1 Definitions of Performance Metrics Used

### Directional Symmetry (DS)

The directional symmetry between actual and predicted values expressed as a percentage, was originally defined by Azoff in 1994, and has since been employed in a number of financial studies (Abecasis et al, 1999). By definition, directional symmetry is the percent of forecasts for which the movement of the forecast is the same as the movement of the target variable (Dorsey and Randall, 1998). The value of the directional symmetry provides an indication of the correctness of the predicted direction. Therefore, DS = 50% implies that the predicted direction was correct for half of all predictions hence a larger value suggests a better predictor. An excellent characteristic of the DS is that it is independent of the magnitude of the returns (Thomason and Caldwell, 1998). This metric had been previously used by Dorsey and Randall, (1998); Thomason and Caldwell, (1998); Tay and Cao, (2001); Cao and Tay, (2003); and Kim (2003).

## Upside Risk (USR) and Downside Risks (DSR)

These accuracy metrics are similar to the directional symmetry. The upside risk measures the risk associated with under-forecasting actual values. Mathematically, it is the average of the differences between the predicted values and the corresponding actual values whenever the predicted value is less than the actual value. The downside risk on the other hand, measures the risk associated with over-forecasting the actual values. It is represented as the average of the differences between the actual values and the predicted values where the predicted value is greater than the actual value. These measures were used by Dorsey and Randall, (1998), Yang et al, (2002a, 2002b).

## Mean Squared Error (MSE)

Mean squared error is one of the performance measures that has been used extensively for model evaluation purposes (Poon and Granger, 2003). It is often used in place of the Mean Error (ME), because unlike the ME, the tendency of positive and negative errors to offset each other is prevented. The MSE is simple to calculate and is a familiar metric in the forecasting arena. In the financial world, the MSE has been employed by Cherkassky and Ma (2004), Thomason and Caldwell, (1998), Lee and Billings, (2002) just to mention a few.

## Mean Absolute Percentage Error (MAPE)

For comparisons across different time series, it is expedient to use percentage error measurements. The MAPE falls into this category and is often used in accessing the forecasting ability of most time series prediction systems. Similar to the DS, it is a

relative measure of performance whose magnitude does not depend on the scale of the data. To measure the forecasting ability of their models, MAPE was employed by Van Eyden, (1997) and Stickel, (1990).

In this study, our major concern is to be able to develop a model that correctly predicts the direction of the movement of the series hence the directional symmetry serves as our major performance measure. However, we also desire a model that produces forecasts that are as close to the actual price measurements as possible necessitating our use of the other performance measures.

Figure 4.6 shows the concept behind each of the prediction systems developed for this study along with the performance measures used.

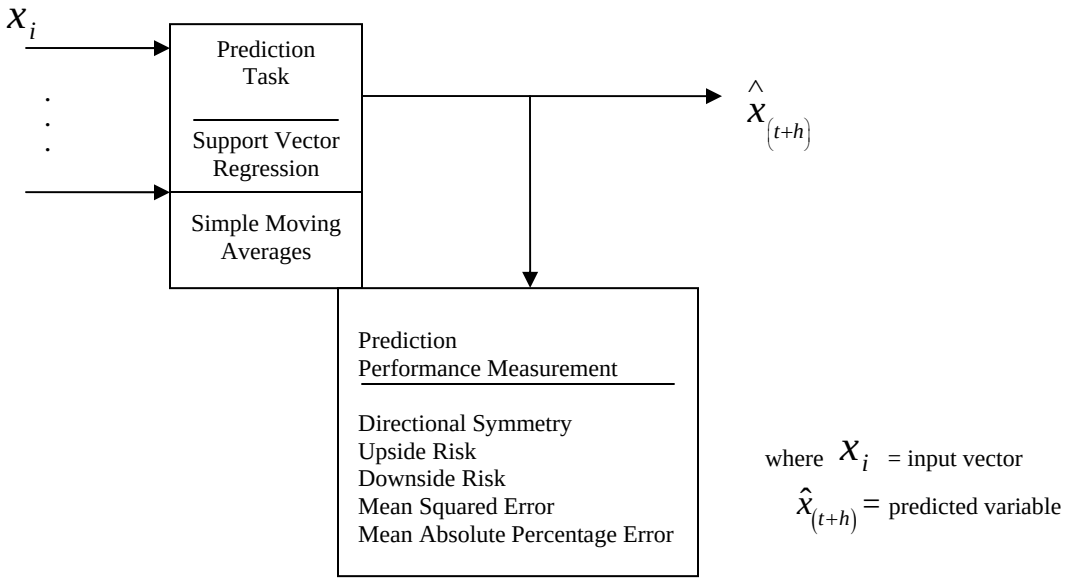



Figure 4.6: General Prediction Concept

## 4.7 Software Used

The MATLAB toolbox developed and first presented by Gunn (1998) was used for the SVR calculations. Separate codes were written for the SMA calculations using the standard MATLAB software.

## 4.8 Methodology and Experimental Results

In this section, I present the methodology used in developing the prediction models and discuss the results obtained.

### 4.8.1 Methodology using Support Vector Regression

As mentioned earlier, three different transformations namely the Relative Difference in Percentage (RDP), Z-score Normalization, and the Natural Logarithm were applied to each of the five series of stock prices. The application of these transformations was performed prior to conducting the SVR algorithm.

The cross-validation procedure was employed to determine the values of $C$ and $\varepsilon$ that works best for each of the five time series data. To obtain the optimal values of $C$, the cross-validation task was performed by varying the values of $C$ from 5 to 100 in steps of 5. According to Tay and Cao (2001), the appropriate range of $C$ is between 10 and 100. However, in order to have a more thorough analysis, we extended the lower limit of the range for $C$ to 5. The value of $\varepsilon$ was varied from 0.001 to 0.1 in steps of 0.005. We searched the literatures for suggestions on appropriate range of $\varepsilon$ but could not find any. However, we found some previous studies where small values of the margin were used

(Vapnik et al, 1997 and Chih-Chung and Chih-Jen, 2001). The kernel used was the Radial Basis Function. The benefits of using this kernel type were already discussed in Section 3.3.

The methodology used in this thesis was implemented using the following steps. The first step is to set the value of $\varepsilon$ to 0.01 while we varied the value of *C* for each training set. The value of *C* that produced the maximum Directional Symmetry (DS) value was selected as the best value for each of the training set. The next step is to set the value of *C* as obtained in step one and vary the value of $\varepsilon$. Again, the best of $\varepsilon$ is the value that gives the maximum DS value. When there is more than one value of *C* or $\varepsilon$ with the maximum DS value, the selection of the best *C* or $\varepsilon$ is based on other performance measures such as Mean Squared Error (MSE) and Mean Absolute Percentage Error (MAPE). This methodology used is depicted in Figure 4.7.

As we can deduce from Figure 4.7, two independent replications of SVR modeling was carried out for each of the three data transformations and for each of the five series of stock prices, resulting in a grand total of 30 experiments. With an embedding of five, the training data set for each SVR replication generated 360 data patterns while the test data set generated 115 data patterns. For each of the 30 experiments, every single input data pattern of 5 lagged data points has a corresponding target to be predicted.

### 4.8.2 Experimental Results of Support Vector Regression

The several experimental results obtained using SVR are discussed in this section.

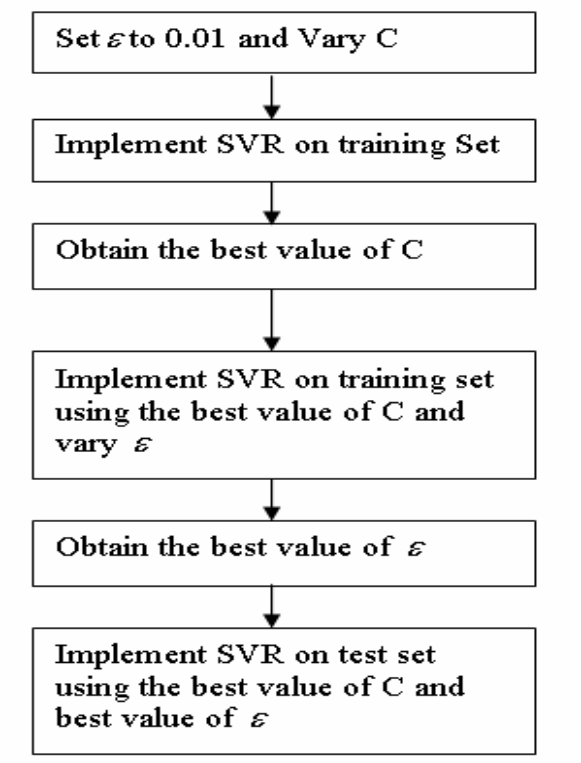



Figure 4.7: Steps used in the SVR Experimentation

Table 4.2: Results of SVR using RDP Transformation

| Transformation: RDP<br>Range of C: 5:100 in steps of 5<br>Range of $\varepsilon$: 0.001:0.1 in steps of 0.005 | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| **Name of Stock** | SVR Parameters and Number of Support Vectors | | | | Performance Metrics | | | | |
| | **Opt C** | **Opt $\varepsilon$** | **NSV** | **%NSV** | **DS** | **DSR** | **UPR** | **MSE** | **MAPE** |
| American Airlines | 5 | 0.001 | 360 | 100.0 | 47.826 | 0.431 | 0.357 | 0.964 | 2.399 |
| McDonalds | 5 | 0.076 | 356 | 98.8 | 53.913 | 0.415 | 0.325 | 0.984 | 2.101 |
| Sears | 5 | 0.096 | 353 | 98.1 | 52.174 | 0.426 | 0.465 | 1.581 | 2.673 |
| Toys 'R Us | 5 | 0.096 | 346 | 96.1 | 56.522 | 0.133 | 0.164 | 0.153 | 2.190 |
| Microsoft | 5 | 0.096 | 350 | 97.2 | 50 | 0.666 | 0.510 | 2.835 | 2.776 |

#### 4.8.2.1 Using RDP Transformation

Table 4.2 shows the summary of the results obtained when the SVR technique was applied to stock prices that have been converted to relative differences in percentage.

Based on the results displayed in the Table, it can be seen that for RDP, the value of *C* remained constant at 5 irrespective of the training set used; however, the value of $\varepsilon$ varied from 0.001 and 0.096 with 0.096 occurring as the optimal $\varepsilon$ value for three (3) of the five stock prices. The minimum number of support vectors used by the SVR algorithm was 346 for Toys 'R us and the maximum was 360 (100%) for American Airlines.

The DS values range from 47.826% to 56. 522%, the DSR ranges from 0.133 to 0.666, the UPR ranges from 0.164 to 0.510, and the MSE and MAPE range from 0.153 to 2.835 and 2.101 to 2.776 respectively. The highest and best DS of 56.522% was seen for Toys 'R us and the smallest was seen for American Airlines. The minimum MSE, DSR and UPR were also achieved with the Toys 'R us stock price. However, the smallest MAPE was achieved with McDonalds. The maximum MAPE, DSR, UPR, MSE was seen for Microsoft. It should be noted that the smaller the values of the DSR, UPR, MSE and MAPE, the better the prediction system.

Figures 4.8 to 4.17 show how the performance measures and the number of support vectors change as the values of $C$ and $\varepsilon$ were varied during the first and second replications of the experiments using the RDP transformation.

#### 4.8.2.2 Using Z-score Transformation

Table 4.3 shows the summary of the results obtained when the SVR model was applied to stock prices that have been pre-processed using the Z-score transformation. Based on the results displayed in the Table, it can be seen that for Z-score, the value of $C$ varies randomly from 5 to 60 while the value of $\varepsilon$ varies from 0.036 and 0.096, with 0.096 occurring as the optimal $\varepsilon$ value for two (2) of the five stock prices. The minimum number of support vectors used by the SVR algorithm was 338 (93.9%) for McDonalds and the maximum was 352 (97.8%) for American Airlines.

