FORECASTING STOCK INDEX MOVEMENT: A COMPARISION OF SUPPORT VECTOR MACHINES AND RANDOM FOREST

Manish Kumar, Indian Institute of Technology Madras, India
Email: ms04d001@iitm.ac.in
M. Thenmozhi, Indian Institute of Technology Madras, India
Email: mtm_iitm@yahoo.com

ABSTRACT

There exists vast research articles which predict the stock market as well pricing of stock index financial instruments but most of the proposed models focus on the accurate forecasting of the levels (i.e. value) of the underlying stock index. There is a lack of studies examining the predictability of the direction / sign of stock index movement. Given the notion that a prediction with little forecast error does not necessarily translate into capital gain, this study is an attempt to predict the direction of S&P CNX NIFTY Market Index of the National Stock Exchange, one of the fastest growing financial exchanges in developing Asian countries. Random forest and Support Vector Machines (SVM) are very specific type of machine learning method, and are promising tools for the prediction of financial time series. The tested classification models, which predict direction, include linear discriminant analysis, logit, artificial neural network, random forest and SVM. Empirical experimentation suggests that the SVM outperforms the other classification methods in terms of predicting the direction of the stock market movement and random forest method outperforms neural network, discriminant analysis and logit model used in this study.

Keywords: Support vector machine; Random forest; Forecasting; Stock index

1. INTRODUCTION

The prediction of financial market is a very complex task, because the financial time series are inherently noisy, non-stationary, and deterministically chaotic. The noisy characteristic refers to the unavailability of complete information from the past behavior of financial markets to fully capture the dependency between future and past prices. The information that is not included in the model is considered as noise. The non-stationary characteristic implies that the distribution of financial time series is changing over time. By deterministically chaotic, one means that financial time series are short-term random but long-term deterministic. Many factors and unexpected events or incidents like economic or political situation, trader' s expectations, catastrophe or war may cause the change of a financial time series such as stock market index and exchange rates. At the same time the relationship of any financial time series with the other related data series may also change with time. Therefore, predicting the financial market' s movements is quite difficult.

Trading in stock market indices has gained popularity in major financial markets around the world. Accurate predictions of stock market indexes are important for many reasons. Chief among these are the need for the investors to hedge against potential market risks, and opportunities for speculators and arbitrageurs to make profit by trading in stock index. Clearly, being able to accurately forecast the stock market index has profound implications and significance for both researchers and practitioners.

There exist vast literatures which concentrate on the predictability of stock market return. In almost all cases, the performance metrics and the acceptability of the proposed models are measured by the deviations of forecast value from the actual values. Different dealers, investors, and other market players adopt different trading strategies, so the forecasting models which rank first in terms minimization of forecast error may not be suitable to meet the expectation of the dealers or investors. It is because the trading driven by a particular forecasting model with a minimal forecast error may not be profitable as trading guided by an accurate prediction of the direction of movement of stock index. Therefore it is very important to forecast the direction of movement of stock index for developing effective market trading strategies (Leung *et al.* 2000).

In recent years, there have been a growing number of studies looking at the direction or trend of movements of financial markets (such as Maberly, 1986; Wu and Zhang, 1997; O' Connor, Remus and Griggs, 1997). However, none of these studies provide a comparative evaluation of different classification techniques regarding their ability to predict the sign of the index return. Although there exists some articles addressing the issue of forecasting financial time series such as stock market index most of the empirical findings are associated with the developed financial markets (UK, USA, and Japan). However, few researches exist in the literature to predict direction of stock market index movement in emerging markets. Nowadays, many international investment bankers and brokerage firms have major stakes in overseas markets. Harvey (1995) found emerging market returns are more likely to be influenced by local information than developed markets; in fact, emerging market returns are generally more predictable than developed market returns. Indian stock markets have received relatively little attention until recently. Now there is more interest and research on Indian market data due to the country' s rapid growth and potential opportunities for investors. Since the establishment of National Stock Exchange (NSE), the financial markets in this Asian country have attracted considerable global investments.

Realizing the growing importance of Indian stock market and given the notion that the studies related to the comparative evaluation of different classification techniques seems to be absent, this study attempts to develop various forecasting model based on classification technique.