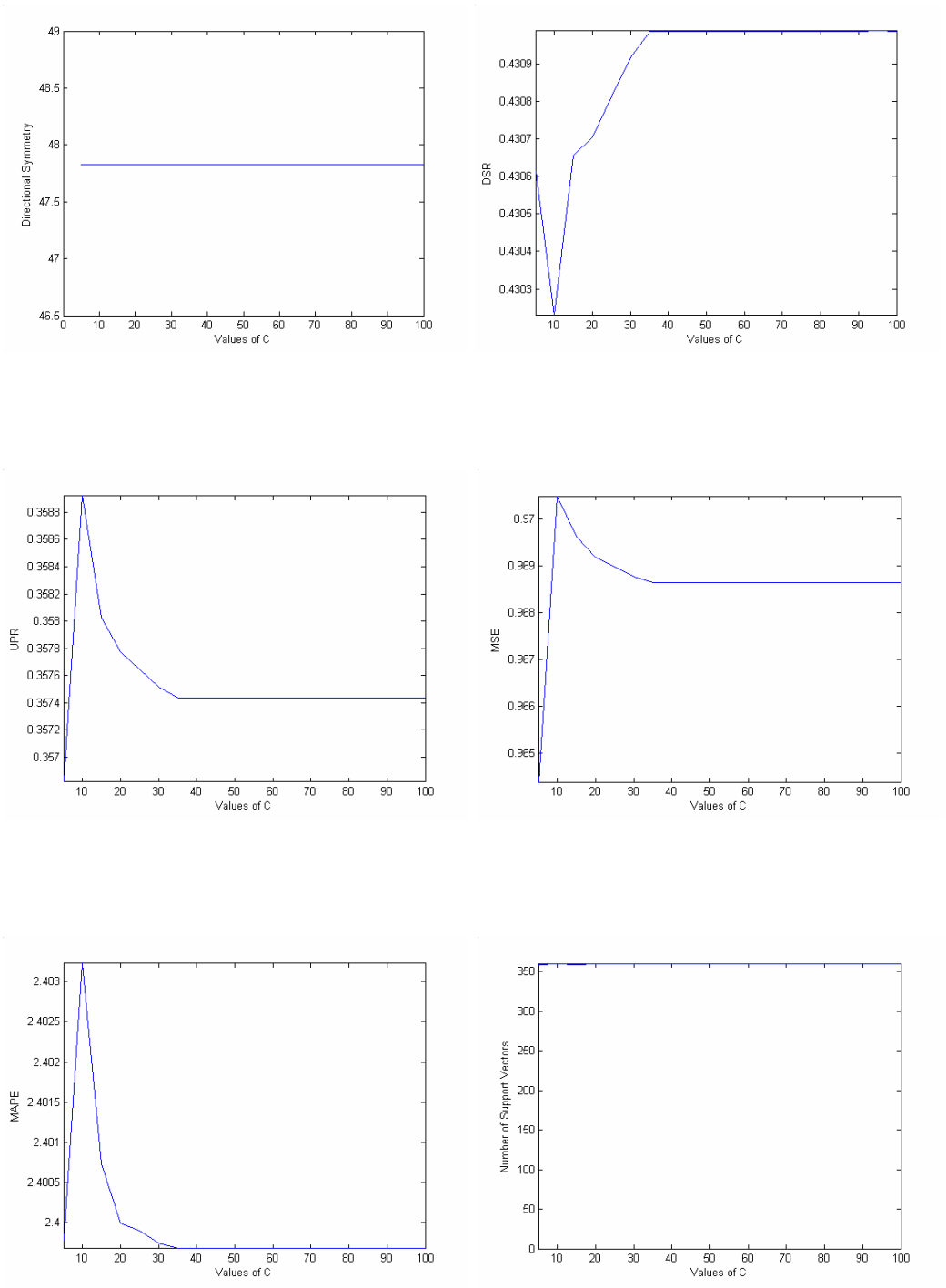



Figure 4.8: American Airlines - First Replication using RDP

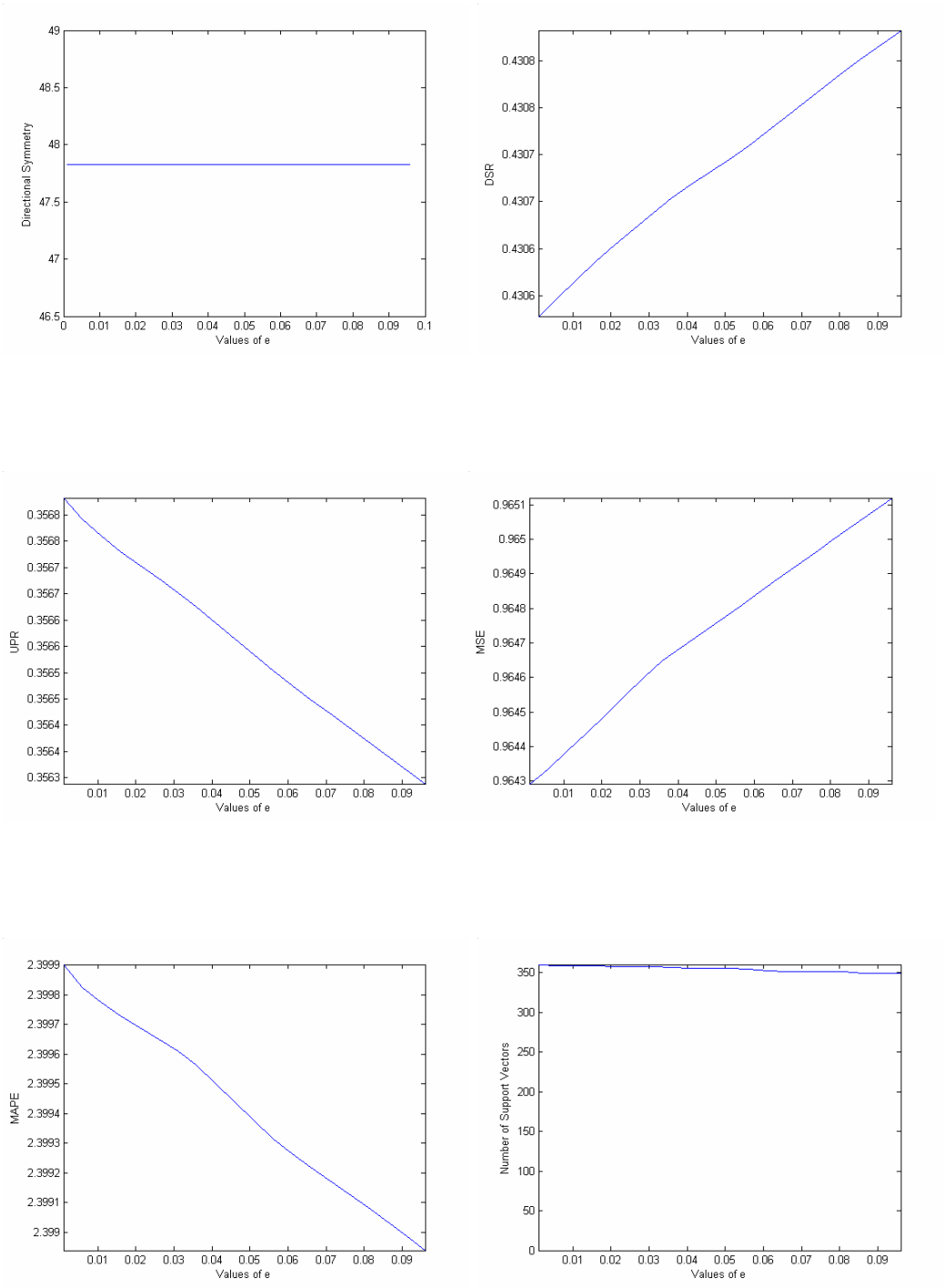



Figure 4.9: American Airlines – Second Replication using RDP

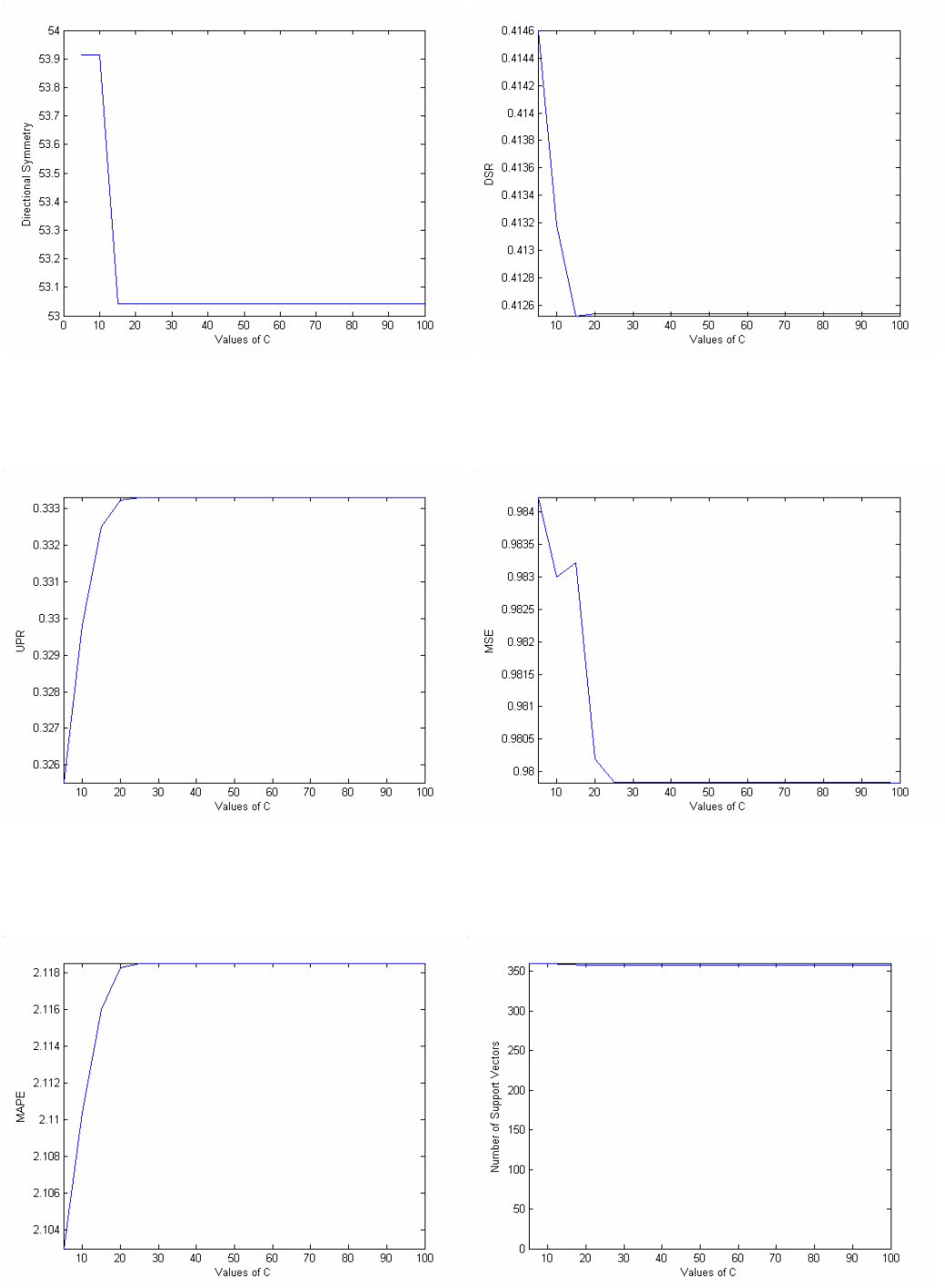



Figure 4.10: McDonalds – First Replication using RDP

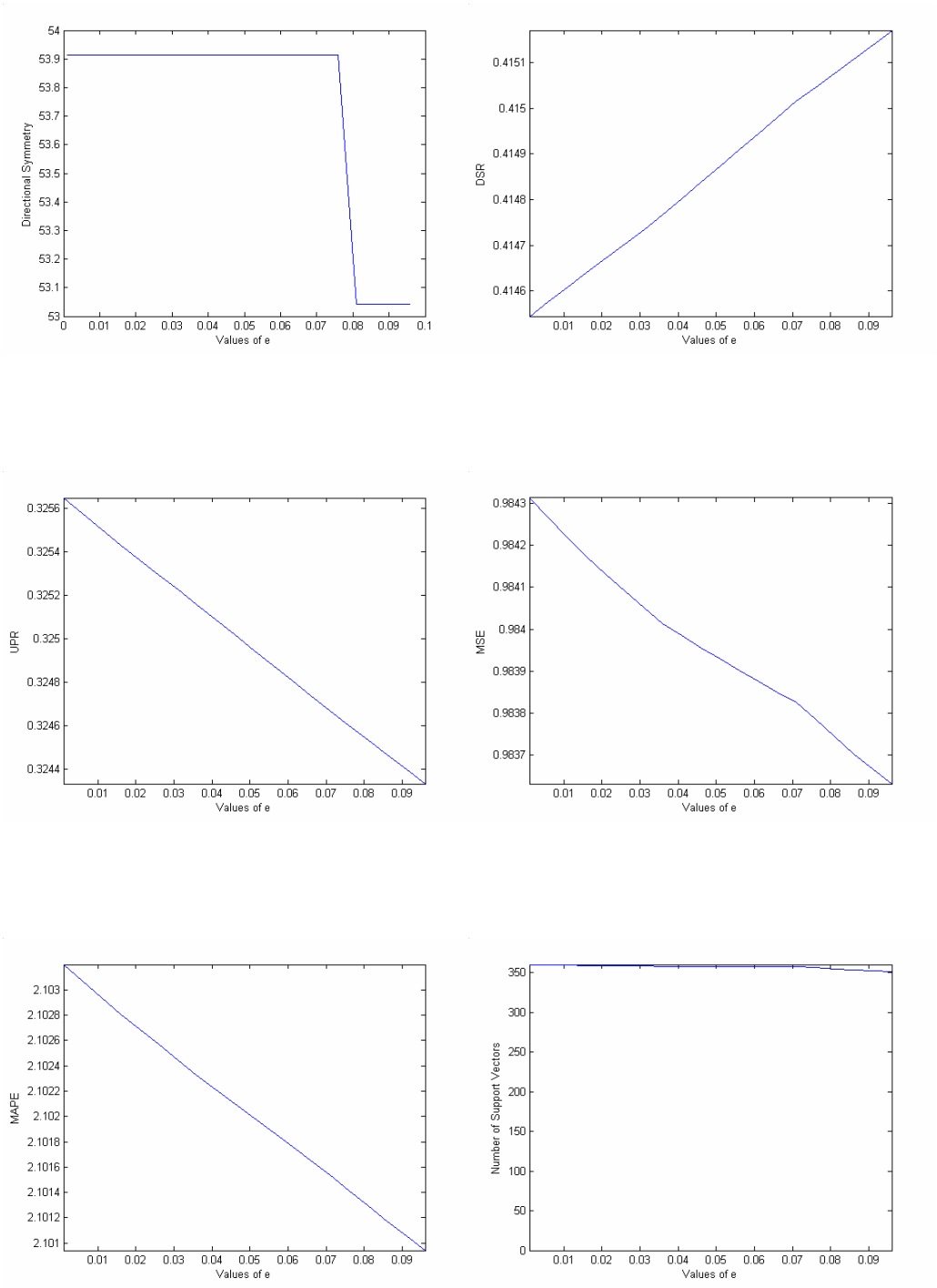



Figure 4.11: McDonalds – Second Replication using RDP

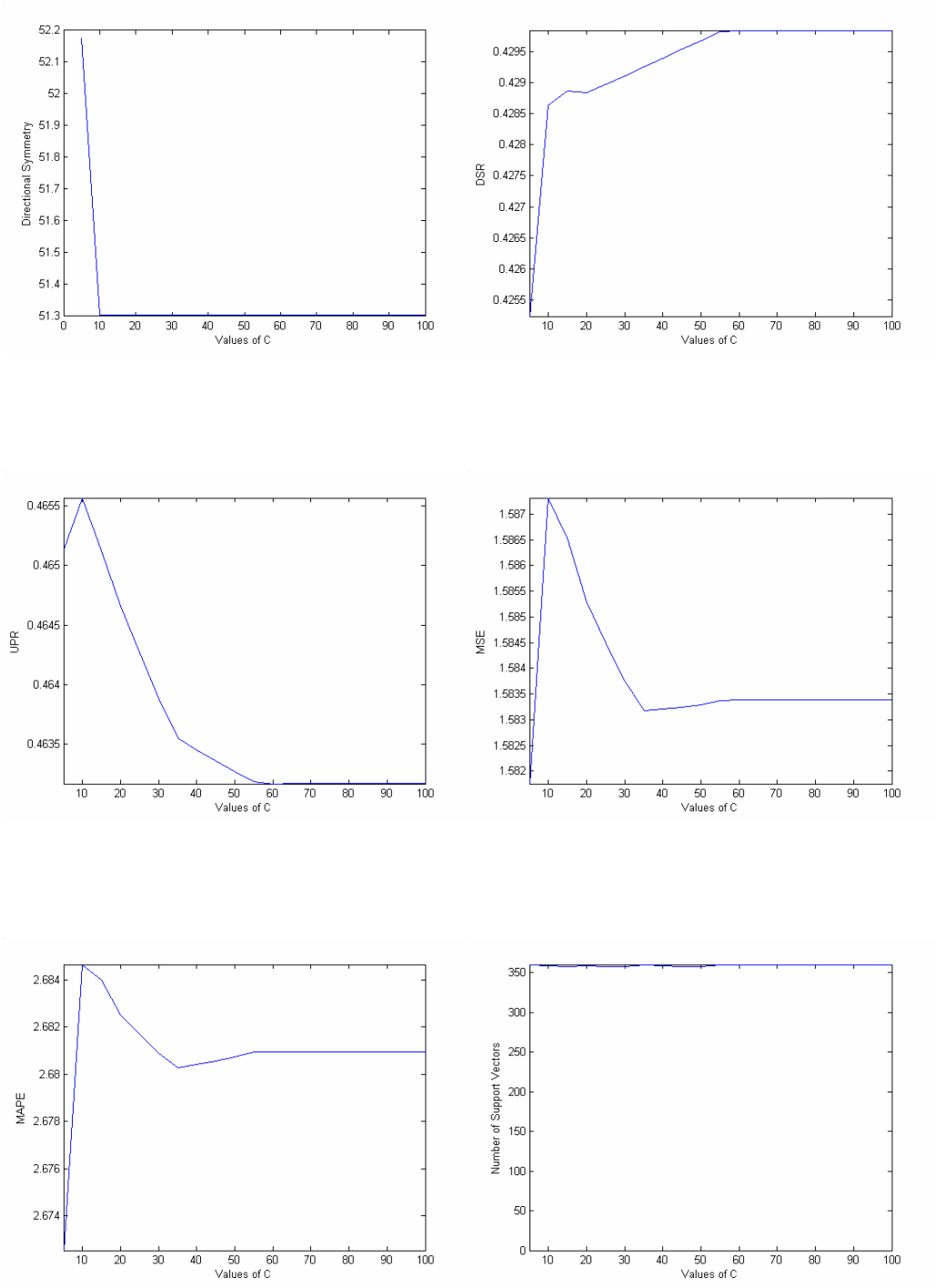



Figure 4.12: Sears – First Replication using RDP

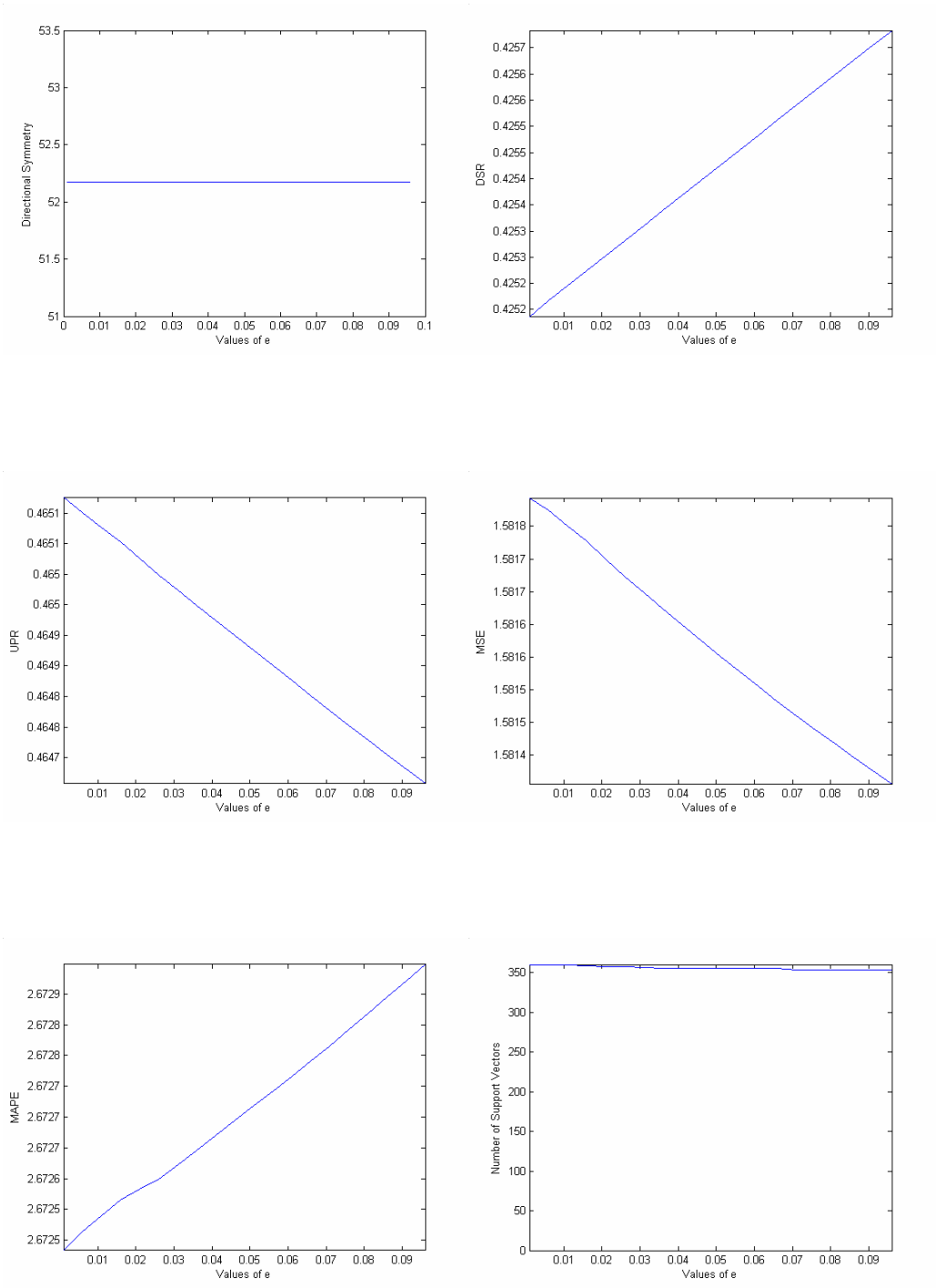



Figure 4.13: Sears – Second Replication using RDP

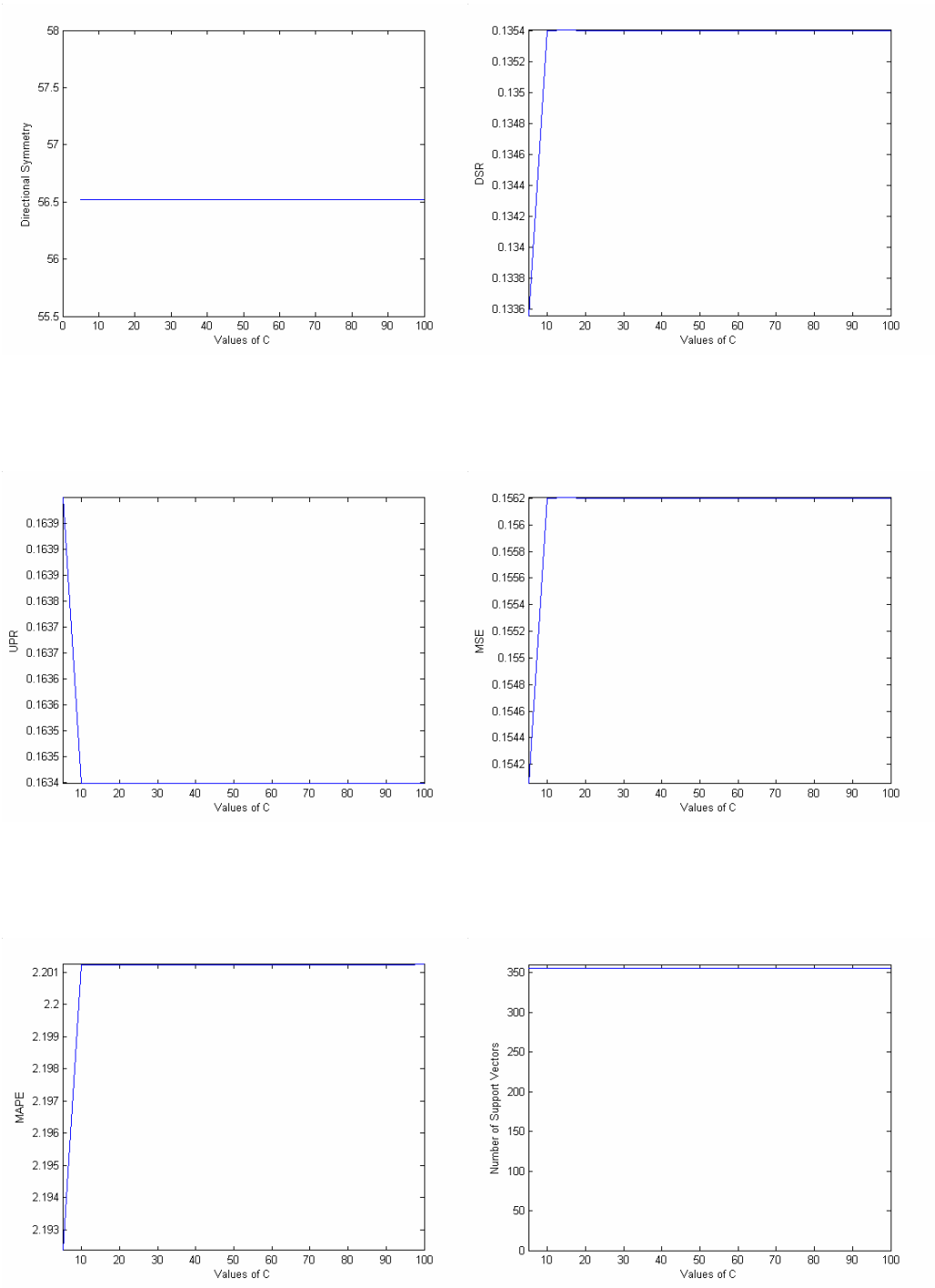



Figure 4.14: Toys ‘R us – First Replication using RDP

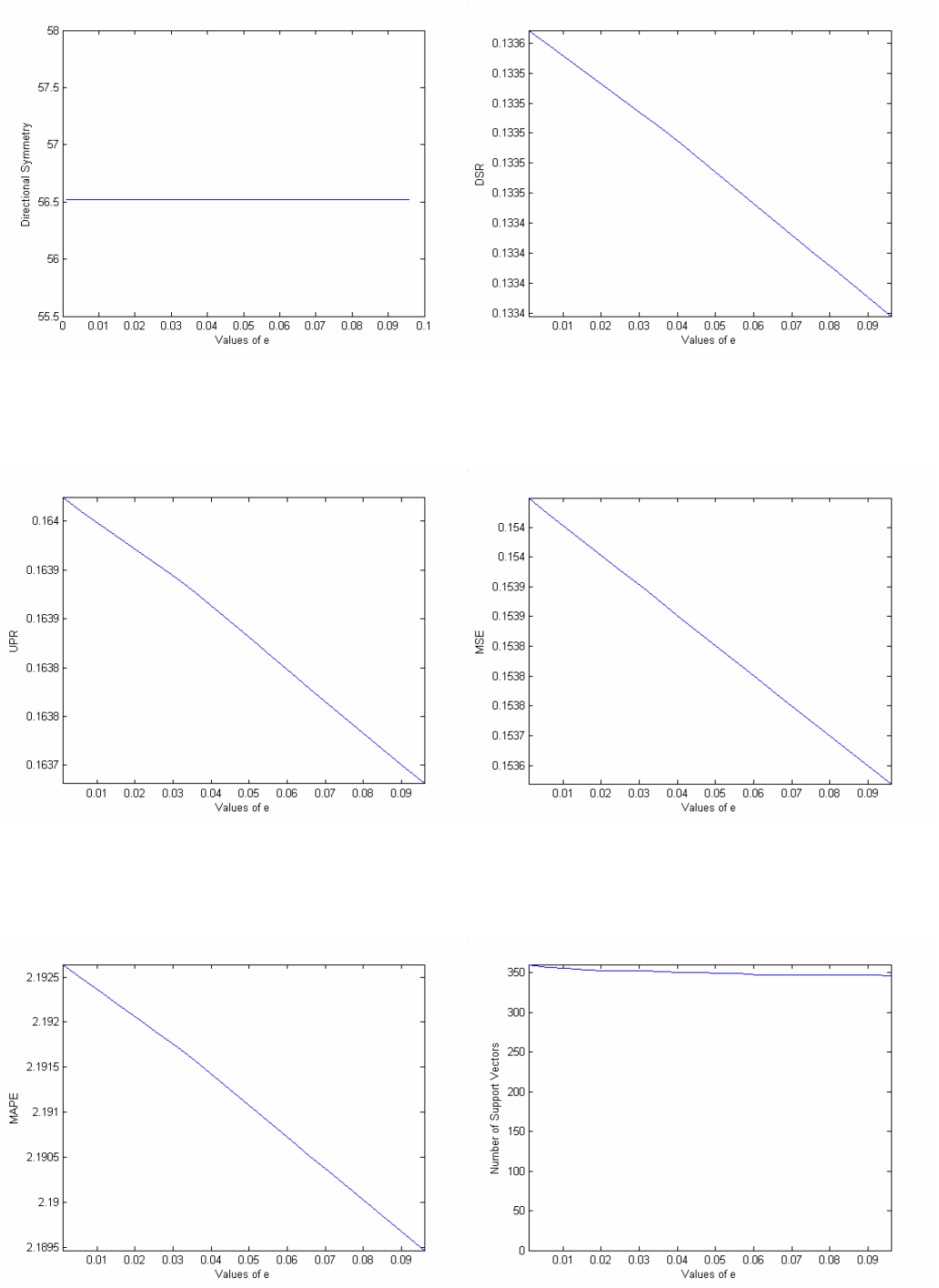



Figure 4.15: Toys ‘R us – Second Replication using RDP

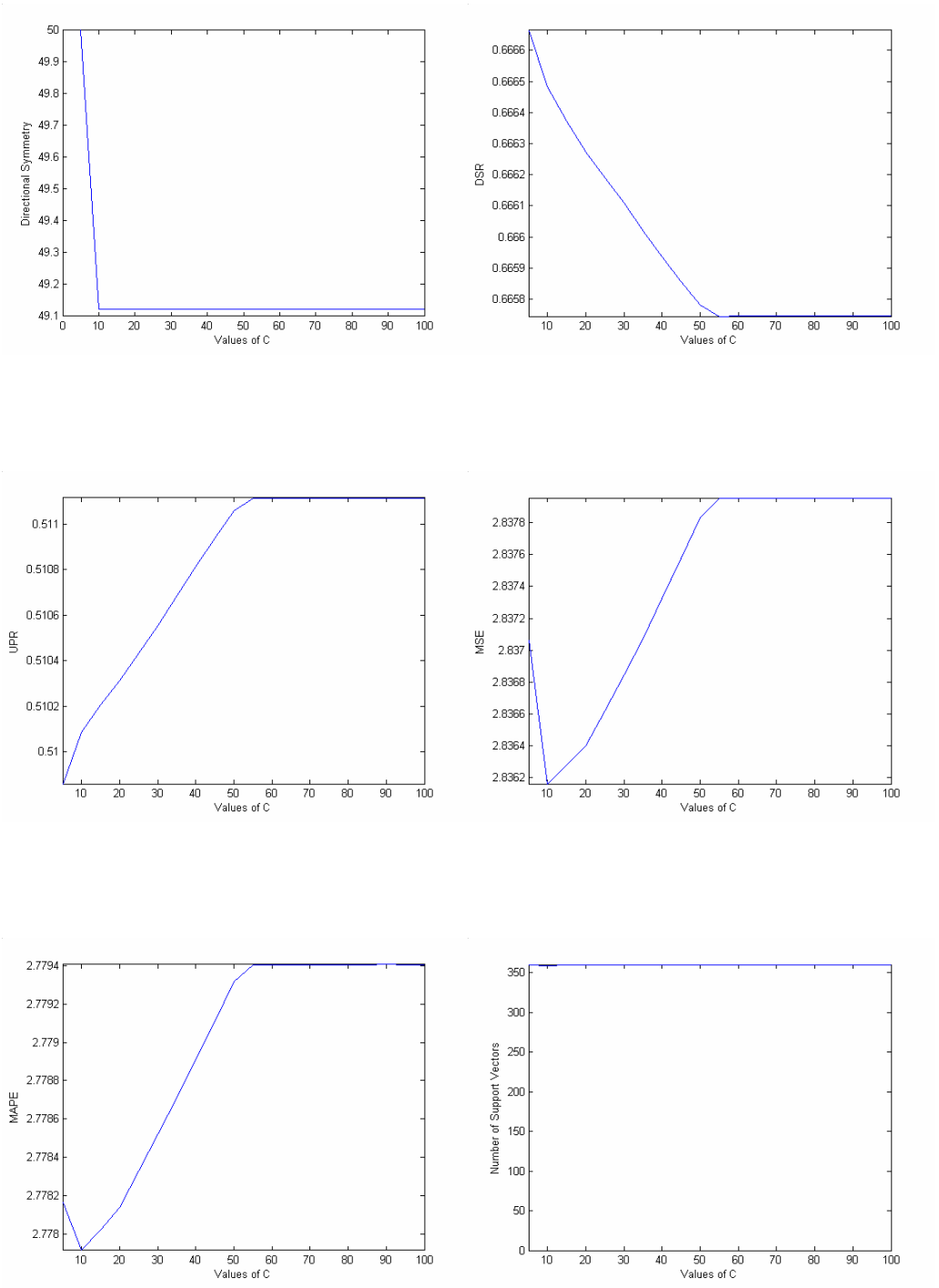



Figure 4.16: Microsoft – First Replication using RDP

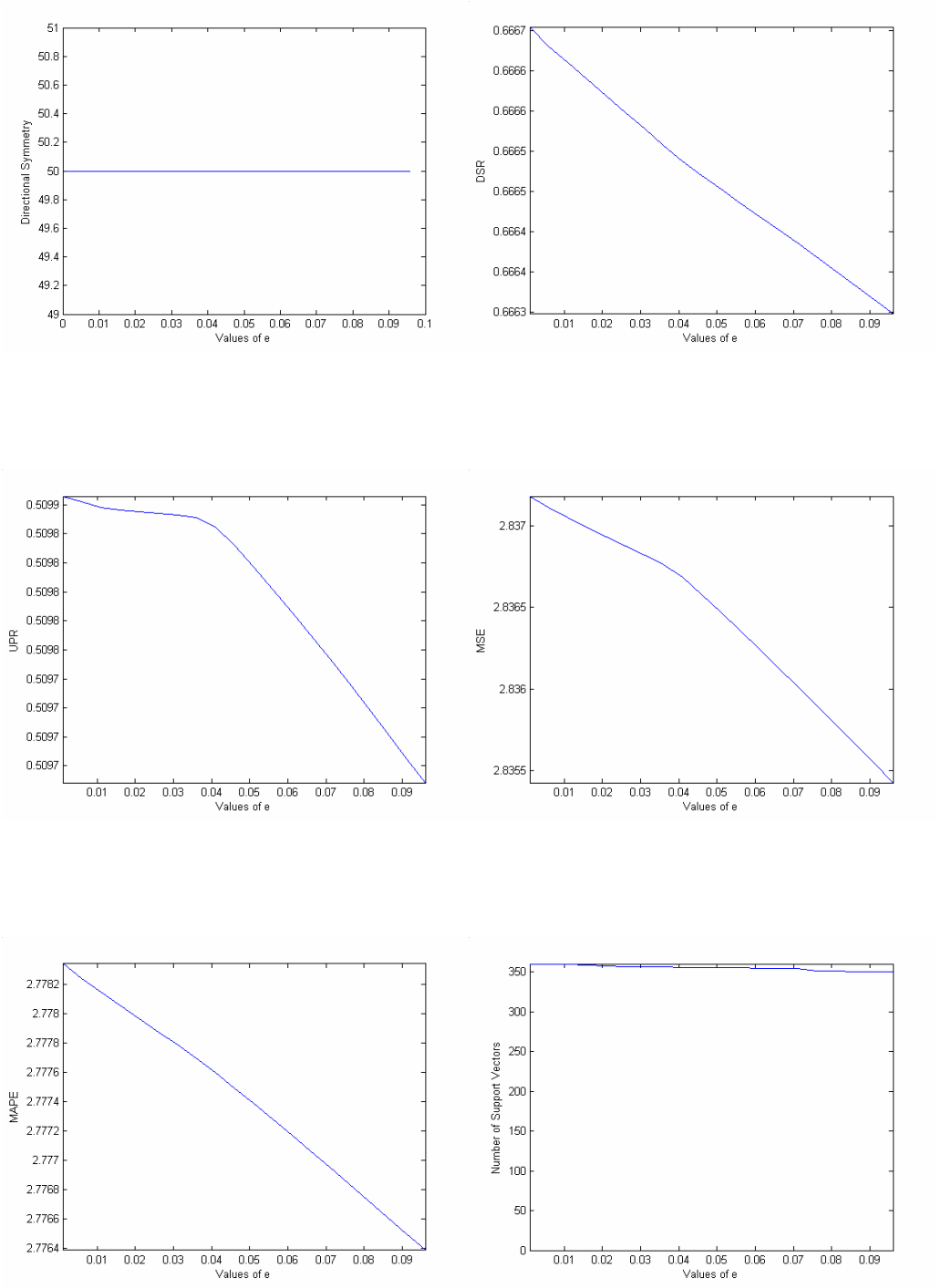



Figure 4.17: Microsoft – Second Replication using RDP

Table 4.3: Results of SVR using Z-score Transformation

| Transformation : Z-score<br>Range of C: 5:100 in steps of 5<br>Range of $\varepsilon$ : 0.001:0.1 in steps of 0.005 | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| **Name of Stock** | SVR Parameters and Number of Support Vectors | | | | Performance Metrics | | | | |
| | **Opt C** | **Opt $\varepsilon$** | **NSV** | **%NSV** | **DS** | **DSR** | **UPR** | **MSE** | **MAPE** |
| American Airlines | 45 | 0.036 | 352 | 97.8 | 74.783 | 0.263 | 0.233 | 0.444 | 1.499 |
| McDonalds | 60 | 0.096 | 338 | 93.9 | 73.043 | 0.331 | 0.221 | 0.649 | 1.563 |
| Sears | 35 | 0.096 | 339 | 94.2 | 76.316 | 0.338 | 0.406 | 1.472 | 2.243 |
| Toys 'R us | 5 | 0.041 | 346 | 96.1 | 88.696 | 0.047 | 0.077 | 0.025 | 0.909 |
| Microsoft | 10 | 0.066 | 342 | 95.0 | 77.193 | 0.524 | 0.314 | 1.806 | 1.981 |

The values of the DS ranges from 73.043% to 88.696% while the DSR ranges from 0.047 to 0.524. The UPR ranges from 0.077 to 0.406, and the MSE and MAPE range from 0.025 to 1.806 and 0.909 to 2.243 respectively. Toys 'R us produced the highest and best DS value of 88.696%; the smallest was obtained for McDonalds. The minimum DSR, UPR, MSE, and MAPE were also achieved with the Toys 'R us stock price. The maximum DSR, UPR, and MSE were obtained for Microsoft while the maximum MAPE was obtained for Sears.

Figures 4.18 to 4.27 show how the performance measures and the number of support vectors change as the values of $C$ and $\varepsilon$ were varied during the first and second replications of the experiments using the Z-score transformation.

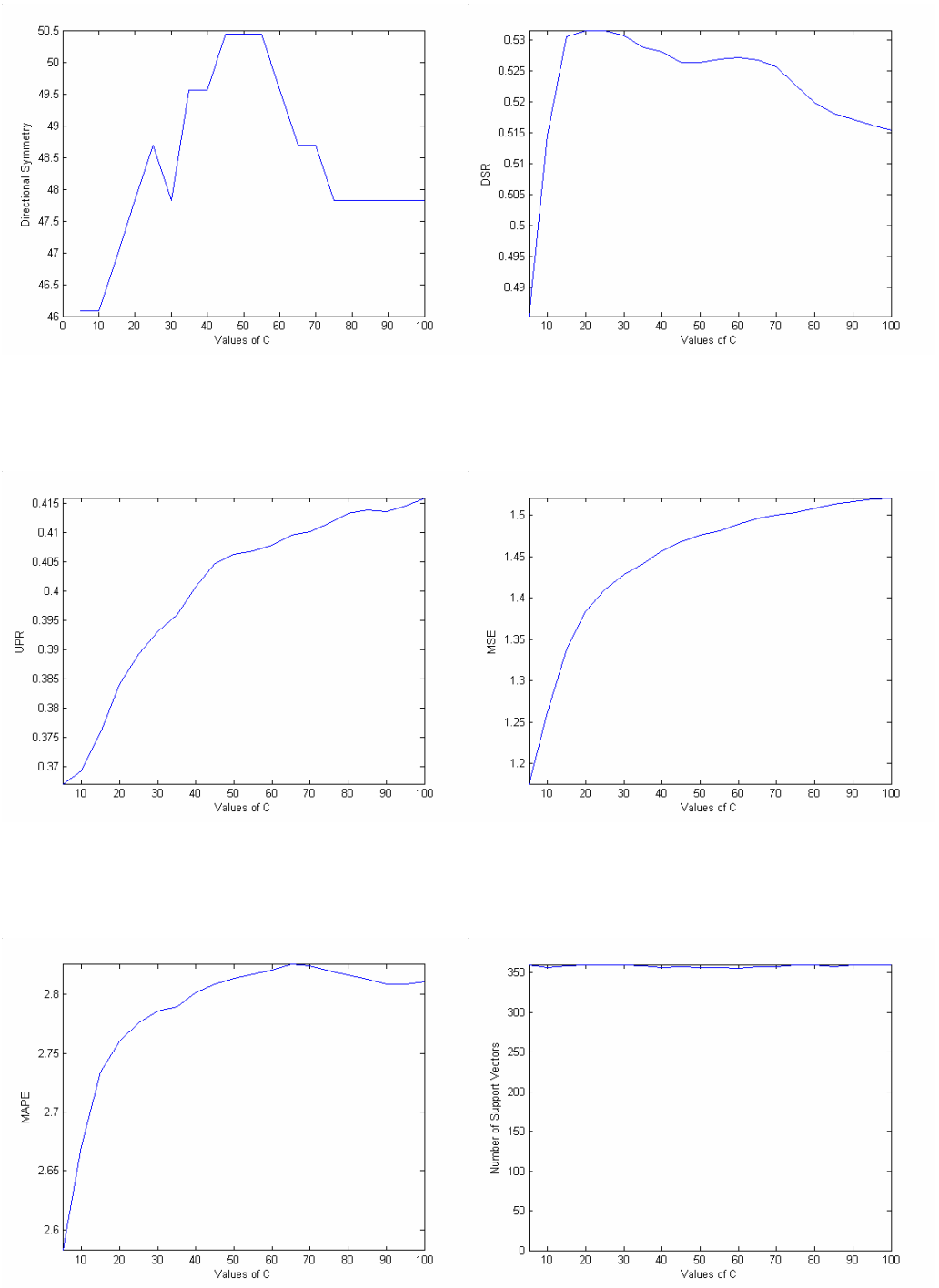



Figure 4.18: American Airlines – First Replication using Z-score

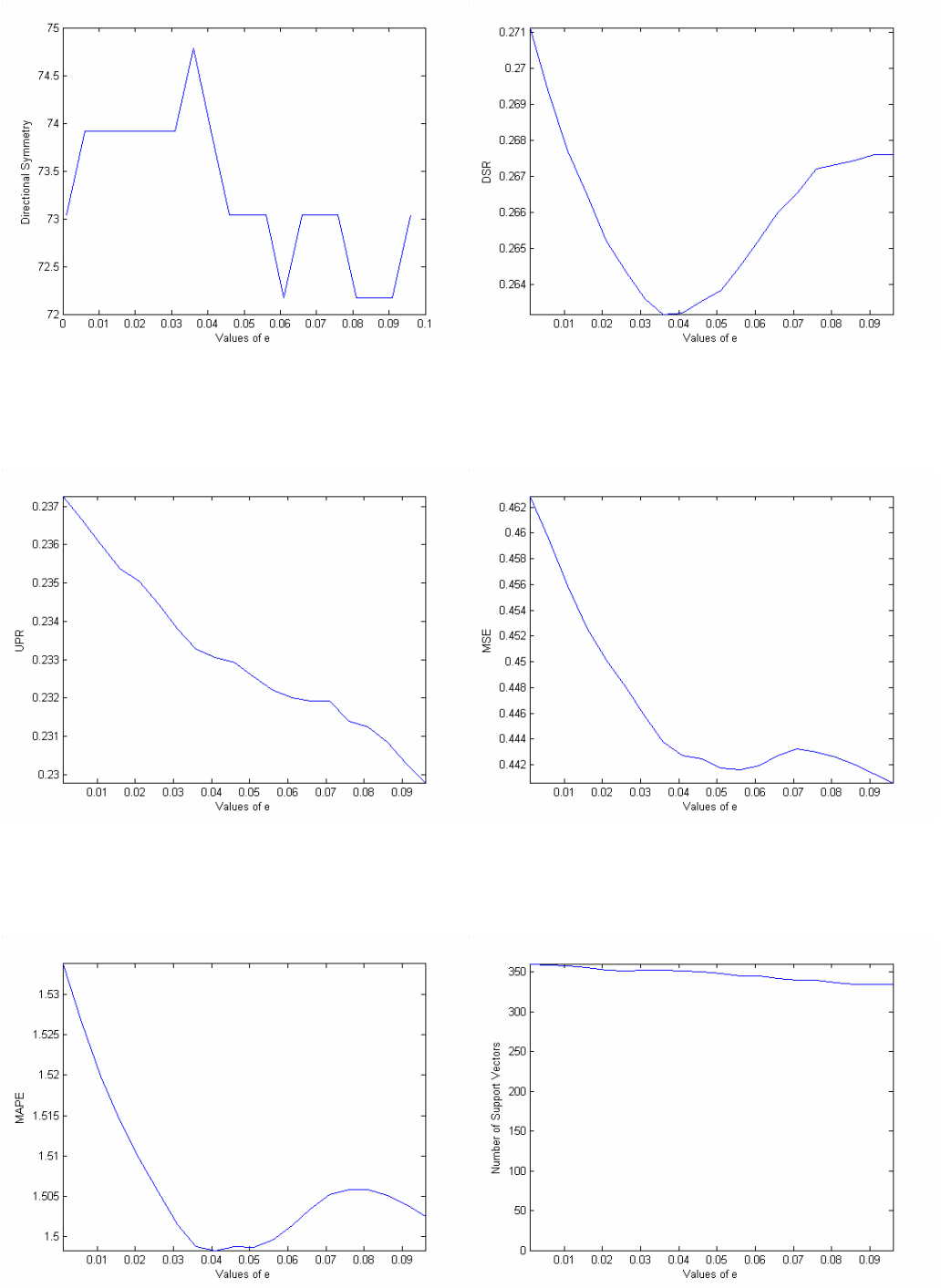



Figure 4.19: American Airlines – Second Replication using Z-score

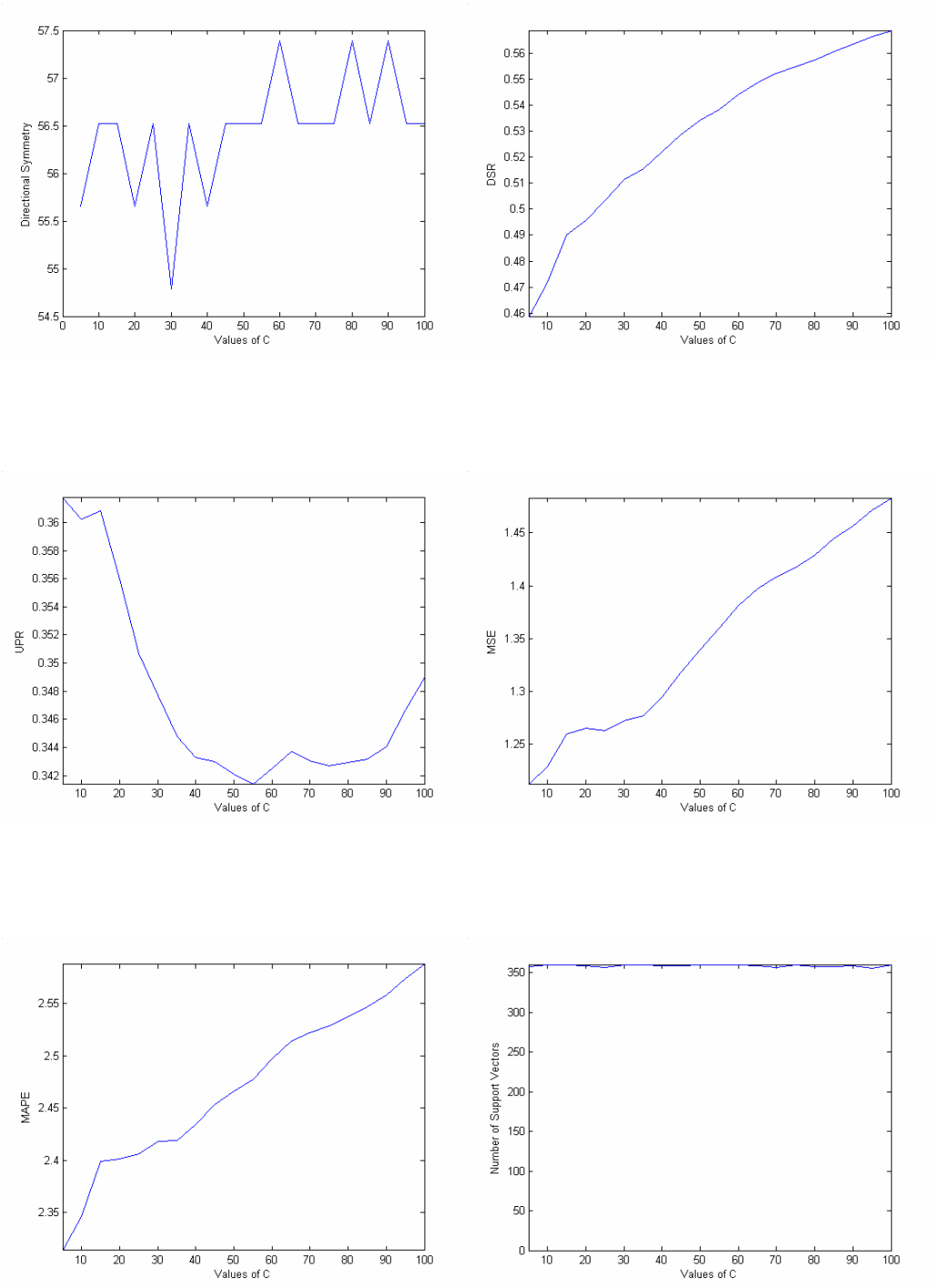



Figure 4.20: McDonalds – First Replication using Z-score

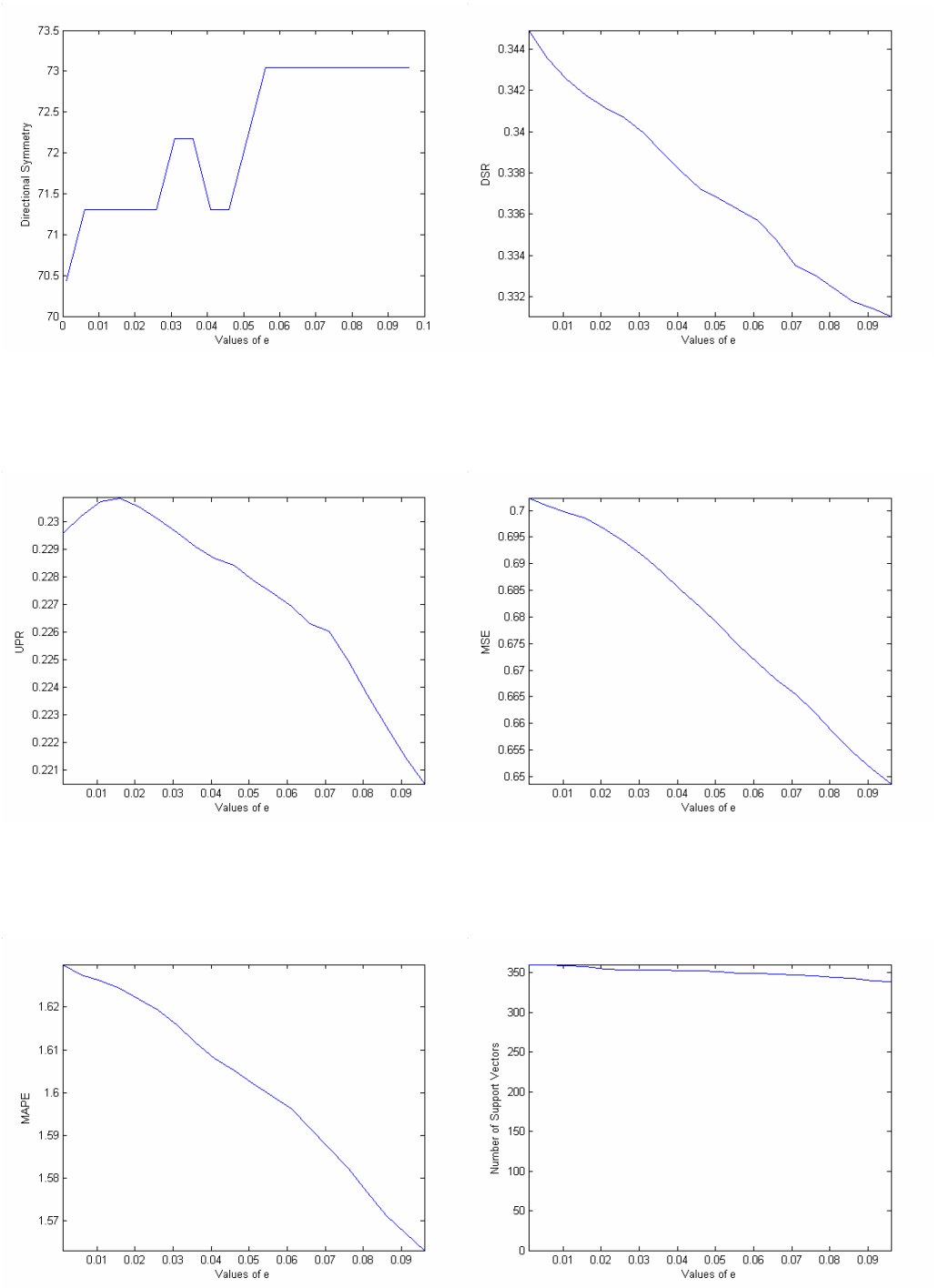



Figure 4.21: McDonalds – Second Replication using Z-score

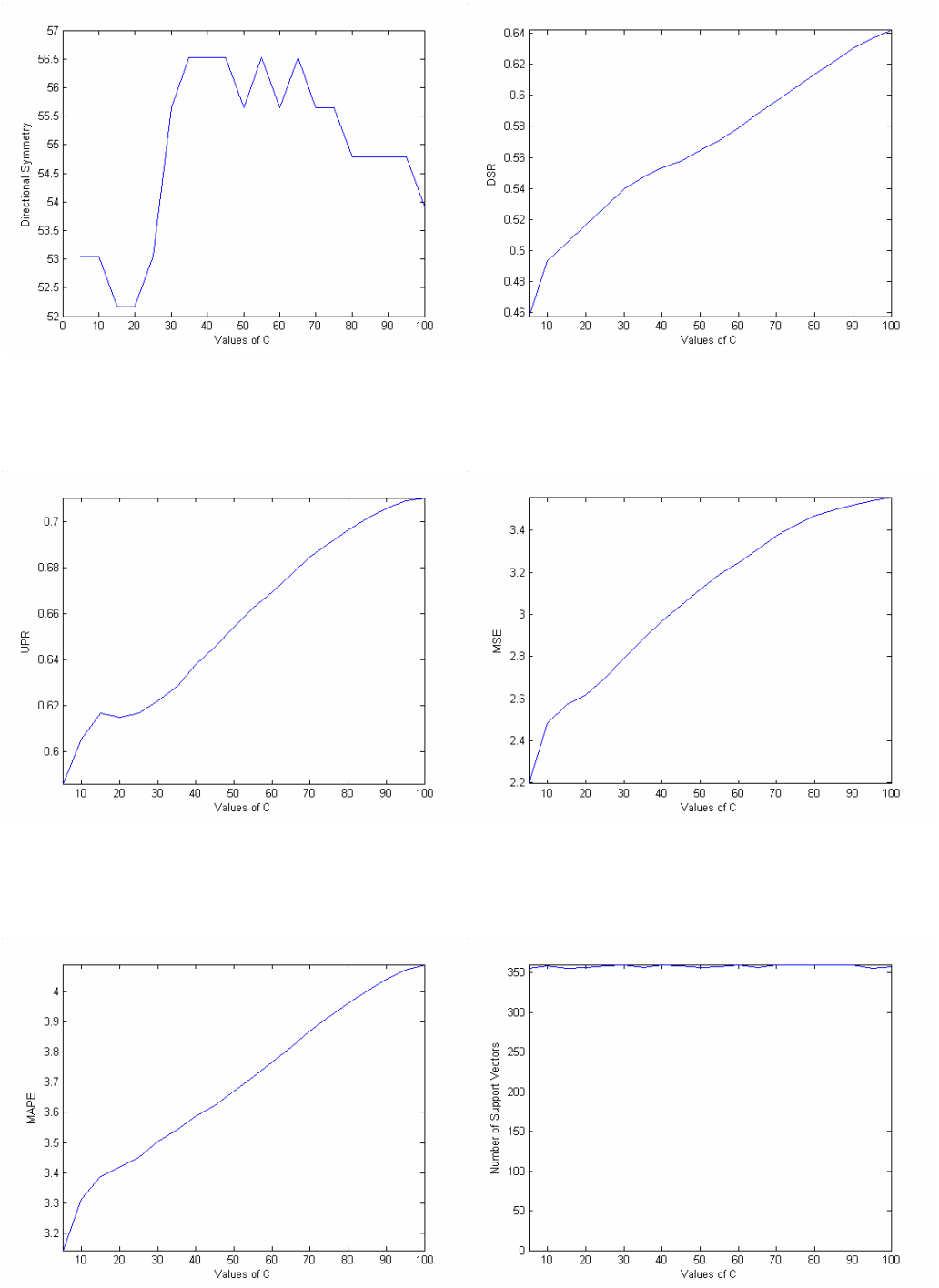



Figure 4.22: Sears – First Replication using Z-score

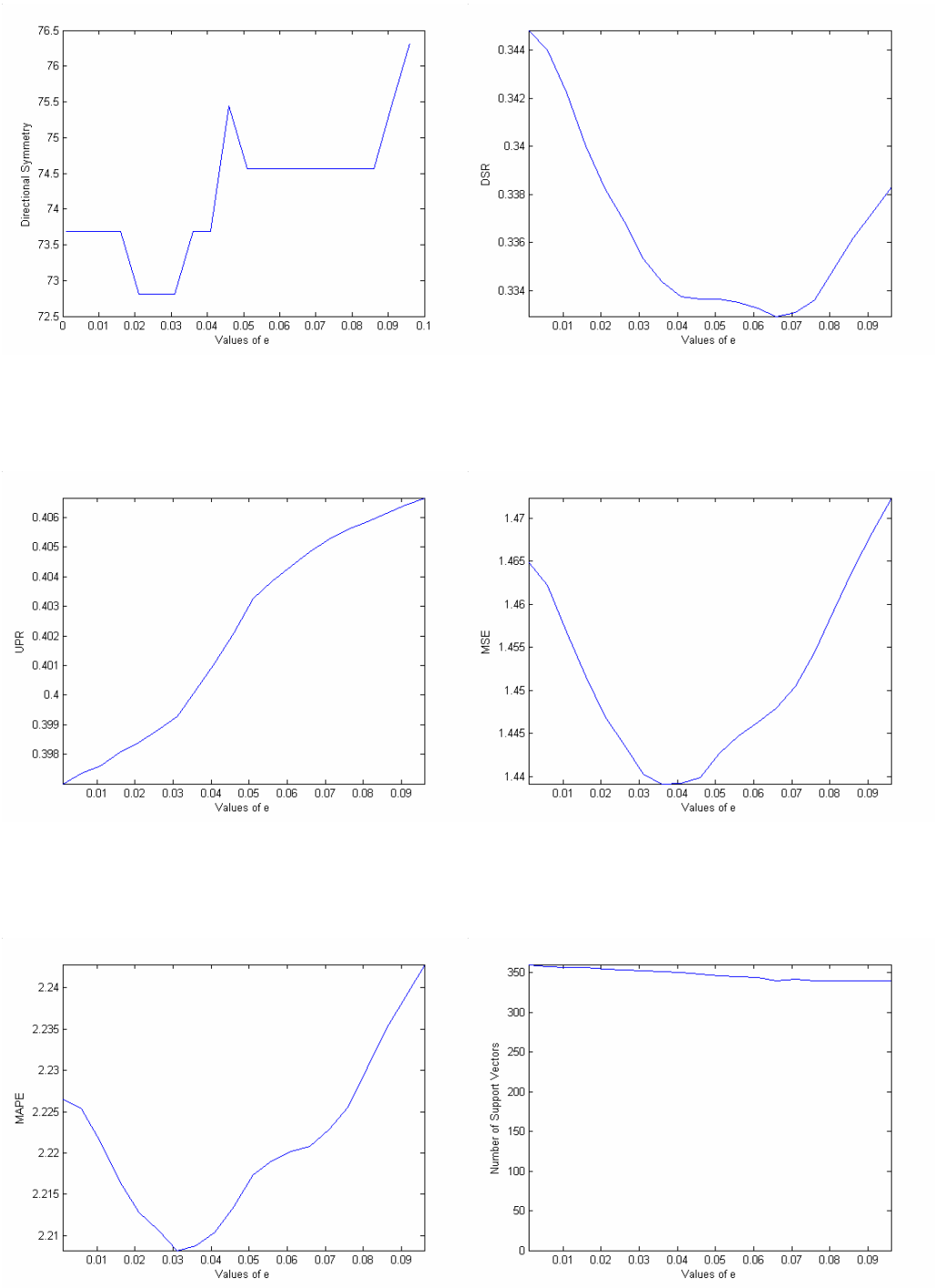



Figure 4.23: Sears – Second Replication using Z-score

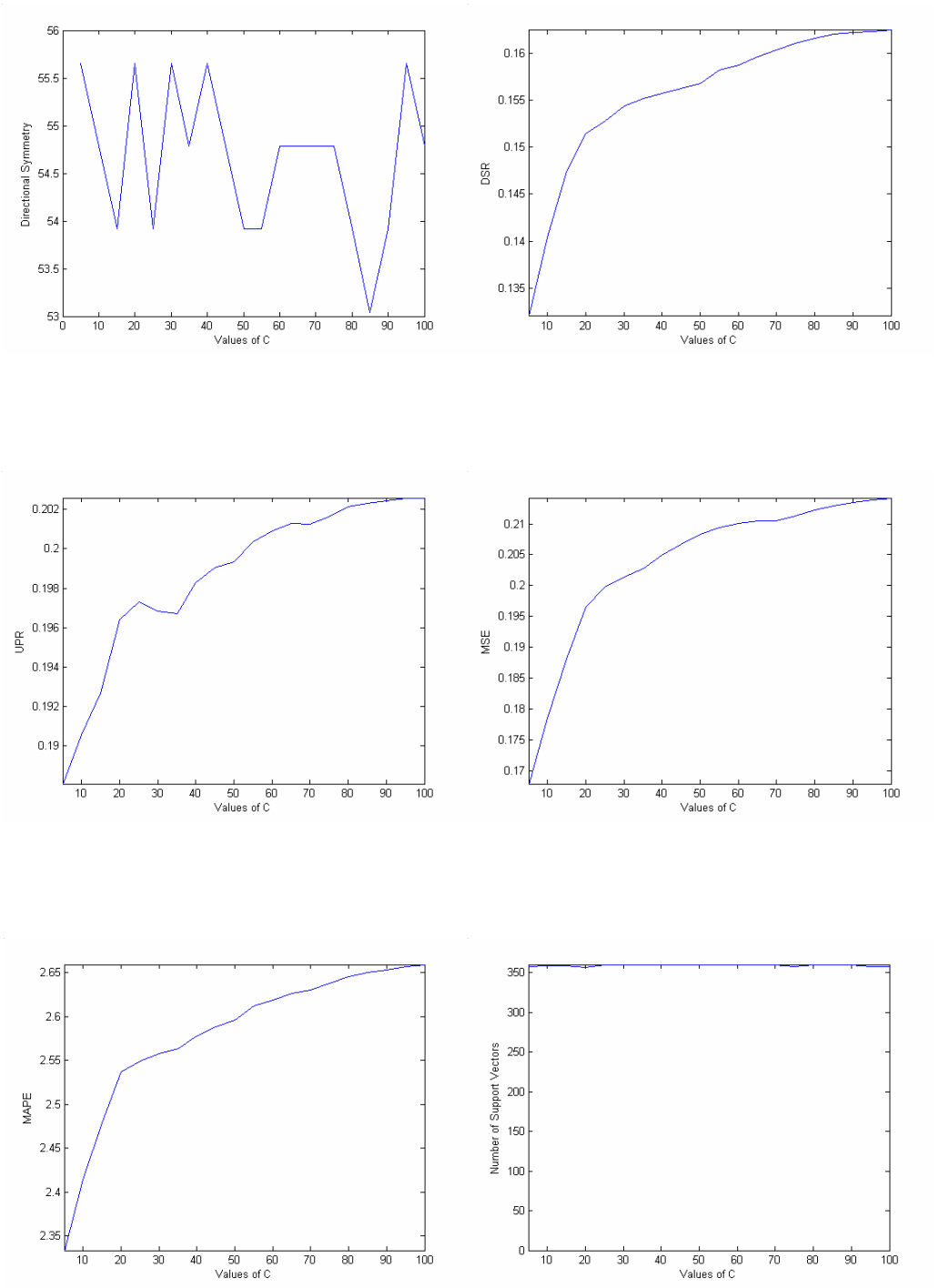



Figure 4.24: Toys ‘R us – First Replication using Z-score

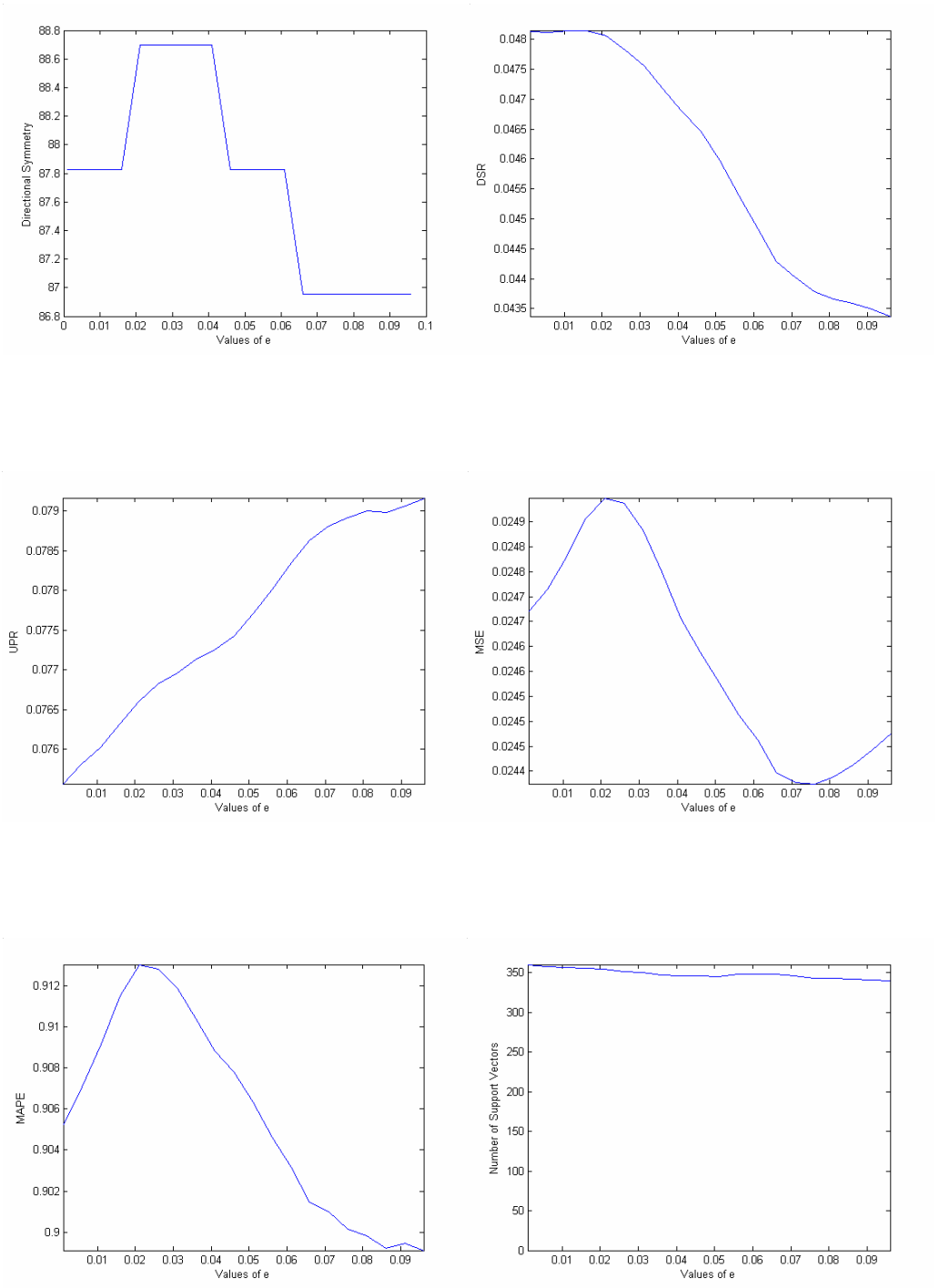



Figure 4.25: Toys ‘R us – Second Replication using Z-score

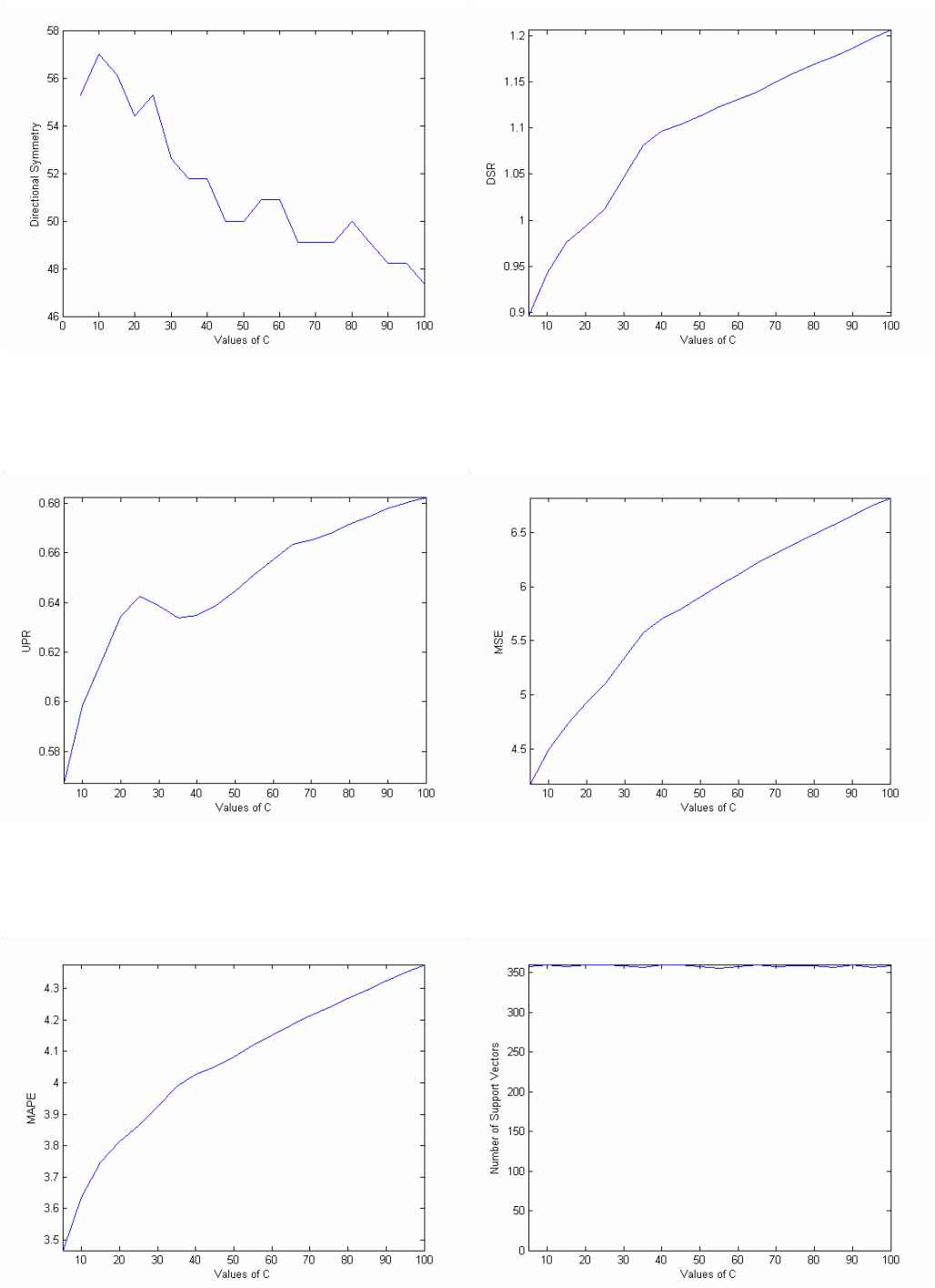



Figure 4.26: Microsoft – First Replication using Z-score

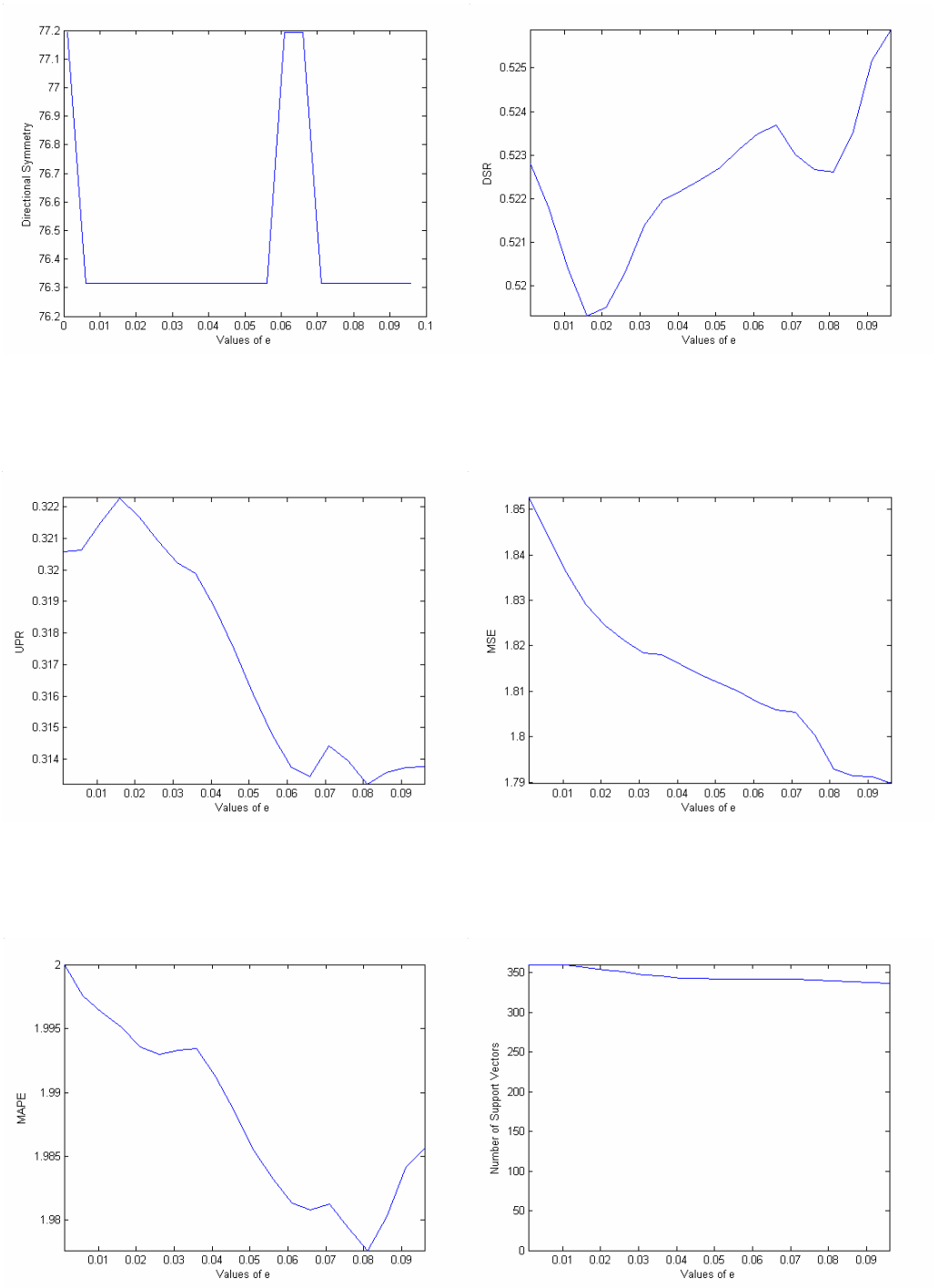



Figure 4.27: Microsoft – Second Replication using Z-score

#### 4.8.2.3 Using Natural Logarithm Transformation

Table 4.4 shows the summary of the results obtained when the SVR model was applied to the logarithms of the five stock prices. Based on the results displayed in the Table, it can be seen that for Log transformation, the value of $C$ varies randomly from 10 to 80 while the value of $\varepsilon$ varies from 0.006 and 0.061, with 0.061 occurring as the optimal $\varepsilon$ value for two (2) of the five stock prices.

The minimum number of support vectors used by the SVR algorithm was 10 (2.8%) for Sears and the maximum was 291 (80.8%) for Toys 'R us. The DS values range from 52.632% to 60.870%, the DSR ranges from 0.103 to 0.714, the UPR ranges from 0.269 to 0.686 and the MSE and MAPE range from 0.332 to 3.644 and 2.229 to 3.170 respectively. Similar to what was obtained with the RDP and Z-score transformations, Toys 'R Us produced the highest and best DS value of 60.870%; the smallest DS was obtained for Microsoft. Toys 'R us had the minimum DSR and MSE values, American Airlines had the minimum UPR, and the smallest MAPE was obtained for McDonalds. Microsoft produced the maximum DSR, MSE and MAPE, while Sears had the largest UPR.

Figure 4.28 to 4.37 show how the performance measures and the number of support vectors were changing as the values of C and $\varepsilon$ were varied during the first and second replications of the experiments using the logarithm transformation.

Table 4.4: Results of SVR using Natural Log Transformation

| **Transformation: Natural Logarithm**<br>**Range of C: 5:100 in steps of 5**<br>**Range of $\varepsilon$ : 0.001:0.1 in steps of 0.005** | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| **Name of Stock** | SVR Parameters and Number of Support Vectors | | | | Performance Metrics | | | | |
| | **Opt C** | **Opt $\varepsilon$** | **NSV** | **%NSV** | **DS** | **DSR** | **UPR** | **MSE** | **MAPE** |
| American Airlines | 60 | 0.041 | 43 | 11.9 | 53.913 | 0.520 | 0.269 | 1.016 | 2.399 |
| McDonalds | 15 | 0.011 | 208 | 57.8 | 53.043 | 0.445 | 0.340 | 1.093 | 2.229 |
| Sears | 10 | 0.061 | 10 | 2.8 | 53.913 | 0.333 | 0.686 | 2.122 | 3.063 |
| Toys ‘R us | 65 | 0.006 | 291 | 80.8 | 60.870 | 0.103 | 0.300 | 0.332 | 3.134 |
| Microsoft | 80 | 0.061 | 14 | 3.9 | 52.632 | 0.714 | 0.631 | 3.644 | 3.170 |

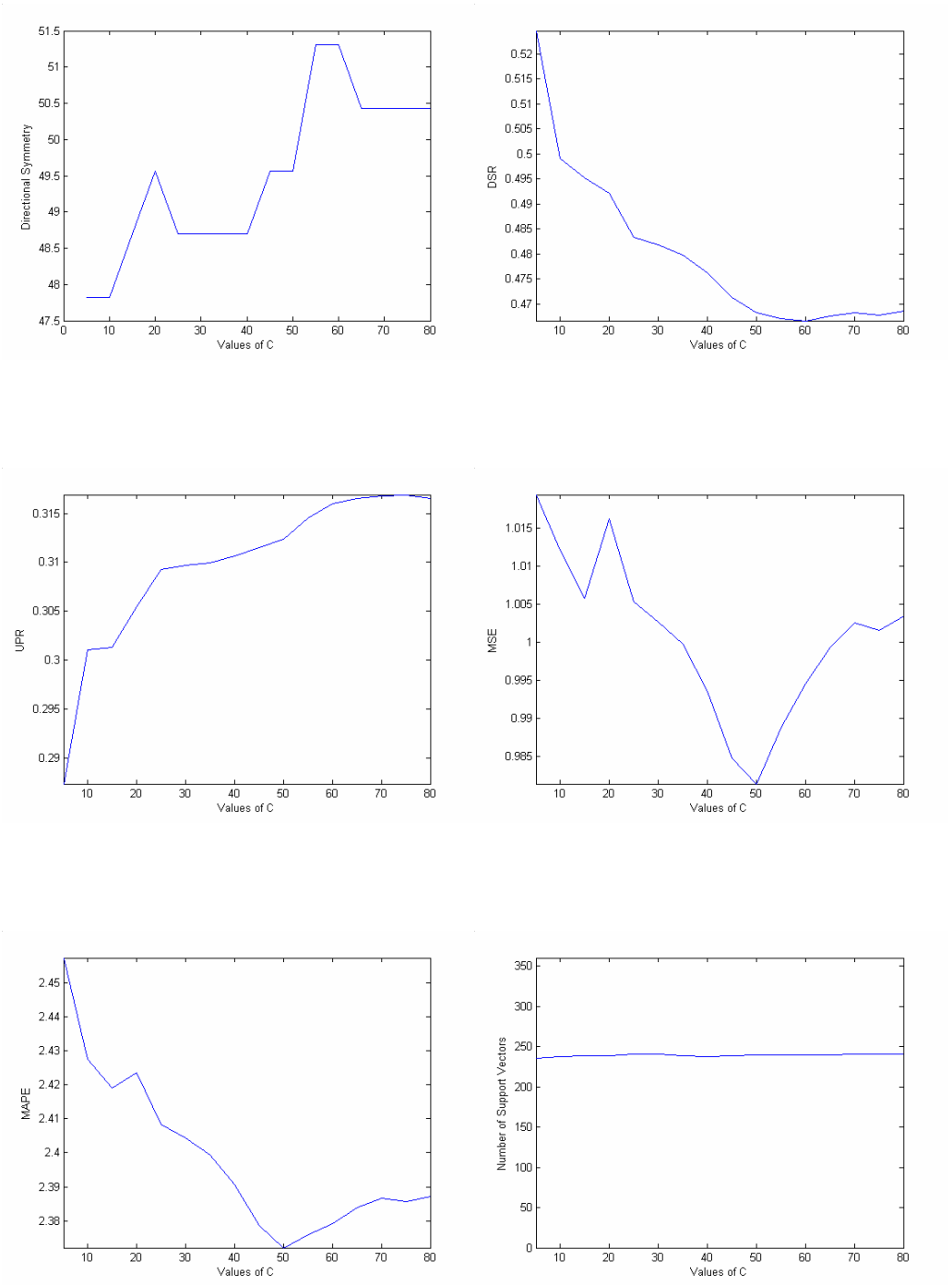



Figure 4.28: American Airlines – First Replication using Natural Logarithm

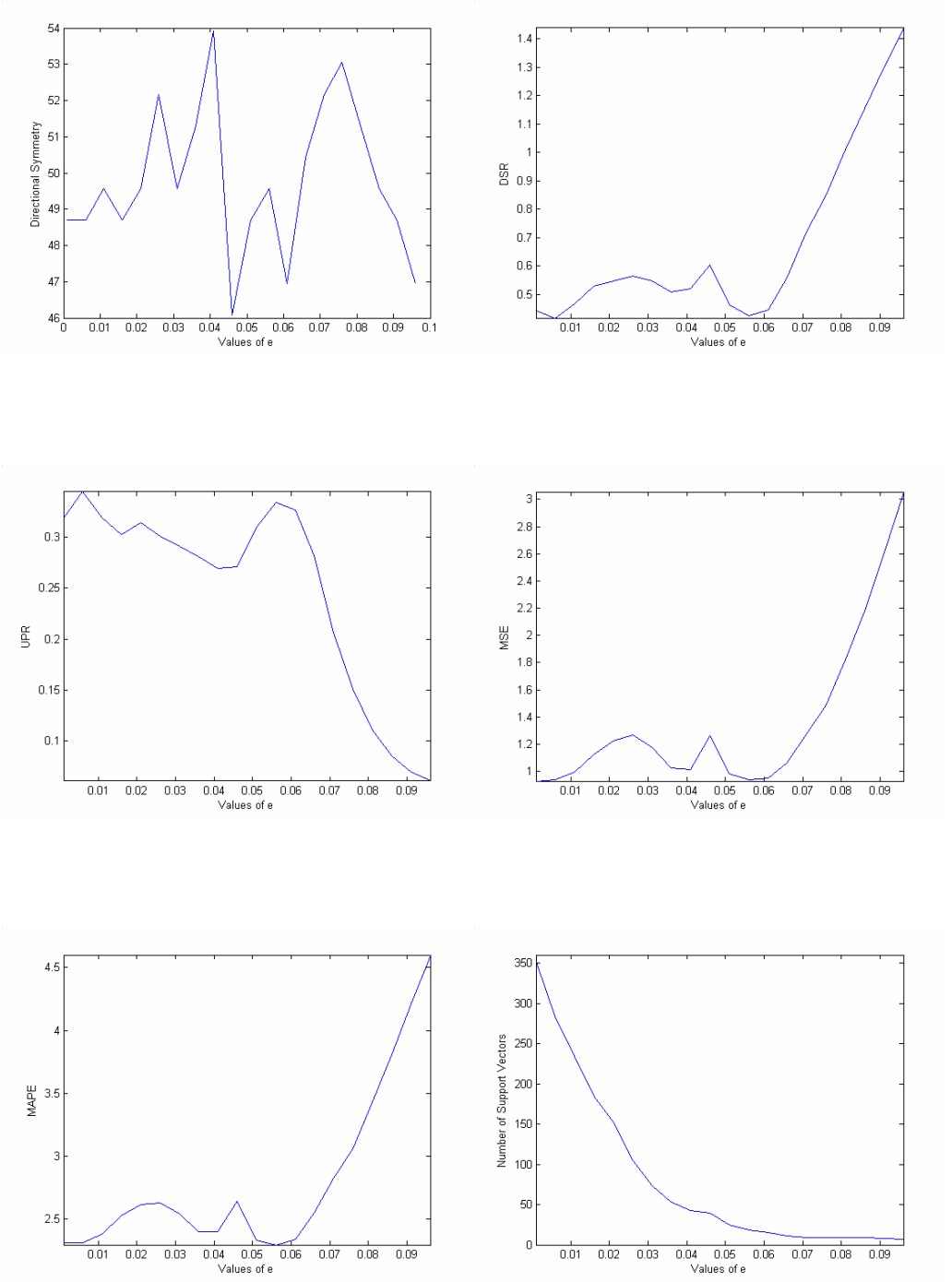



Figure 4.29: American Airlines – Second Replication using Natural Logarithm

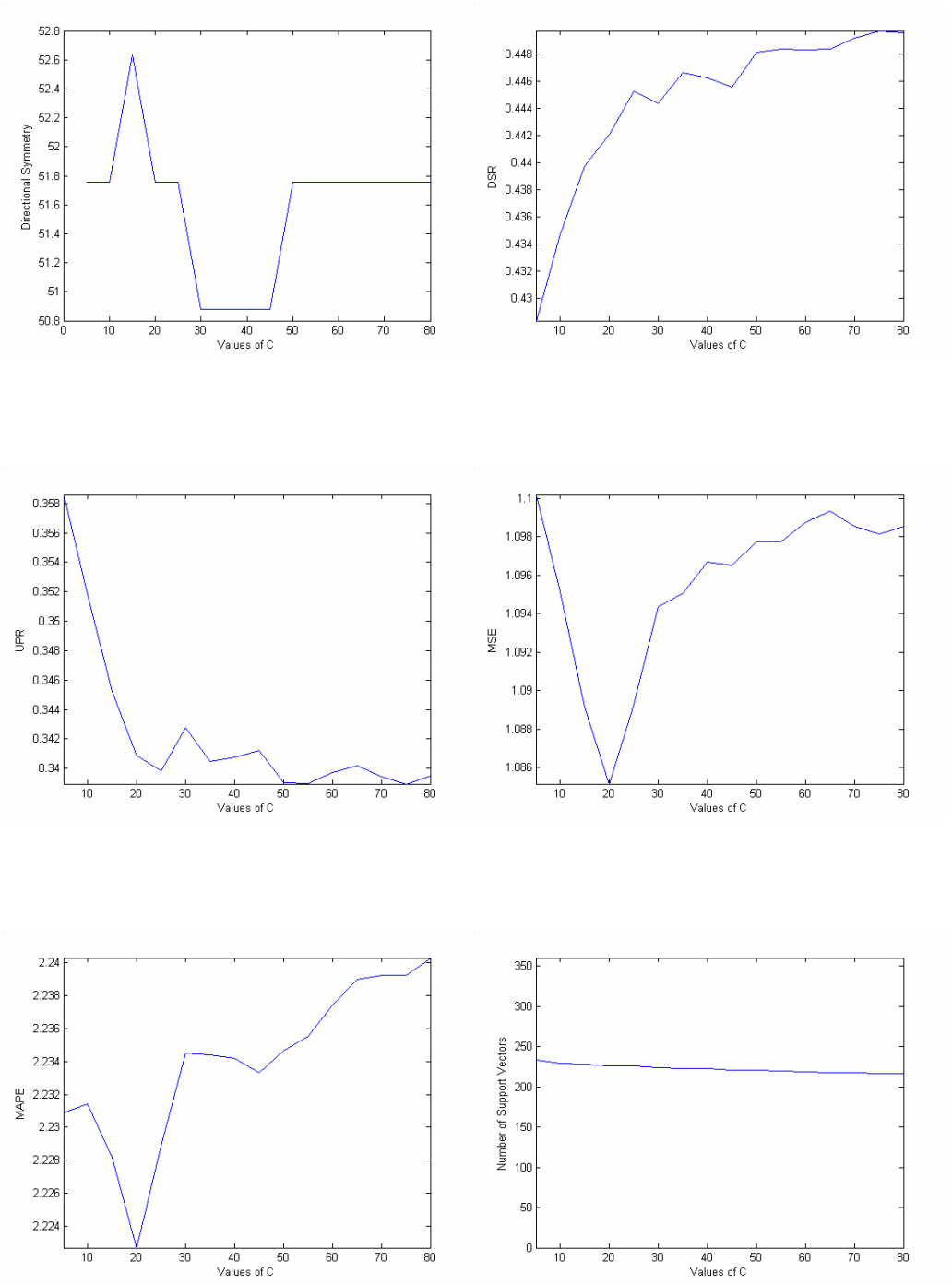



Figure 4.30: McDonalds – First Replication using Natural Logarithm

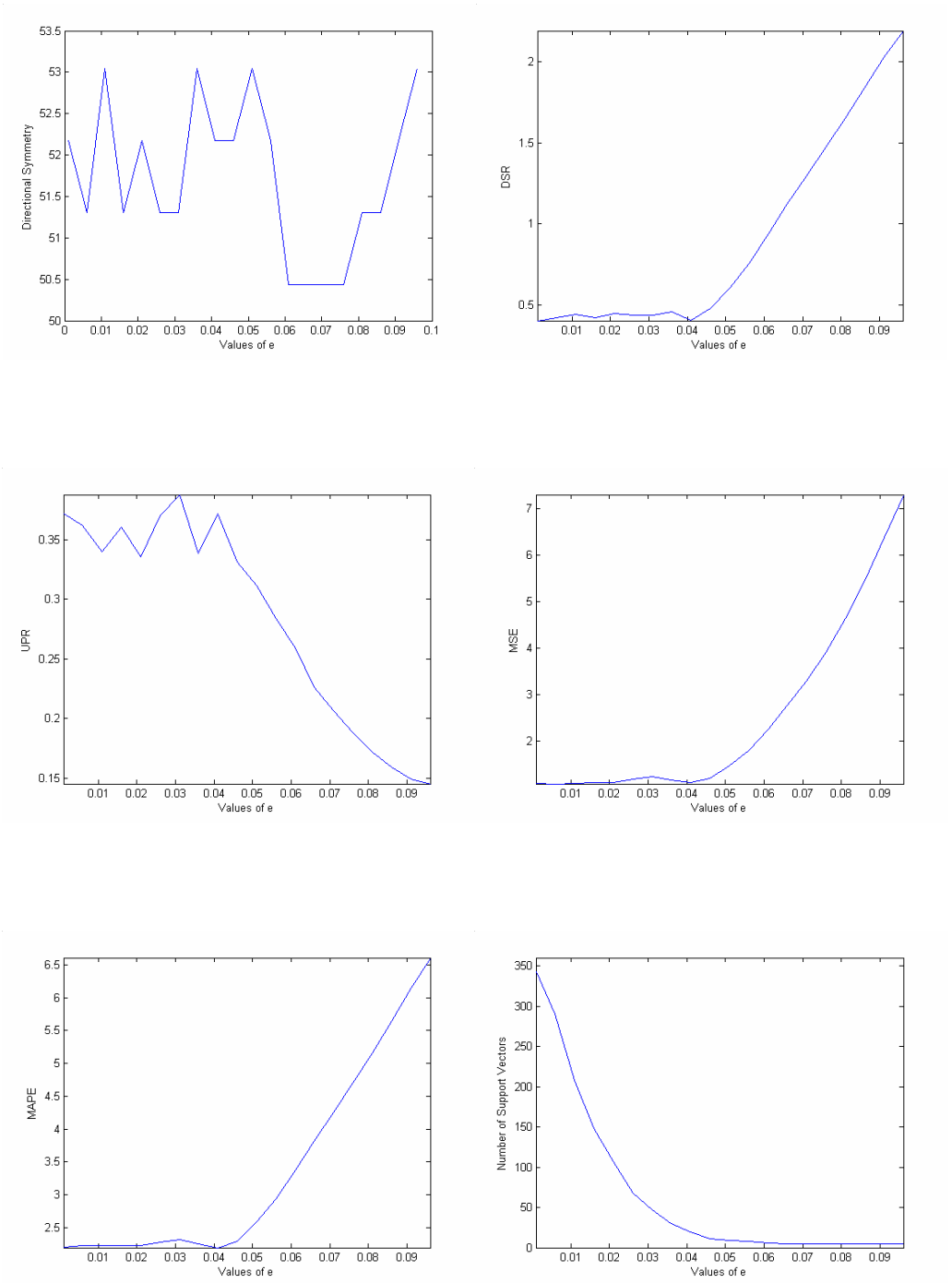



Figure 4.31: McDonalds – Second Replication using Natural Logarithm

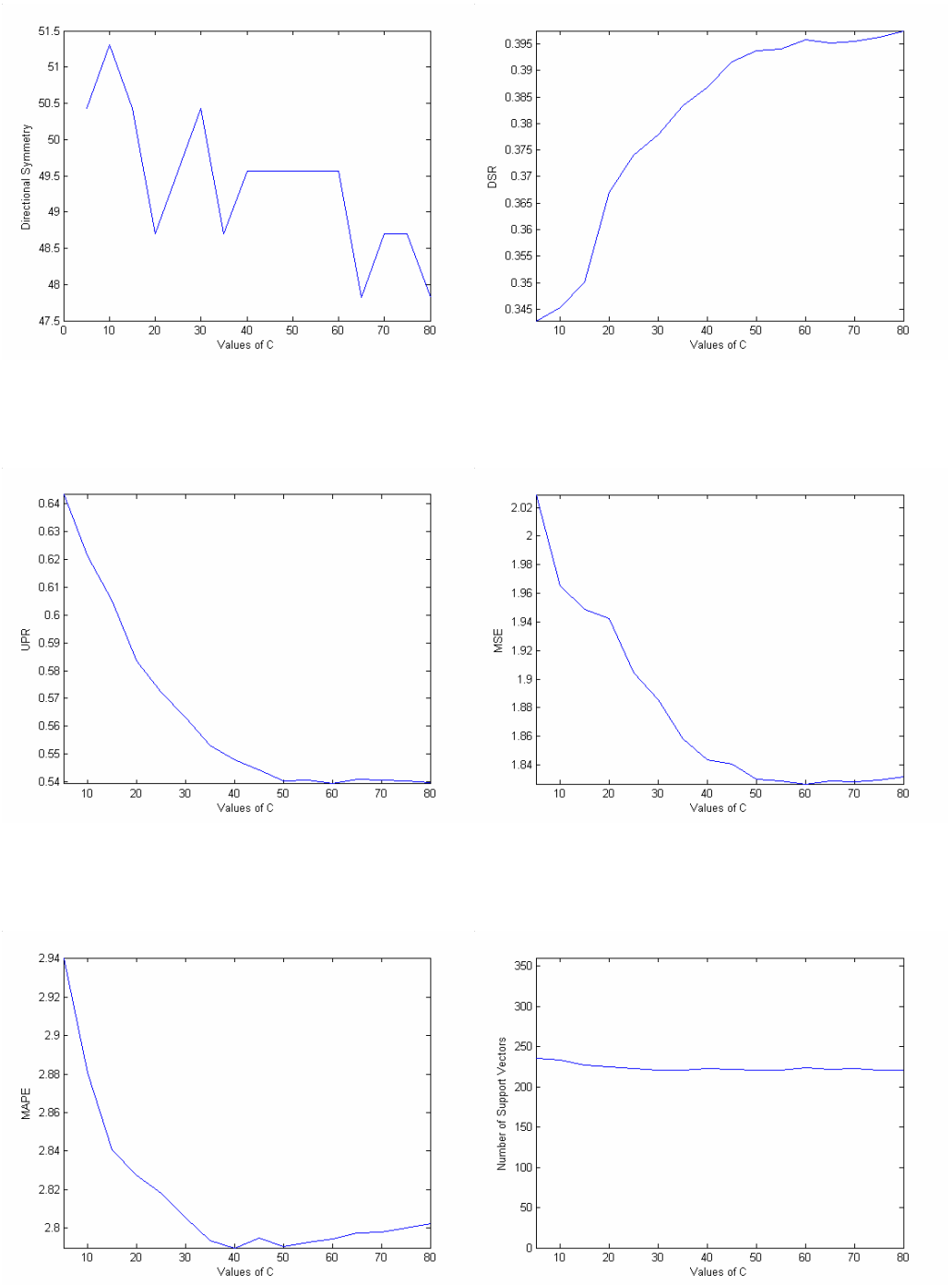



Figure 4.32: Sears – First Replication using Natural Logarithm

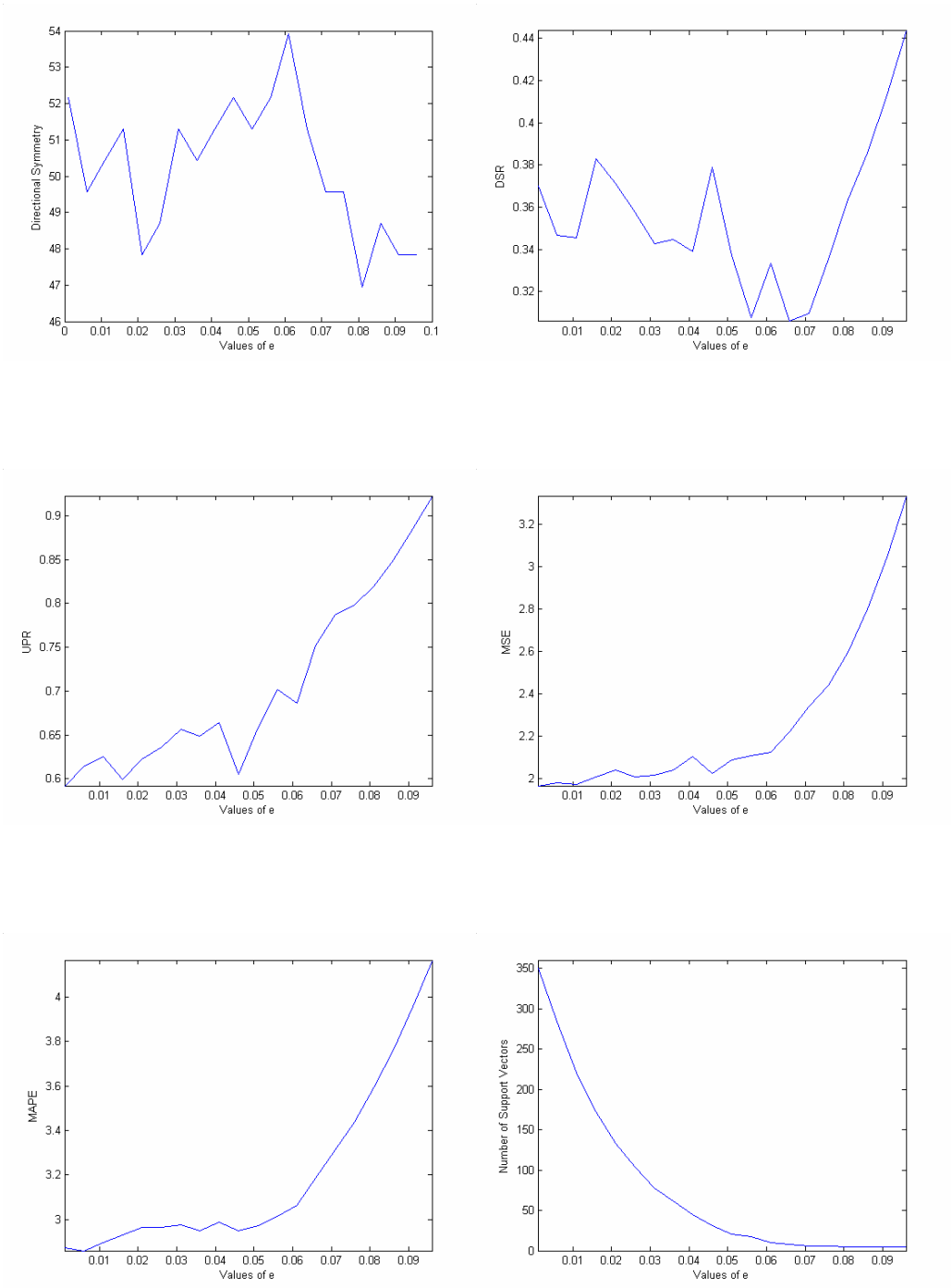



Figure 4.33: Sears – Second Replication using Natural Logarithm

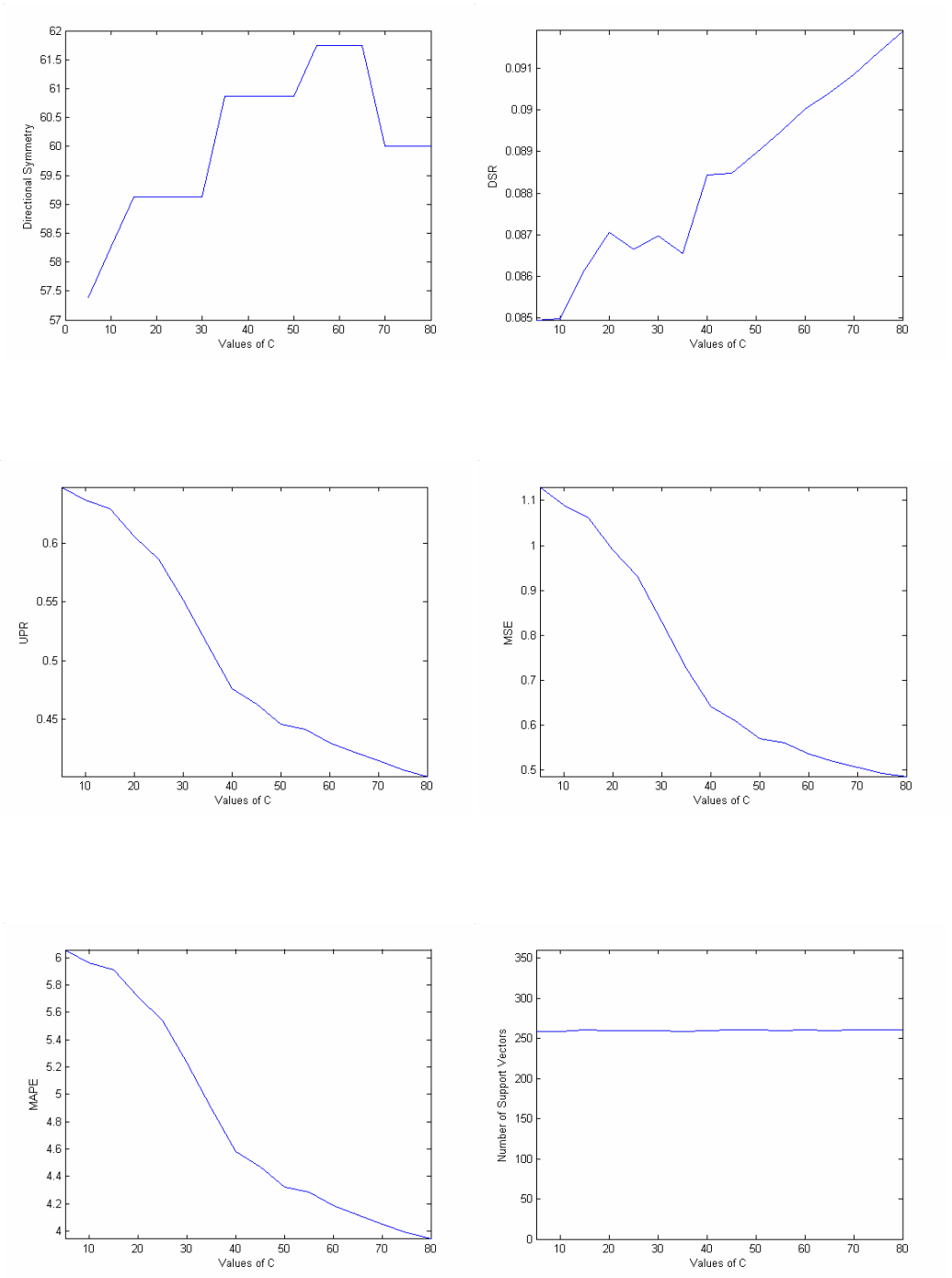



Figure 4.34: Toys ‘R us – First Replication using Natural Logarithm

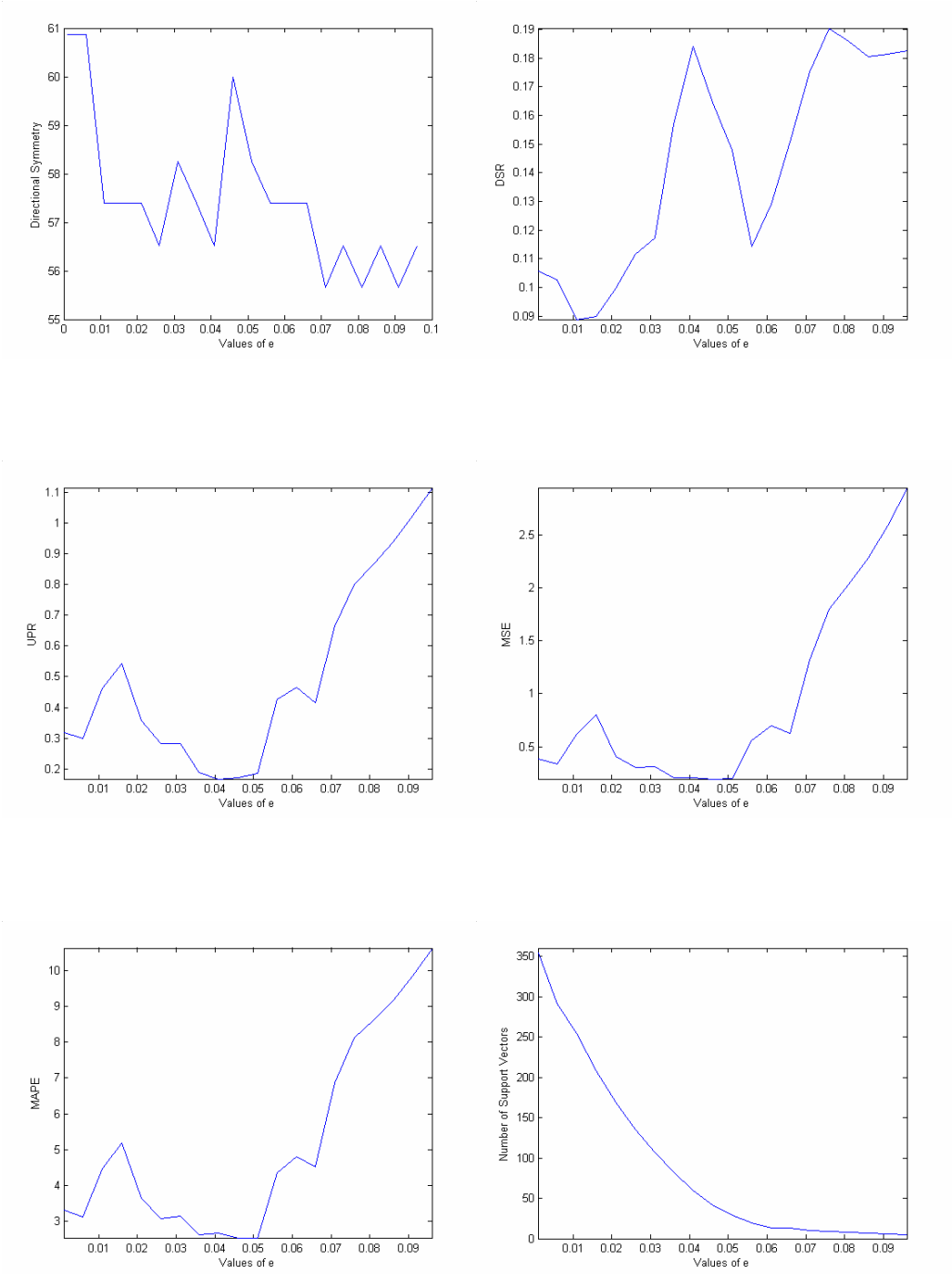



Figure 4.35: Toys ‘R us – Second Replication using Natural Logarithm

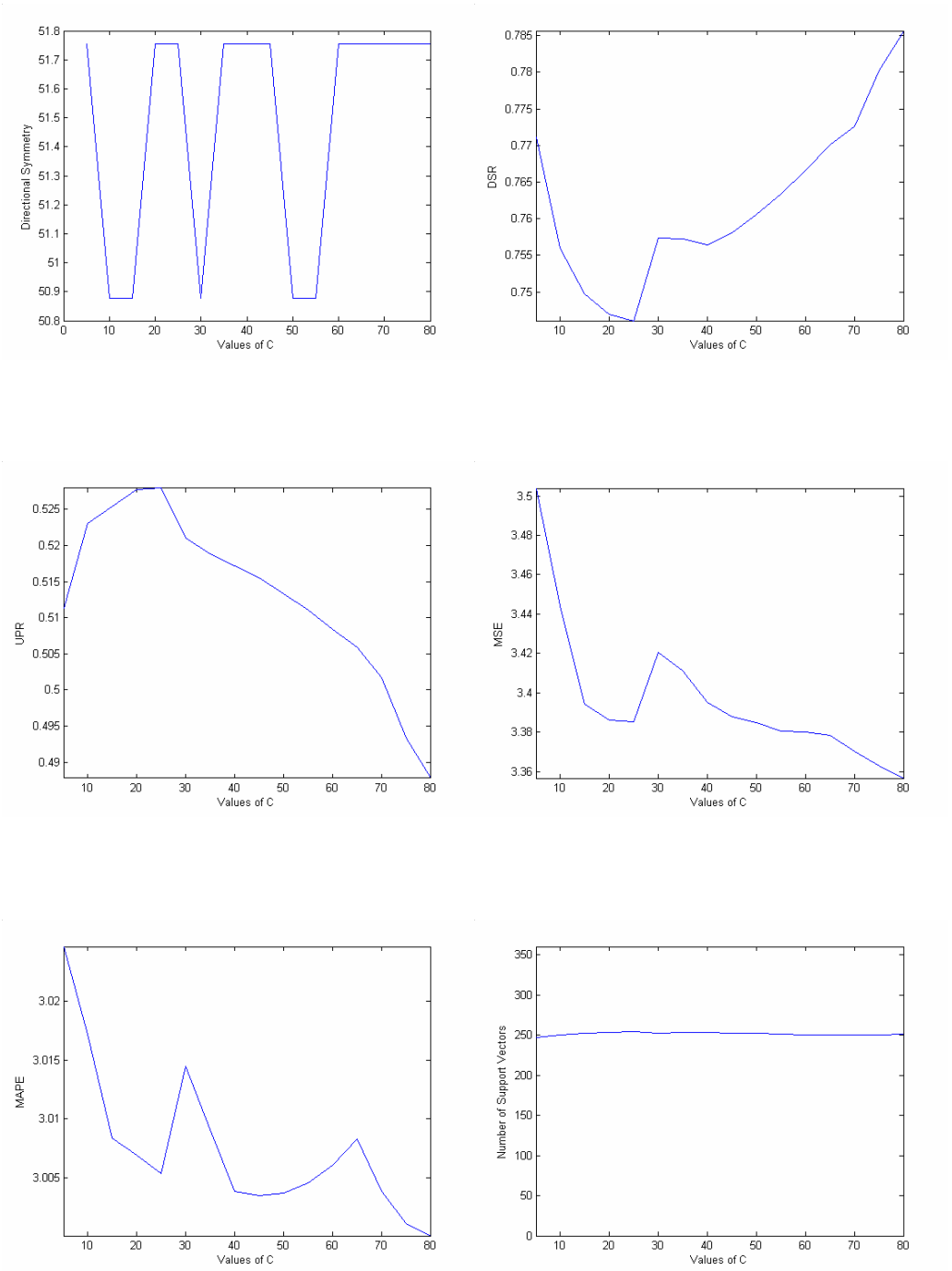



Figure 4.36: Microsoft – First Replication using Natural Logarithm

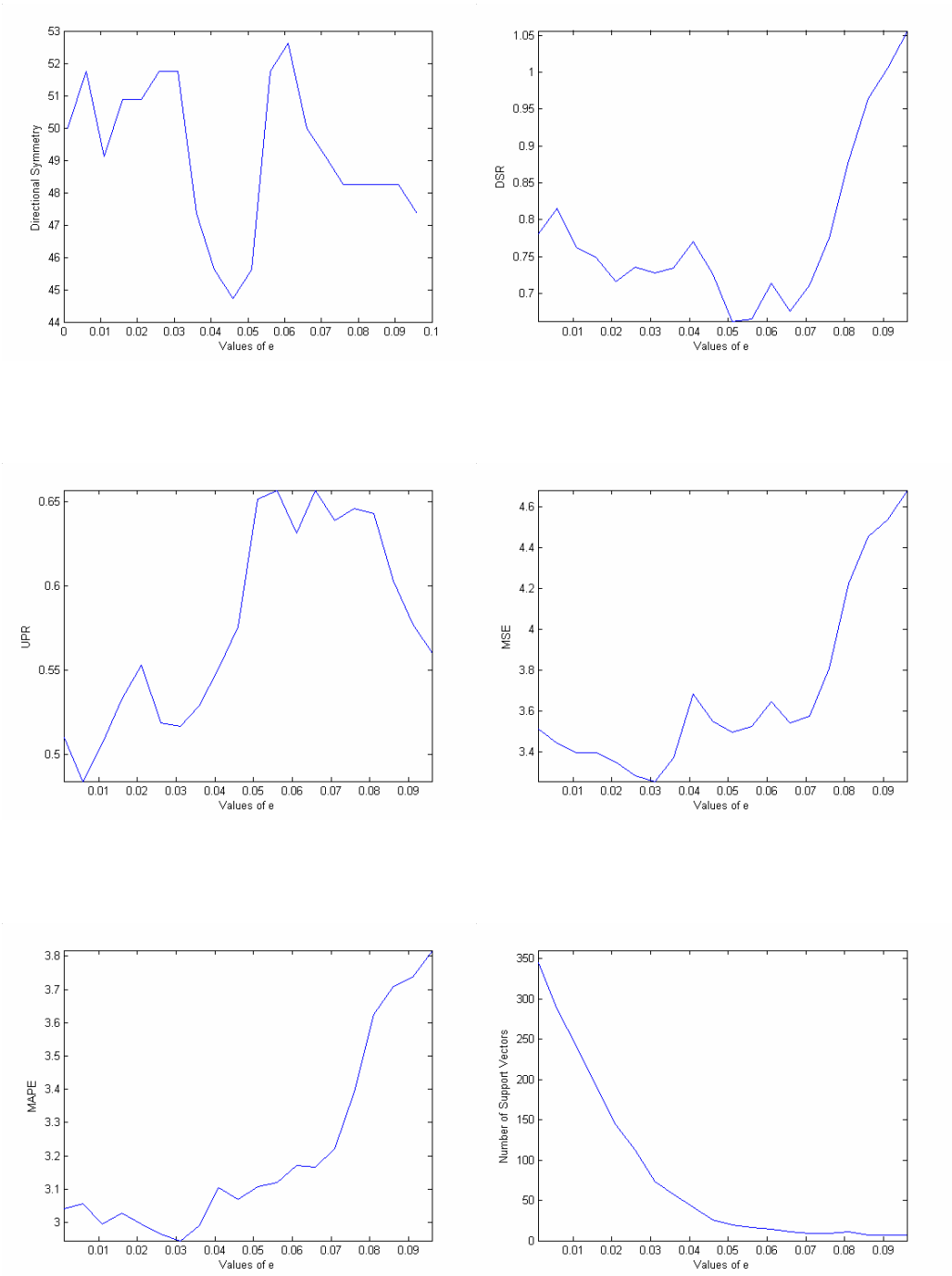



Figure 4.37: Microsoft – Second Replication using Natural Logarithm

#### 4.8.2.4 Comparisons of Results using RDP, Z-score and Log Transformations

A highlight of this study was the overall success of the Z-score transformation. The Z-score method produced better prediction results than either the RDP or the Natural Log on the basis of all the five performance measures considered. Table 4.5 shows the summary of the SVR results for all the transformation methods used on each of the daily stock price series considered. Figures 4.38 to Figure 4.42 represent the plots of the actual and predicted values, and the plots of the prediction errors expressed in percentages, for the five stocks using the Z-score method of scaling. The plots of the percentage errors provide some useful information about the performance of the SVR models that were based on Z-score transformation. For American Airlines, and Toys 'R us, the percentage errors range from -5% to 5%, for McDonalds and Sears, the errors are between -10% to 10% and -10% and 25% respectively and for Microsoft, the percentage errors are between -25% and 15%.

The results also show that log transformations performed better than RDP on the basis of DS for four of the five stocks except McDonalds. However, for all the stocks, the RDP's prediction performance was better than the Log on the basis of MSE and MAPE. On the basis of DSR and UPR, the RDP also performed better than the log for three stock prices and for four out of the five stock price series considered respectively.

Table 4.5: Results of the SVR Experiments using RDP, Z-score and Natural Log Transformations

| Stocks | Method: Support Vector Regression Transformation: RDP | | | | | Method: Support Vector Regression Transformation: Z-score | | | | | Method: Support Vector Regression Transformation: Natural Logarithm | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | DS (%) | DSR | UPR | MSE | MAPE | DS (%) | DSR | UPR | MSE | MAPE | DS (%) | DSR | UPR | MSE | MAPE |
| American Airlines | 47.826 | 0.431 | 0.357 | 0.964 | 2.399 | 74.783 | 0.263 | 0.233 | 0.444 | 1.499 | 53.913 | 0.520 | 0.269 | 1.016 | 2.399 |
| McDonalds | 53.913 | 0.415 | 0.325 | 0.984 | 2.101 | 73.043 | 0.331 | 0.221 | 0.649 | 1.563 | 53.043 | 0.445 | 0.340 | 1.093 | 2.229 |
| Sears | 52.174 | 0.426 | 0.465 | 1.581 | 2.673 | 76.316 | 0.338 | 0.406 | 1.472 | 2.243 | 53.913 | 0.333 | 0.686 | 2.122 | 3.063 |
| Toys 'R us | 56.522 | 0.133 | 0.164 | 0.153 | 2.190 | 88.696 | 0.047 | 0.077 | 0.025 | 0.909 | 60.870 | 0.103 | 0.300 | 0.332 | 3.134 |
| Microsoft | 50 | 0.666 | 0.510 | 2.835 | 2.776 | 77.193 | 0.524 | 0.314 | 1.806 | 1.981 | 52.632 | 0.714 | 0.631 | 3.644 | 3.170 |

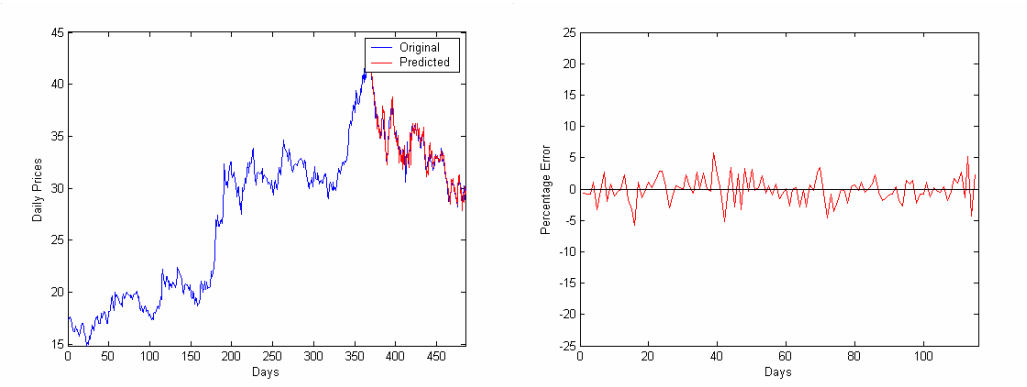



Figure 4.38: Plot of Predicted Values and Percentage Errors for American Airlines

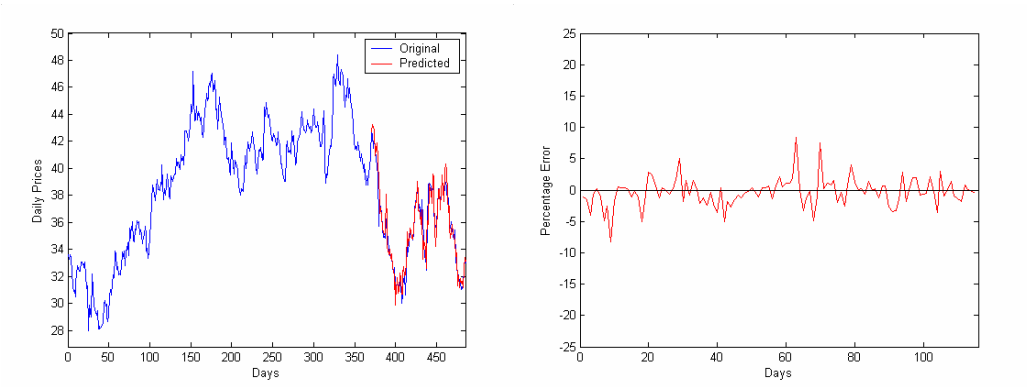



Figure 4.39: Plot of Predicted Values and Percentage Errors for McDonalds

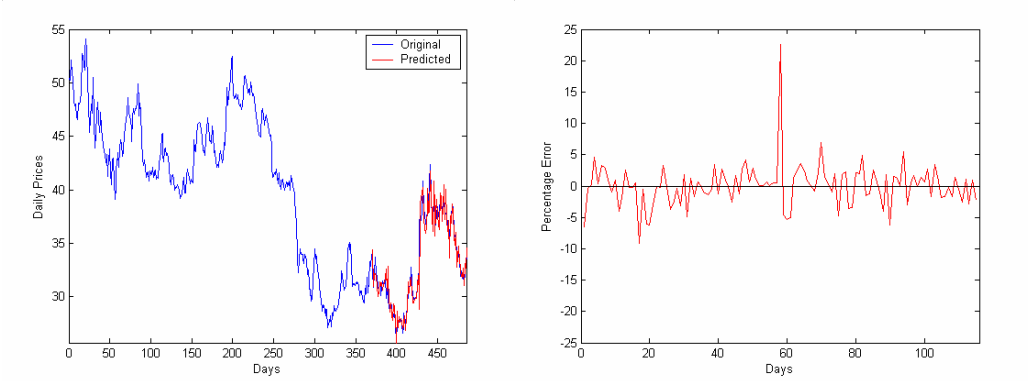



Figure 4.40: Plot of Predicted Values and Percentage Errors for Sears

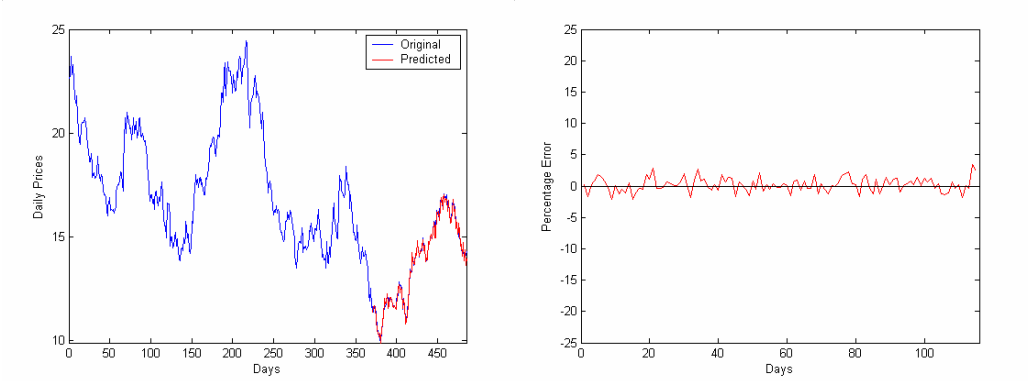



Figure 4.41: Plot of Predicted Values and Percentage Errors for Toys 'R us

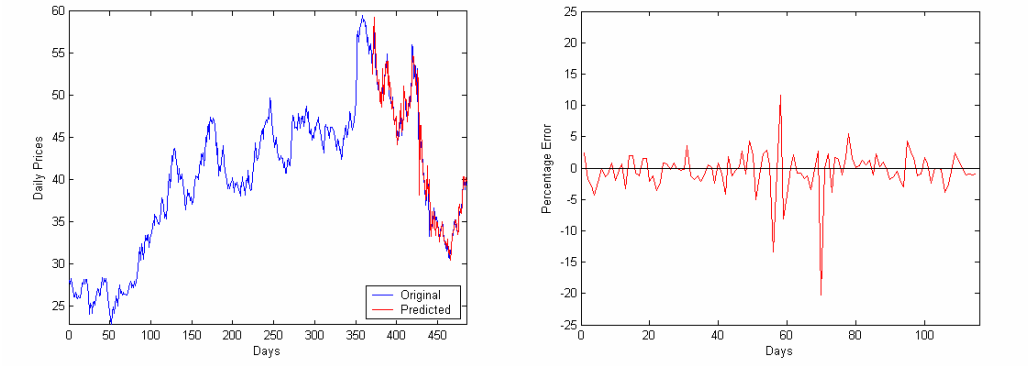



Figure 4.42: Plot of Predicted Values and Percentage Errors for Microsoft

Although, the RDP performed better on most or all of the stocks than the Log transform on the basis of DSR, UPR, MSE and MAPE, we were surprised that the Log transform beat the RDP for four of the stocks on the basis of directional symmetry which is the main performance measure used in this study. Even for the McDonalds stock price, the DS obtained using RDP was only slightly better that the one obtained using Log transformation. This result is somewhat disappointing because RDP is a more popular transformation technique in the literature than Z-score and Log data transformations because it has a more meaningful interpretation among other benefits (refer to Section 4.3.3). Unfortunately, there are no visible reasons for the poor performance of the RDP; however we believe the peculiar nature of our data may have contributed to the low performance.

It is worthy to note that for each of the three data transformations, the highest DS and the minimum DSR, and MSE values were obtained for the Toys 'R us Stock Price. This is not surprising because re-examining the time plots in Figures 4.1 to 4.5; we can infer that this data is almost a linear series as displayed in Figure 4.4. It is perhaps the most stable or most stationary of the five data series making it the easiest to predict.

### 4.8.3 Methodology using Simple Moving Averages

Models based on Simple Moving Averages (SMA) were constructed after the series have been transformed by applying RDP transformation, Z-score and Natural Logarithm transformations. However, the results obtained using both the Z-score transformation and the Natural Log transformations were so poor that we decided not to consider them in

our analysis. Although not found in the literatures, discussions with financial analysts indicated that forecasting models developed for stock market predictions, which are based on moving averages often involves converting the data first to RDP, and then making predictions of the transformed data. The predictions are then converted back to the original scale for the purpose of performing the necessary model evaluations and diagnostics.

Although an "Embedding" of five was used for all the SVR experiments, for a more comprehensive study involving the use of the SMA, the Embedding size was varied from 5 to 300 in steps of 5 for the SMA calculations. With this setting, a total of 5 experiments, one for each of the stock prices was performed.

The SMA worked by taking the most recent k-day RDPs of the stock prices (where k here is the varying Embedding sizes), finds the mean of these selected prices and set that as the prediction for the next element in the series. The predicted value is then converted back to the original scales for each of the stock prices.

### 4.8.4 Experimental Results of Simple Moving Averages

The experimental results obtained using SMA are discussed in this section.

#### 4.8.4.1 Using Varying Embedding Sizes

Table 4.6 shows the summary of the results obtained when SMA models were developed for each of the stock prices after they have been converted to relative differences in percentage. As was done with SVR, the optimal Embedding sizes were selected based on the directional symmetry criterion. Looking at the results displayed in Table 4.6, we can see that the optimal Embedding size was not the same for all the stocks. At these Embedding sizes, the DS values range from 50% to 61.739%. The DSR ranges from 0.164 to 0.632, the UPR ranges from 0.159 to 0.557, and the MSE and MAPE range from 0.163 to 2.877 and 2.069 to 2.807 respectively. The highest and best DS of 61.739% was again achieved with Toys 'R us and the smallest was seen for Microsoft. The minimum DSR, UPR, and MSE were also obtained for the Toys 'R us stock price. However, the smallest MAPE was achieved with McDonalds. The worst results in terms of DSR, UPR, MSE, and MAPE were obtained for Microsoft.

Figures 4.43 to 4.47 show how the performance measures changed as the Embedding sizes were varied from 5 to 300. Figures 4.48 to 4.52 represent the plots of the actual and predicted values, and the plots of the prediction errors expressed in percentages, for the five stocks using the optimal Embedding sizes as displayed in Table 4.6.

For American Airlines, McDonalds and Toys 'R us, the percentage errors range from -15% to 10%, for Sears, the errors are between -10% and 20%, and for Microsoft, the percentage errors are between -20% and 10%.

Table 4.6: Prediction Performance using Simple Moving Averages

| **Method: Simple Moving Averages**<br>**Transformation: RDP**<br>**Range of Embedding: 5:300 in steps of 5** | | | | | | |
|---|---|---|---|---|---|---|
| **Name of Stock** | **Optimal Embedding** | **DS** | **DSR** | **UPR** | **MSE** | **MAPE** |
| American Airlines | 30 | 51.304 | 0.408 | 0.372 | 0.972 | 2.371 |
| McDonalds | 215 | 54.783 | 0.396 | 0.333 | 0.968 | 2.069 |
| Sears | 300 | 53.043 | 0.418 | 0.467 | 1.583 | 2.653 |
| Toys ‘R us | 5 | 61.739 | 0.164 | 0.159 | 0.163 | 2.400 |
| Microsoft | 20 | 50 | 0.632 | 0.557 | 2.877 | 2.807 |

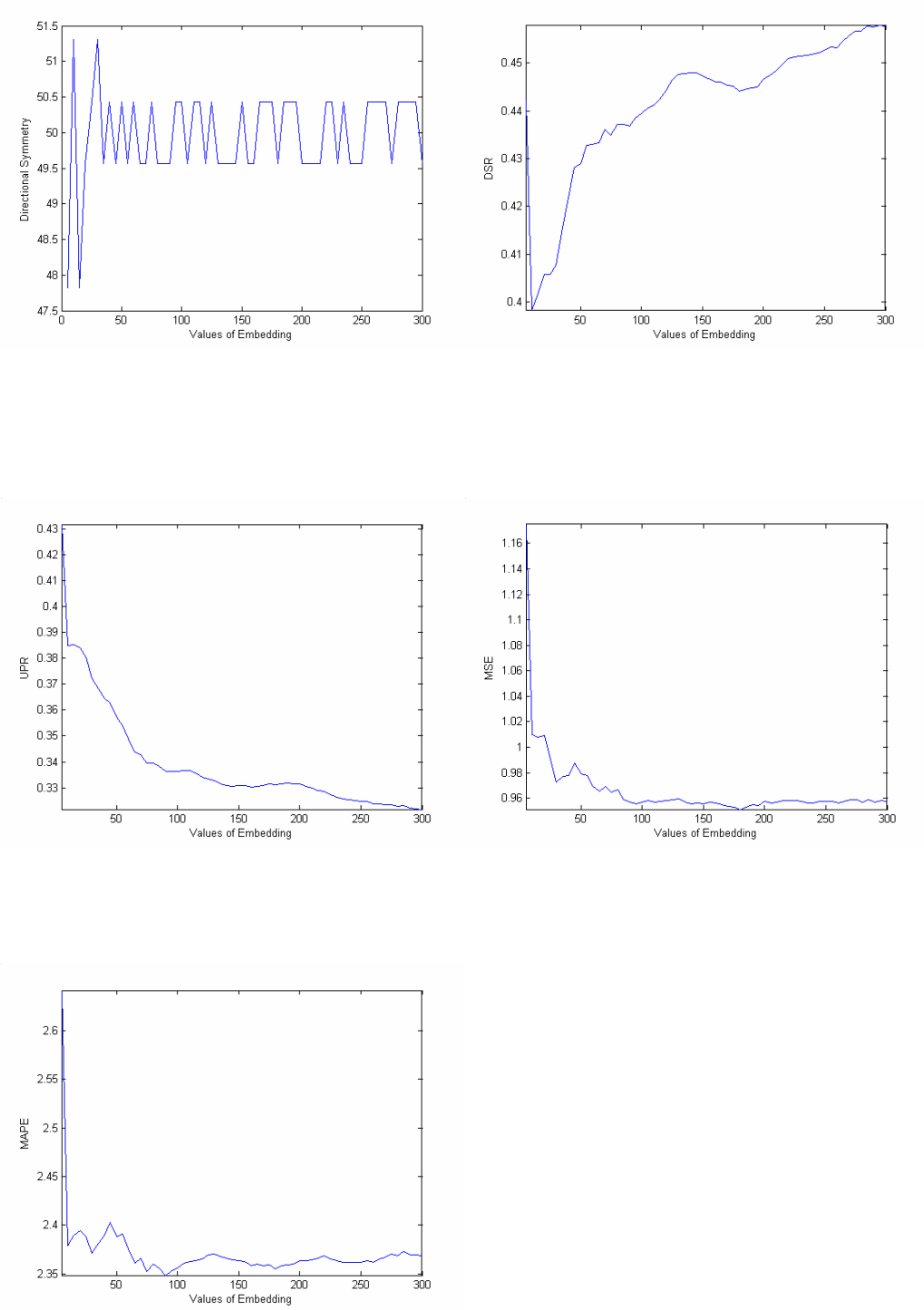



Figure 4.43: American Airlines: Changing Performance Metrics for Varying Embedding Sizes

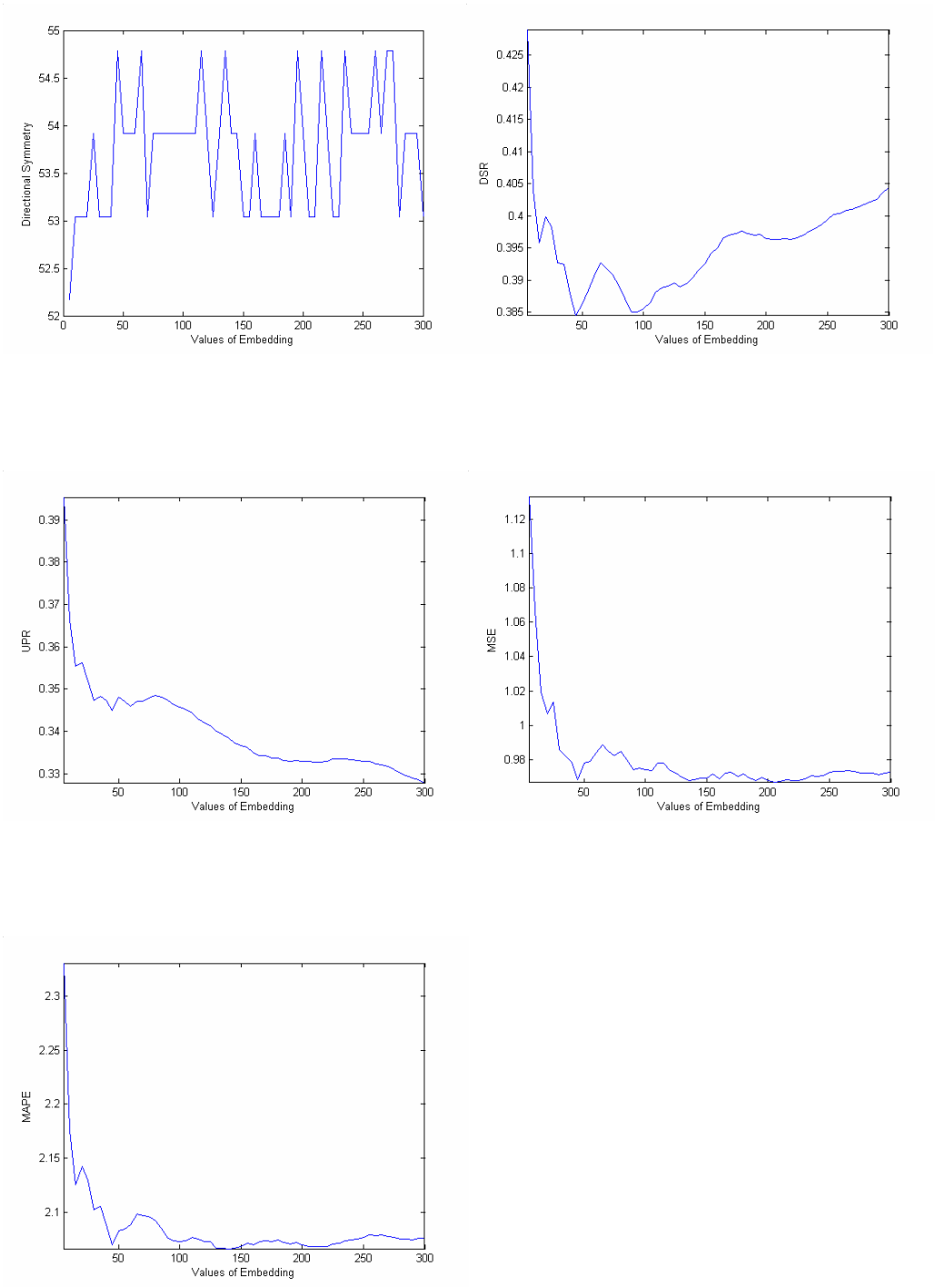



Figure 4.44: McDonalds: Changing Performance Metrics for Varying Embedding Sizes

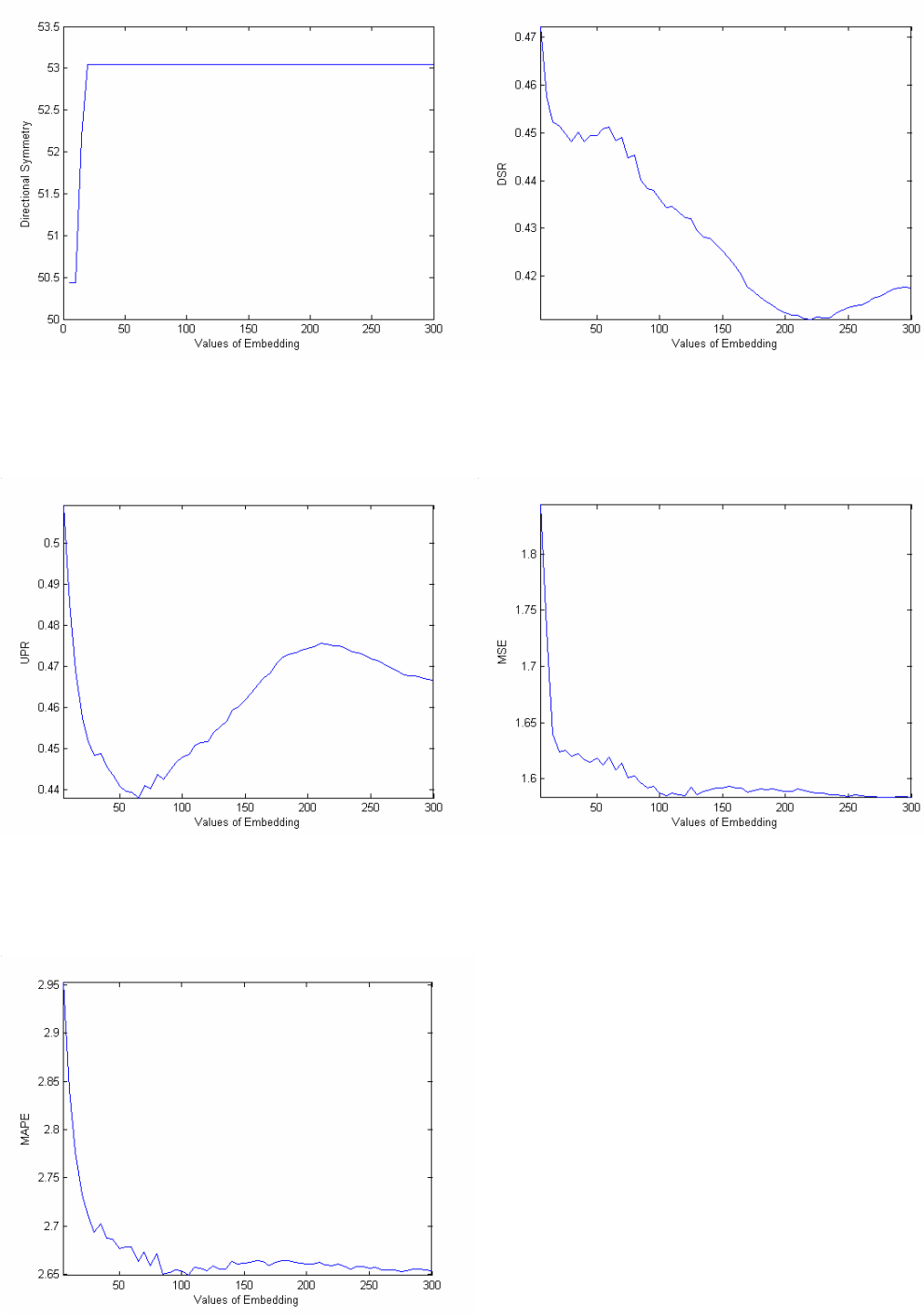



Figure 4.45: Sears: Changing Performance Metrics for Varying Embedding Sizes

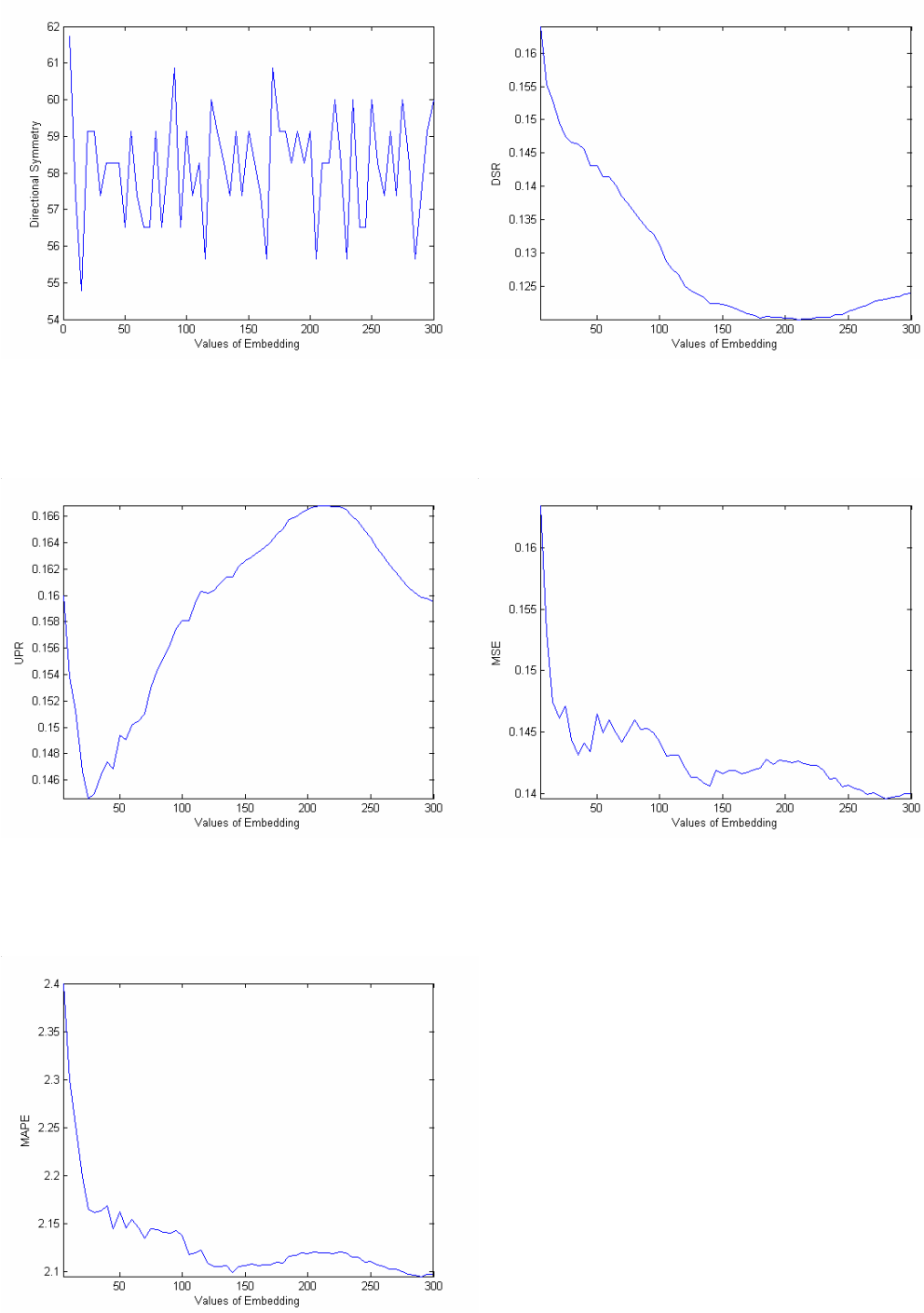



Figure 4.46: Toys ‘R us: Changing Performance Metrics for Varying Embedding Sizes

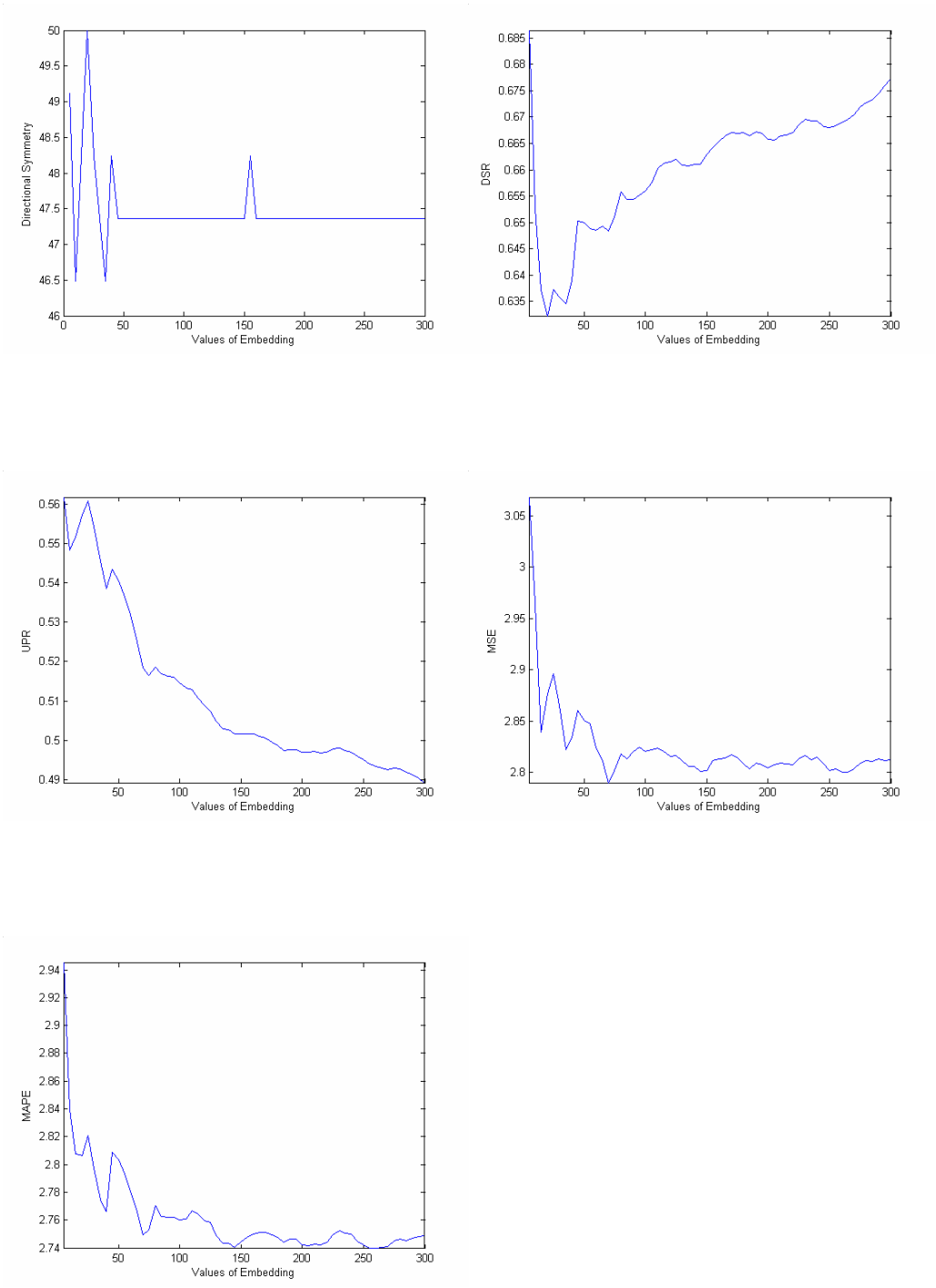



Figure 4.47: Microsoft: Changing Performance Metrics for Varying Embedding Sizes

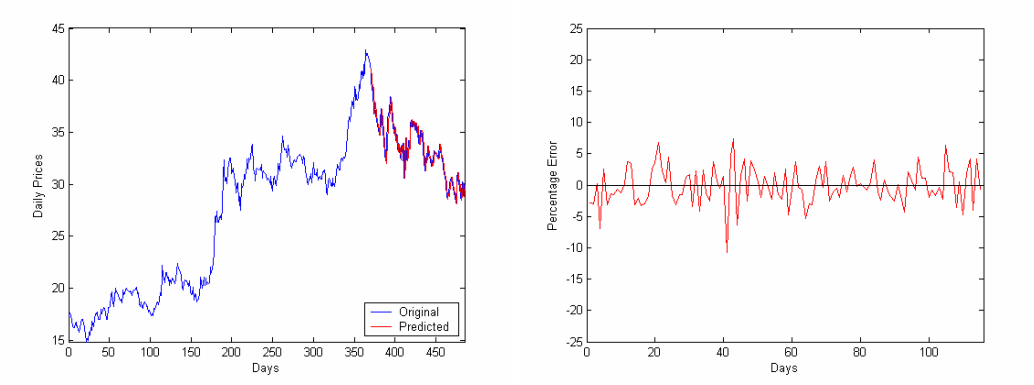



Figure 4.48: Plot of Predicted Values and Percentage Errors for American Airlines using Optimal Embedding Size

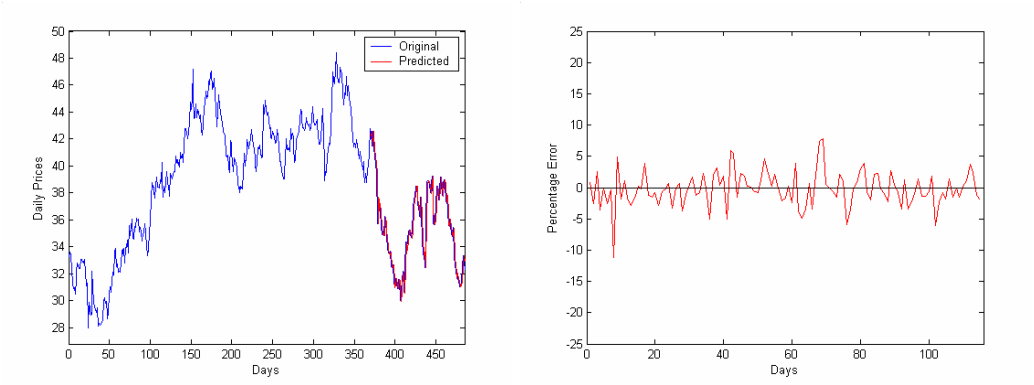



Figure 4.49: Plot of Predicted Values and Percentage Errors for McDonalds using Optimal Embedding Size

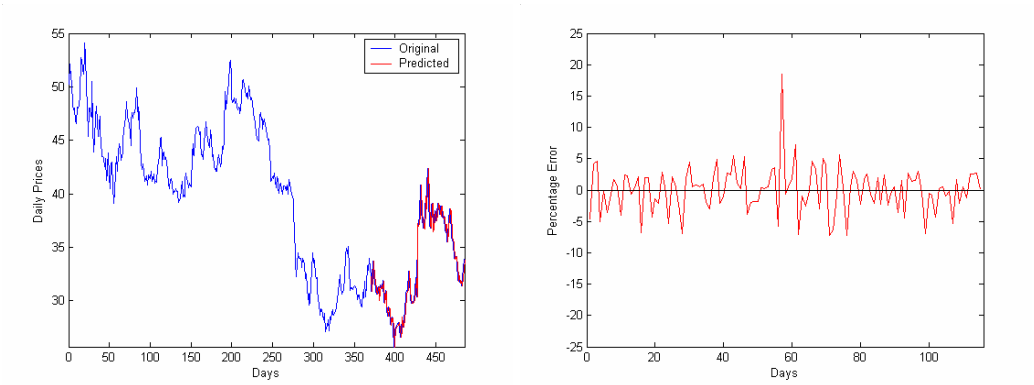



Figure 4.50: Plot of Predicted Values and Percentage Errors for Sears using Optimal Embedding Size

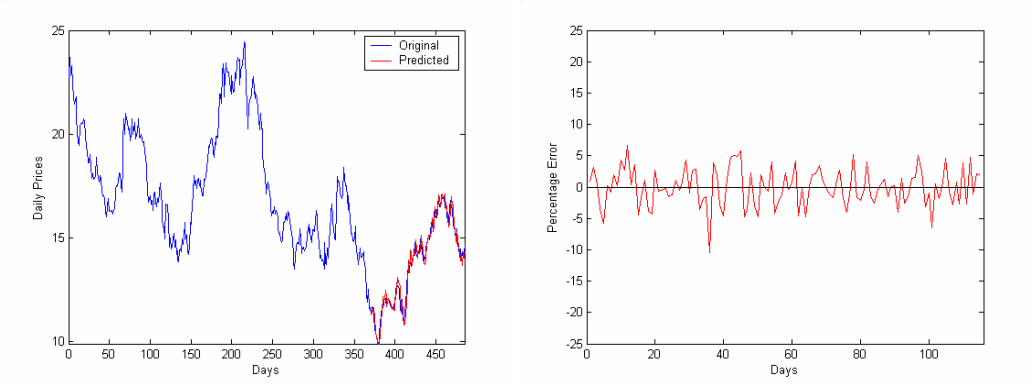



Figure 4.51: Plot of Predicted Values and Percentage Errors for Toys ‘R us using Optimal Embedding Size

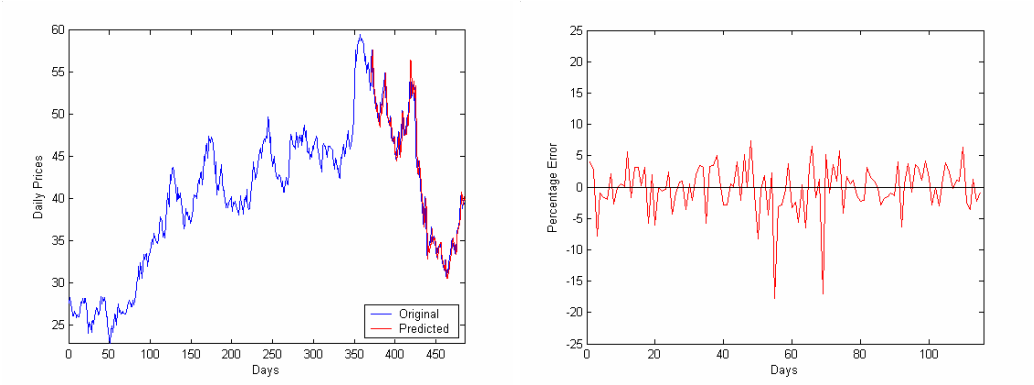



Figure 4.52: Plot of Predicted Values and Percentage Errors for Microsoft using Optimal Embedding size

#### 4.8.4.2 Using Constant Embedding Size

From Table 4.6, we see that the optimal Embedding sizes for most of the stock prices were different from 5; however, as we may recall, all our SVR models were built using Embedding size of 5. Therefore, it makes sense to fix the Embedding at this value for the SMA models as well, so comparisons can be made with the results obtained using SVR.

Table 4.7 shows the values of the performance measures obtained for each of the stock prices when the Embedding size was set to 5. For this scenario, the directional symmetry values range from 47.826% for American airlines to 61.739% for Toys 'R us. The downside risks are between 0.164 and 0.686 while the upside risks are between 0.159 and 0.562. The ranges of the MSE and MAPE values are 0.163 to 3.068 and 2.331 to 2.945 respectively. Toys 'R Us produced the best results in terms of the minimum downside risk, upside risk and MSE values. However, the minimum MAPE was obtained for McDonalds. The prediction performance of the SMA model for Microsoft was poor producing the largest values of DSR, UPR, MAE and MAPE. Even for the DS, Microsoft was only slightly better than American Airlines.

Figures 4.53 to Figure 4.57 represent the plots of the actual and predicted values, and the plots of the prediction errors expressed in percentages, for the five stocks using an Embedding size of 5. For McDonalds, and Toys 'R us, the percentage errors are within the -10% to 10% range, percentage errors for Sears fall between -10% and 20%, while the errors for American Airlines and Microsoft are between -15% and 10% and -20% and 10% respectively.

Table 4.7: Results of Simple Moving Averages for Constant Embedding Size

| **Method: Simple Moving Averages**<br>**Transformation: RDP**<br>**Embedding: 5** | | | | | | |
|---|---|---|---|---|---|---|
| **Name of Stock** | **Embedding Size** | **DS (%)** | **DSR** | **UPR** | **MSE** | **MAPE** |
| American Airlines | 5 | 47.826 | 0.442 | 0.432 | 1.175 | 2.642 |
| McDonalds | 5 | 52.174 | 0.429 | 0.395 | 1.133 | 2.331 |
| Sears | 5 | 50.435 | 0.472 | 0.509 | 1.844 | 2.952 |
| Toys 'R us | 5 | 61.739 | 0.164 | 0.159 | 0.163 | 2.400 |
| Microsoft | 5 | 49.123 | 0.686 | 0.562 | 3.068 | 2.945 |

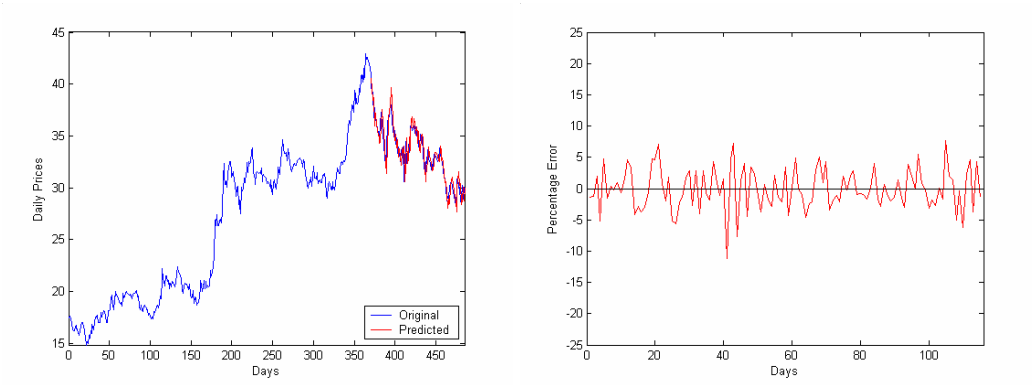



Figure 4.53: Plot of Predicted Values; Percent Errors for American Airlines using Embedding of 5

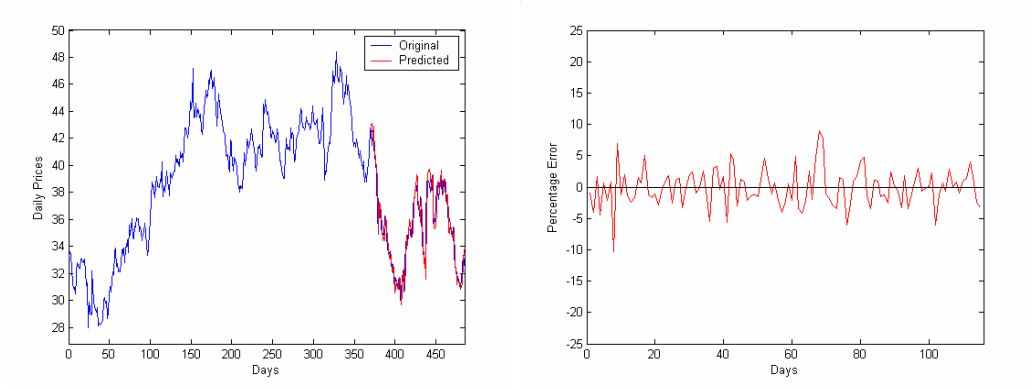



Figure 4.54: Plot of Predicted values; Percent Errors for McDonalds using Embedding of 5

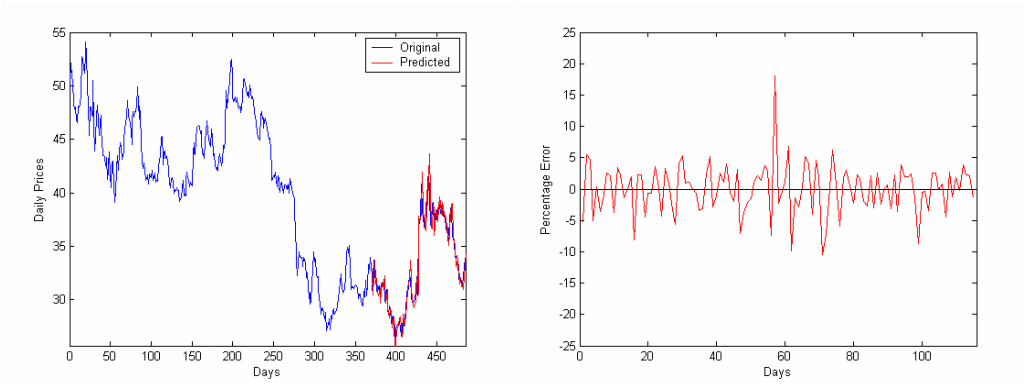



Figure 4.55: Plot of Predicted Values; Percent Errors for Sears using Embedding of 5

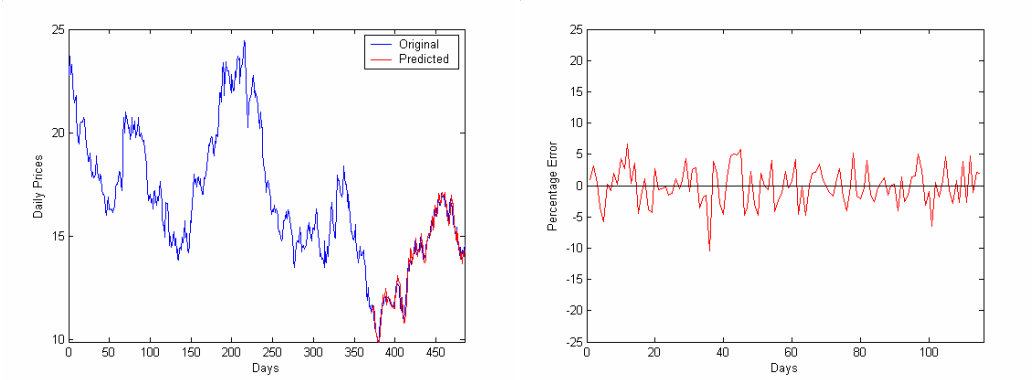



Figure 4.56: Plot of Predicted values; Percent Errors for Toys ‘R us using Embedding of 5

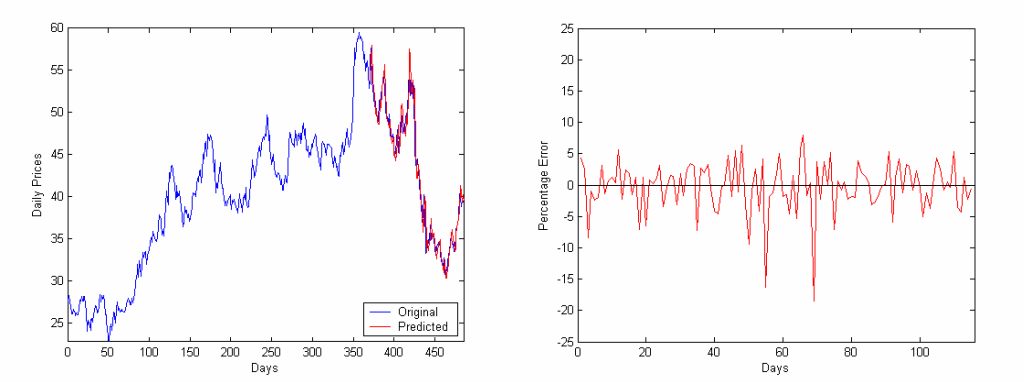



Figure 4.57: Plot of Predicted values; Percent Errors for Microsoft using Embedding of 5

## 4.9 Comparison of Results using SVR and SMA

In the preceding sections, we presented the individual results of using Support Vector Regression and Simple Moving Averages for five different stock prices. In this section we compare the results obtained using these two forecasting techniques.

### 4.9.1 RDP Transformation and Constant Embedding

First of all, we compared the results of using Support Vector Regression when the RDP transformation was used on each of the data series with that of Simple Moving Averages. Table 4.8 shows the results of using SVR and SMA with the Embedding size set to 5. From this Table, we see that for all except the Toys 'R us stock price and American Airlines, the directional symmetry using SVR are higher than that of Moving Averages. In addition, for all the stock prices, the DSR, MSE and MAPE are lower with SVR. Also for all stocks except Toys 'R us, the UPR values are lower using SVR than using SMA.

Table 4.8: Results of SVR and SMA using RDP Transformation

| Stocks | Method: Support Vector Regression<br>Transformation: RDP<br>Embedding Size: 5 | | | | | Method: Simple Moving Averages<br>Transformation: RDP<br>Embedding Size: 5 | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | DS (%) | DSR | UPR | MSE | MAPE | DS (%) | DSR | UPR | MSE | MAPE |
| American Airlines | 47.826 | 0.431 | 0.357 | 0.964 | 2.399 | 47.826 | 0.442 | 0.432 | 1.175 | 2.642 |
| McDonalds | 53.913 | 0.415 | 0.325 | 0.984 | 2.101 | 52.174 | 0.429 | 0.395 | 1.133 | 2.331 |
| Sears | 52.174 | 0.426 | 0.465 | 1.581 | 2.673 | 50.435 | 0.472 | 0.509 | 1.844 | 2.952 |
| Toys 'R us | 56.522 | 0.133 | 0.164 | 0.153 | 2.190 | 61.739 | 0.164 | 0.159 | 0.163 | 2.400 |
| Microsoft | 50 | 0.666 | 0.510 | 2.835 | 2.776 | 49.123 | 0.686 | 0.562 | 3.068 | 2.945 |

### 4.9.2 SVR with Z-Score Transformation and SMA

Table 4.9 and Table 4.10 show the results of using SVR when the Z-score transformation was applied to each of the data series and that of Simple Moving Averages (SMA). In this comparative analysis, we consider two variants of the SMA process where the Embedding size was varied and when it was set to 5. From Table 4.9, we see that for all the five stock prices, the SVR method using Z-score transformation out-performs the SMA on the basis of all the accuracy measures. The same inferences can be drawn from Table 4.10 which presents the results of the SVR with Z-score transformation and the results of the SMA using the optimal Embedding sizes as discussed in section 4.8.4.1.

Table 4.9: Results of SVR using Z-score Transformation, and SMA with Constant Embedding Size

| Stocks | Method: Support Vector Regression<br>Transformation: Z-Score<br>Embedding Size: 5 | | | | | Method: Simple Moving Averages<br>Transformation: RDP<br>Embedding Size: 5 | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | DS (%) | DSR | UPR | MSE | MAPE | DS (%) | DSR | UPR | MSE | MAPE |
| American Airlines | 74.783 | 0.263 | 0.233 | 0.444 | 1.499 | 47.826 | 0.442 | 0.432 | 1.175 | 2.642 |
| McDonalds | 73.043 | 0.331 | 0.221 | 0.649 | 1.563 | 52.174 | 0.429 | 0.395 | 1.133 | 2.331 |
| Sears | 76.316 | 0.338 | 0.406 | 1.472 | 2.243 | 50.435 | 0.472 | 0.509 | 1.844 | 2.952 |
| Toys 'R us | 88.696 | 0.047 | 0.077 | 0.025 | 0.909 | 61.739 | 0.164 | 0.159 | 0.163 | 2.400 |
| Microsoft | 77.193 | 0.524 | 0.314 | 1.806 | 1.981 | 49.123 | 0.686 | 0.562 | 3.068 | 2.945 |

Table 4.10: Results of SVR using Z-score Transformation, and SMA with Varying Embedding Sizes

| Stocks | Method: Support Vector Regression<br>Transformation: Z-score<br>Embedding Size: 5 | | | | | Method: Simple Moving Averages<br>Transformation: RDP<br>Embedding Size: Varying | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | DS (%) | DSR | UPR | MSE | MAPE | Emb | DS (%) | DSR | UPR | MSE | MAPE |
| American Airlines | 74.783 | 0.263 | 0.233 | 0.444 | 1.499 | 30 | 51.304 | 0.408 | 0.372 | 0.972 | 2.371 |
| McDonalds | 73.043 | 0.331 | 0.221 | 0.649 | 1.563 | 215 | 54.783 | 0.396 | 0.333 | 0.968 | 2.069 |
| Sears | 76.316 | 0.338 | 0.406 | 1.472 | 2.243 | 300 | 53.043 | 0.418 | 0.467 | 1.583 | 2.653 |
| Toys 'R us | 88.696 | 0.047 | 0.077 | 0.025 | 0.909 | 5 | 61.739 | 0.164 | 0.159 | 0.163 | 2.400 |
| Microsoft | 77.193 | 0.524 | 0.314 | 1.806 | 1.981 | 20 | 50 | 0.632 | 0.557 | 2.877 | 2.807 |

# Chapter 5

# Discussions and Conclusions

In this thesis, I have shown that the Support Vector Regression (SVR) is a more useful technique than the Simple Moving Averages (SMA) for forecasting volatile and non-stationary time series such as daily Stock Prices. I have also shown that the prediction performance of the SVR depends on the type of transformation that is applied to the series. For the time series considered in this study, we have found the Z-score transformation as producing the best results followed by the Relative Difference in Percentages. Using the Logarithmic transformation produced the worst results.

The superior performance of SVR over the SMA can be attributed to the following reasons;

1) SVR is based on the Structural Risk Minimization (SRM) principle which seeks to minimize an upper bound on the generalization error rather than the training error. This eventually leads to better prediction performance.
2) Unlike the SMA, global solutions are guaranteed with SVR. This is because implementing the SVR algorithm is equivalent to solving a linearly constrained

quadratic programming problem and the resulting solution is always unique, optimal and global.

3) The SVR works best for any type of data whether linear or non-linear. As discussed in the thesis, SVR is able to map a non-linear data into a higher dimensional feature space that is linear, using a kernel function.

Although the SMA using an Embedding size of 5 did not perform as good as SVR for all the transformations considered, the results of the SMA tend to improve when optimal embedding sizes were selected by varying the Embedding values from 5 to 300. The inference that can be deduced from this is that analysts of financial data who use SMA for predictions should not employ fixed Embedding sizes for the different data series that they handle but should seek the Embedding size that works best from each series.

## 5.1 Recommendations and Future Work

In this study, the cross-validation approach used for determining the optimal values of the SVR parameters was time-consuming and expensive. Hence a future study that will investigate simple and cheaper ways of obtaining the parameters is recommended. Other forecasting techniques that are more suitable for modeling volatile series can be employed for the same set of data used here and the results compared to what we obtained using SVR. A possible extension of this work is the development of a data dependent weighting function for computing SVR parameters without increasing the number of parameters required.

# LIST OF REFERENCES

# List of References


1. Abecasis, S.M., and Lapenta, E.S., (1996). Non-stationary time series forecasting within a neural network framework, NeuroVest Journal, Vol**. 4**, No. 4: 9 - 16

2. Abecasis, S.M., Lapenta, E.S. and Pedreira, C.E., (1999). Performance metrics for financial time series forecasting, Journal of computational intelligence in finance, Vol. **7**, No. 4: 5 - 23

3. Aoki, M., (1990). State space modeling of time series. Springer-Verlag; New York: Springer-Verlag, 2nd edition.

4. Azoff, M.E., (1994). Neural Network Time Series Forecasting of Financial Markets, John Wiley and Sons Ltd., Chichester, England. (Reprinted May 1995)

5. Baestaens, D.E., (1994). Neural Network Solutions for Trading in Financial Markets, London: Financial Times: Pitman Pub.

6. Belousov, A.I., Verzakov, S.A., and Frese, J.V., (2002). A flexible classification approach with optimal generalization performance: support vector machines, Chemometrics and Intelligent Laboratory systems Vol. **64**: 15 - 25

7. Bennett, K., and Bredensteiner, E.J., (2000). Duality and geometry in SVM classifiers, In Langley, P., editor, Proc. of Seventeenth Intl. Conf. on Machine Learning, pages 57 - 64, San Francisco, Morgan Kaufmann.

8. Bollerslev, T., (1986). Generalized autoregressive conditional heteroskedasticity, Econometrics, Vol. **31**: 307 - 327.

9. Boser, B.E., Guyon, I., and Vapnik, V.N., (1992). A training algorithm for optimal margin classifiers, In Computational Learning Theory, pages 144 - 152.

10. Box, G.E.P. and Jenkins, G.M., (1976). Time Series Analysis, Forecasting and Control. Holden-Day, Oakland, CA.

11. Box, G.E.P. and Jenkins, G.M., (1994). Time Series Analysis, Forecasting and Control. San Francisco: Holden-Day, third edition.

12. Brown, R. G., (1963). Smoothing, Forecasting and Prediction of Discrete Time Series. Englewood Cliffs, N.J.: Prentice Hall.

13. Burges, C.J.C., (1998). A Tutorial on Support Vector Machines for Pattern Recognition, Data Mining and Knowledge Discovery Vol. **2**:121 - 167

14. Burges, C. and Crisp, D., (2000). Uniqueness of the SVM Solution, In Solla, S.A., Leen, T.K., and Muller, K.-R., editors, Advances in Neural Information Processing Systems, Vol. **12**: 223 - 229, Cambridge, MA, MIT Press.

15. Campbell, J.Y., Lo, A.W., and MackinLay, A.C., (1997). The Econometrics of Financial Markets, Princeton University Press, New Jersey.

16. Cao, L., and Tay, F.E.H., (2001). Financial Forecasting Using Support Vector Machines, Neural Computing & Applications Vol. **10**: 184 – 192.

17. Cao, L., (2002). Support vector machines experts for time series forecasting, NeuroComputing Vol. **51**: 321 - 339.

18. Cao, L., and Gu, Q., (2002). Dynamic support vector machines for non-stationary time series forecasting, Intelligent Data Analysis Vol. **6**: 67 - 83.

19. Cao, L.J., and Tay, F.E.H., (2003). Support vector machine with adaptive parameters in financial time series forecasting, IEEE Transactions on Neural Networks, Vol. **14**, No. 6: 1506 - 1518

20. Cao, L.J., Chua, K.S., and Guan, L.K., (2003). c-Ascending Support Vector Machines for Financial Time Series Forecasting, IEEE 317 - 323

21. Chatfield, C., (1997). Forecasting in the 1990s, The Statistician., Vol. **46**, No. 4: 461 - 473.

22. Chatfield, C., (2001). Time Series Forecasting. Chapman and Hall.

23. Chatfield, C., (2004). The analysis of time series: an introduction. Chapman and Hall, Sixth edition.

24. Cheadle, C., Vawter, M.P., Freed, W.J., and Becker, K.G., (2003). Analysis of Microarray Data using Z Score Transformation, Journal of Molecular Diagnostics, Vol. **5**, No. 2: 1525 - 1578.

25. Cheng, B., and Titterington, D.M., (1994). Neural Networks: A Review from a Statistical Perspective. Statistical Science, Vol. **9**: 2 - 54.

26. CherKassky, V., and Mulier, F., (1998). Learning from Data: Concepts, Theory and Methods. John Wiley & Sons Inc. New York, NY.

27. Cherkassky, V., and Ma, Y., (2004). Practical selection of SVM parameters and noise estimation for SVM regression, Neural Networks, Vol. **17,** No. 1: 113 - 126.

28. Chih-Chung, C., and Chih-Jen, L., (2001). LIBSVM: a Library for Support Vector Machines (Version 2.31).

29. Cortes C., and Vapnik, V.N. (1995). Support-Vector Networks, Machine Learning, Vol. **20,** No. 3: 273 - 297.

30. Cronbach, L. J., (1987). Statistical tests for moderator variables: Flaws in analyses recently proposed. Psychological Bulletin, Vol. **102**: 414 - 417.

31. Delurgio, Stephen A., (1998). Forecasting principles and applications, Irwin/McGraw Hill, Boston.

32. Dionísio, A., Menezes, R., and Mendes, D.A., (2005). Uncertainty analysis in financial markets: can entropy be a solution? http://www.american.edu/cas/econ/faculty/golan/Papers/Papers05/DionisioMenezesMendesPaper.pdf

33. Dorsey, R., and Randall, S., (1998). The use of parsimonious neural networks for forecasting financial time series, Journal of Computational Intelligence in Finance, Vol. **6**. No. 1: 24 - 31.

34. Duda, R.O., and Hart, P.E., (1973). Pattern classification and scene analysis. Wiley Interscience, New York, New York.

35. Engle, R.F., (1982). Autoregresssive Conditional Heteroscedasticity with Estimates of the Variance of United Kingdom Inflation, Econometrica, Vol. **50**, No. 4: 987 - 1007.

36. Engle, R.F., (1995), editor. ARCH: selected readings. Oxford; New York: Oxford University Press.

37. Fama, E.F., (1965). The Behavior of Stock-Market Prices. Journal of Business, Vol. **38**, No. 1: 34 - 105.

38. Fama, E.F., (1970). Efficient Capital Markets: A Review of Theory and Empirical Work, Journal of Finance, Vol. **25**, No. 2: 383 - 417.

39. Fernandez, R., (1999). Predicting time series with a local support vector regression machine. In Advanced Course on Artificial Intelligence (ACAI '99). Available on-line at: http://www.iit.demokritos.gr/skel/eetn/acai99/ (Downloaded on January 15, 2005).

40. Gourieroux, C., (1997). ARCH Models and Financial Applications. Springer-Verlag.

41. Granger, C.W.J., and Andersen, A.P., (1978). An Introduction to Bilinear Time Series Models. Gottingen Vandenhoeck and Ruprecht.

42. Granger, C.W.J., and Joyeux, R., (1980). An introduction to long-memory time series models and fractional differencing, Journal of Time Series Analysis, vol. **1**.

43. Gunn, S. R., (1998). Support vector machines for classification and regression, *Technical Report*, Image Speech and Intelligent Systems Research Group, University of Southampton, UK.
Available at *http://www.isis.ecs.soton.ac.uk/isystems/kernel/*.

44. Harvey, A.C., (1989). Forecasting, Structural Time Series Models and the Kalman Filter. Cambridge: Cambridge Univ. Press.

45. Haykin, S., (1999). Neural Networks: A Comprehensive Foundation. Upper Saddle River, N.J.: Prentice Hall, 2nd edition.

46. Harvey, A.C., (1989). Forecasting, Structural Time Series Models and the Kalman Filter. Cambridge: Cambridge Univ. Press.

47. Haykin, S., (1999). Neural Networks: A Comprehensive Foundation. Upper Saddle River, N.J.: Prentice Hall, 2nd edition.

48. Jain, A.K., Murty, M.N., and Flynn, P.J., (1999). Data Clustering: a review, ACM Computing Surveys, Vol. **31**, No. 3: 264 - 323.

49. Joachims, T., (1997). Text Categorization with Support Vector Machines, Technical Report

50. Joachims, T., (1998). Text Categorization with Support Vector Machines: Learning with Many Relevant Features, In Claire Nedellec and Celine Rouveirol, editors, Proceedings of ECML-98, 10th European Conference on Machine Learning, pages 137 - 142, Chemnitz, DE, Springer Verlag, Heidelberg, DE.

51. Kim, K., (2003). Financial time series forecasting using support vector machines, Neurocomputing, Vol. **55**: 307 - 319

52. Kwok, J.T., and Tsang I.W., (2003). Linear dependency between $\varepsilon$ and the input noise in $\varepsilon$-support vector regression. IEEE Transactions on Neural Networks, Vol. XX, No. YY: 1 - 8

53. Lee, K.L., and Billings, S.A., (2002). Time series prediction using support vector machines, the orthogonal and the regularized orthogonal least-squares algorithms, International Journal of Systems Science Vol. **33**: 811 - 821

54. Lin, H.T., and Lin, C.J., (2003). A study on sigmoid kernels for SVM and the training of non-PSD kernels by SMO-type methods, Technical Report. Available: http://www.csie.ntu.edu.tw/~cjlin/papers/tanh.pdf

55. Malkiel, B.G., (1987). Efficient Market Hypothesis. Macmillan, London.

56. Mattera, D., and Haykin, S., (1999). Support vector machines for dynamic reconstruction of a chaotic system. In B. Schölkopf, J. Burges, A. Smola (Eds), *Advances in Kernel Methods: Support Vector Machine*. Cambridge, MA: MIT Press.

57. Mukherjee, S., Osuna E., and Girosi, F., (1997). Nonlinear prediction of chaotic time series using support vector machines, Proc. of IEEE NNSP'97, Amelia Island, FL.

58. Muller, K.R, Smola, A., Ratsch, G., Schokolpf, B., Kohlmorgen, J., Vapnik, V., (1997). Predicting time series with support vector machines, in: Gerstner, W., Germond, A., Hasler, M., Nicoud, J.-D., (Eds.), Proceedings of ICANN '97, Springer LNCS 1327, Berlin, pp. 999-1004.

59. Muller, K.R, Smola, A.J., Ratsch, G., Schokolpf B., and Kohlmorgen, J., (1999). Using support vector machines for time series prediction, in: Advances in Kernel Methods – Support Vector Learning, Schokolpf, B., Burges, C.J.C., and Smola, A.J., eds, pp. 243-254.

60. Nelson, H.L. Jr., and Granger, C.W.J., (1979). Experience with Using the Box-Cox Transformation When Forecasting Economic Time Series, Journal of Econometrics Vol. **10**: 57 - 69.

61. Nicholls, D.F., and Pagan, A., (1985). Varying Coefficient Eegression. In E.J. Hannan, P.R. Krishnaiah, and M.M. Rao, editors, Handbook of Statistics, Vol. **5**: 413 - 449, North Holland, Amsterdam.

62. Nogales F.J., Contreras J., Conejo A.J., and Espínola R., (2002). Forecasting next-day electricity prices by time series models, IEEE Transactions on Power Systems Vol. **17**, No. 2: 342 - 348

63. Osuna, E., Freund, R., and Girosi, F., (1997). Support Vector Machines: Training and Applications, Technical Report AIM-1602, MIT.

64. Ozgur, T., (2004). Predicting financial performance of publicly traded Turkish firms: A comparative study, Unpublished.

65. Pantazopoulos, K.N., Tsoukalas, L.H., Bourbakis, N.G., Br¨un, M.J., and Houstis, E.N., (1998). Financial prediction and trading strategies using neurofuzzy

approaches, IEEE Transactions On Systems, Man, and Cybernetics – Part B: Cybernetics, Vol. **28**, No. 4: 520 – 531

66. Pawelzik, K., Muller, K.R., and Kohlmorgen, J., (1996). Annealed competition of experts for a segmentation and classification of switching dynamics, Neural Computation Vol. **8**: 340 - 356.

67. Priestley, M.B., (1981). Spectral Analysis and Time Series, New York: Academic Press, London.

68. Poon, S., and Granger C.W.J., (2003). Forecasting volatility in financial markets: A review, Journal of Economic Literature, Vol. **41,** No. 2: 478 - 539

69. Quinlan, J. R., (1986). Induction of Decision Trees, Machine Learning, Vol. **1**: 81 - 106.

70. Raj, B., and Ullah, A., (1981). Econometrics: a varying coefficients approach. New York: St. Martin's Press, 2nd edition.

71. Ripley, B.D., (1993). Statistical aspects of neural networks, In Barndor-Nielsen, O.E., Jensen, J.L., and Kendall, W.S., editors, Network and Chaos – Statistical and Probabilistic Aspects, pages 40-123, London, Chapman and Hall.

72. Sanders, N.R., and Manrodt, K.B., (1994). Forecasting practices in US corporations: survey results, Interfaces Vol. **24**, No. 2: 92 - 100.

73. Schmidt, M., (1996). Identifying speaker with support vector networks, Interface 1996 Proceedings, Sydney.

74. Scholkopf, B., Burges, C., and Vapnik, V., (1995). Extracting support data for a given task, In Fayyad, U.M., Uthurusamy, R. (Eds.), Proceedings of the First International Conference on Knowledge Discovery & Data Mining, AAAI Press, Menlo Park, CA.

75. Scholkopf, B., Burges, C., and Smola, A., (1999). Advances in Kernel Methods: Support Vector Learning, MIT Press, Cambridge, Massachusetts.

76. Schölkopf, B., and Smola, A., (2002). Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond. Cambridge, MA: MIT Press.

77. Smith, K.W., and Sasaki, M.S., (1979). Decreasing Multicollinearity: A Method for Models with Multiplicative Functions, Sociological Methods and Research, Vol. **8**: 35 - 36.

78. Smola, A.J., and Scholkopf, B., (1998). A tutorial on support vector regression, NeuroCOLT Technical Report NC-TR-98-030, Royal Holloway College, University of London, UK.

79. Smola, A.J., Murata, N., Schölkopf, B., and Muller, K. (1998). Asymptotically optimal choice of $\varepsilon$-loss for support vector machines, In Niklasson L., Boden M., and Ziemke T., (Eds.), Proceedings of the International Conference on Artificial Neural Networks (ICANN 1998), Perspectives in Neural Computing, Berlin: Springer, 105 - 110

80. Smola, A.J., and Schölkopf, B., (2004). A tutorial on support vector regression, Statistics and Computing, Asymptotically optimal choice of $\varepsilon$-loss for support vector machines Vol. **14**: 199 - 222

81. Stickel, S.E., (1990). Predicting Individual Analyst Earnings Forecasts, Journal of Accounting Research, Vol. **28**, No. 2: 409 - 417

82. Stitson, M.O., Weston, J.A.E, Gammerman, A., Vork, V., and Vapnik, V., (1996). Theory of support vector machines, Technical report CSD-TR-96-17

83. Tashman, L.J., (2000). Out-of-sample tests of forecasting accuracy: An analysis and review. International Journal of Forecasting, Vol. **16**: 437 - 450

84. Tay, F.E.H., and Cao, L.J., (2001). Application of support vector machines in financial time series forecasting, Omega Vol. **29**: 309 - 317

85. Tay, F.E.H., Cao, L.J., (2002). Modified support vector machines in financial time series forecasting, Neurocomputing Vol. **48**: 847 - 861

86. Tay, F.E.H., Cao, L.J., (2002). $\varepsilon$-Descending Support Vector Machines for Financial Time Series Forecasting, Neural Processing Letters Vol. **15**:179 - 195

87. Thissen, U., Brakel, R.V., de Weijer, A.P., Melssen, W.J., Buydens L.M.C., (2003). Using support vector machines for time series prediction, Chemometrics and Intelligent Laboratory Systems Vol. **69,** 35 - 49

88. Thomason, M., and Caldwell, R., (1998). The practitioner methods, Journal of Computational Intelligence in Finance, Vol. **6**, No.1, 42 - 47

89. Thomason, M., (1999). The practitioner methods, Journal of Computational Intelligence in Finance, Vol. **7**, No. 3, 36 - 45

90. Tong, H., (1990). Non-Linear Time Series. Clarendon Press, Oxford.

91. Tsay R.S., (2002). Analysis of Financial Time Series, Wiley InterScience, John Wiley and Sons, Inc.

92. Tsibouris, G., and Zeidenberg, M., (1995). Testing the efficient markets hypothesis with gradient descent algorithms. In Refenes, A., editor, Neural Networks in the Capital Markets, John Wiley and Sons.

93. Van Eyden, R.J., (1997). The Application of Neural Networks in the Forecasting of Share Prices, Finance and Technology Publishing, Haymarket, VA, USA.

94. Vapnik, V.N., (1979). Estimation of Dependencies Based on Empirical Data (in Russia). Nauka, Moscow.

95. Vapnik, V.N., (1995). The Nature of Statistical Learning Theory. New York, Springer-Verlag.

96. Vapnik, V.N, Golowich, S.E, and Smola, A.J., (1997). Support vector method for function approximation, regression estimation, and signal processing, Advances in Neural Information Processing Systems, Vol. **9**:281 – 287.

97. Vapnik, V.N., (1998). Statistical Learning Theory, Wiley, New York.

98. Vapnik, V., (2000). Support-vector networks, Machine Learning, Vol. **20**, No. 3: 273 - 297.

99. Watkins, C.W., (1989). Learning from Delayed Rewards. PhD Thesis; King's College, Cambridge, England.

100. West, M., and Harrison, J., (1997). Bayesian Forecasting and Dynamic Models. New York: Springer-Verlag.

101. Yang, H., King, I., and Chan, L., (2002a). Non-fixed and asymmetrical margin approach to stock market prediction using Support Vector Regression, In the Proceedings of ICONIP 2002, Singapore.

102. Yang, H., Chan, L., and King, I., (2002b). Support vector machine regression for volatile stock market prediction, Lecture notes in Computer Science, Vol. **2412**: 391 - 396

103. Yi, Y., (1989). On die evaluation of main effects in multiplicative regression models, Journal of the Marketing Research Society, Vol. **31**: 133 - 138.

# VITA

Bukola Titilayo Ojemakinde received a B.S. degree in Industrial and Production Engineering from University of Ibadan, Nigeria in 2000. She won several awards for academic excellence during her under-graduate program including the "Best Graduating Student" award for the College of Engineering. After her B.S., She worked for one year at Schlumberger Wire-line and Testing, Nigeria between 2000 and 2001. She also worked for about two years at Citibank Nigeria (now Nigeria International Bank) between November 2001 and August 2004.

She started a Masters Program in Industrial and Information Engineering at the University of Tennessee, Knoxville in 2004. During her Masters Program, she worked as a graduate assistant for the department of Industrial Engineering and provided assistance in teaching two undergraduate courses - Engineering Economic Analyses and Operations Research, and one graduate course – Statistical Methods in Industrial Engineering.

Bukola is a member of two professional bodies which are the Institute for Operations Research and Management Science (INFORMS) and the Institute of Industrial Engineering (IIE).

Bukola is married to her college heartthrob, Olatokunbo David Ojemakinde, the man she calls "the love of her life".