In recent years, artificial neural networks (ANN) have been successfully used for modeling financial time series (Cheng *et al.* (1996), Sharda and Patil (1994), Van and Robert (1997). Some of these studies, however, showed that ANN had some limitations in learning the patterns because stock market data has tremendous noise and complex dimensionality. ANN often exhibits inconsistent and unpredictable performance on noisy data. Recently, a novel algorithm called Support Vector Machines (SVM) has been used for forecasting financial time series (Mukherjee *et al.* (1997), Tay and Cao (2001), Kim (2003). Another technique called Random Forest method have been found to perform better in many studies like Creamer and Freund (2004) and Lariviere and Poel (2004). SVM technique was developed by Vapnik and

colleague (1997) and Random forest method was developed by Breiman (2001). These machine learning methods such as random forest and support vector machines have been successful because it avoid the question of modeling the underlying distribution and focus on making accurate predictions for some variables given others variables. In addition, the solution of SVM and random forest may be global optimum while other neural network models may tend to fall into a local optimal solution. Thus, over fitting is unlikely to occur with SVM and random forest regression. Though, these models have been experimented in the research work, no attempt has been made to compare the performance of these models for predicting the direction of financial time series like stock market. Moreover, attempt to use random forest method has not been examined for predictability of stock market time series.

Hence, the present study predicts the direction of the movement of S&P CNX NIFTY Index using SVM and random forest regression and the performance of the SVM and random forest regression are compared by benchmarking their results against traditional classification techniques like discriminant analysis, logit model and artificial intelligence techniques like neural network. The major contributions of this study are: (1) to demonstrate and verify the predictability of stock index direction using SVM and random forest. (2) to compare the performance of SVM and random forest in terms of predicting the direction of the stock market movement.

The remaining portion of this paper is organized as follows. A literature review of various classification techniques related to time series forecasting is given in the next section. The data pertaining to the same, the novel SVM, random forest regression and the benchmark models are introduced in section 3. Empirical results from the real data sets are reported in section 4. Finally, section 5 contains the concluding remarks.

2. LITERATURE REVIEW

2.1 Evidence of stock market predictability

There exists considerable evidence showing that stock returns are to some extent predictable. Most of the research is conducted using data from well established stock markets such as the US, Western Europe, and Japan. It is, thus, of interest to study the extent of stock market predictability using data from less well established stock markets such as that of India.

For the US, several studies examine the cross-sectional relationship between stock returns and fundamental variables. Variables such as earnings yield, cash flow yield, book-to-market ratio, and size are shown to have some power predicting stock returns. Banz and Breen (1986), Jaffee *et al.* (1989), and Fama and French (1992) are good examples of this group of research. Further, studies based on European markets report similar findings. The results of Ferson and Harvey (1993) indicate that returns are, to a certain extent, predictable across a number of European markets (e.g., UK, France, Germany). In their study which is aimed at forecasting the UK stock prices, Jung and Boyd (1996) report "reasonably good" performance of their forecasts, suggesting that the predictive strength of their stock return models are not negligible. For the Japanese stock market, the empirical investigations by Jaffe and Westerfield (1985) and Kato *et al.* (1990) also find some evidence of predictability in the behavior of index returns.

Using time-series analysis, Fama and Schwert (1977), Rozeff (1984), Keim and Stambaugh (1986), Campbell (1987), Fama and Bliss (1987), and Fama and French (1988, 1988, 1990) found out that macroeconomic variables such as short-term interest rates, expected inflation, dividend yields, yield spreads between long- and short-term government bonds, yield spreads between low- and high-grade bonds, lagged price–earnings ratios, and lagged returns have some power to predict stock returns.

Refenes *et al*. (1994) also indicated that traditional statistical techniques for forecasting have serious limitations with respect to applications with nonlinearities in the data set such as stock indices. In the last decade, applications associated with artificial neural network (ANN) have been gaining popularity in both academic and corporate research. Many studies have shown that neural networks can be one of the very useful tools in time series forecasting. A number of studies (Refenes *et al.* (1987), Kimoto *et al*. (1990), Takashi *et al*. (1990), Kryzanowski *et al.* (1993), McCluskey (1993)**,** Bansal and Vishwanatahn (1993), Refenes (1994), Donaldson and Kamstra (1996), Zirilli (1997)) have investigated neural network model for

predicting the stock market and the results support the importance of the model. Artificial neural networks, thus can serve as a better prediction model that can overcome many of the drawbacks associated with the traditional techniques.

The use of technical indicators as explanatory variables to predict the stock return has increased tremendously in the recent researches. The basic assumption behind that is there are recurring patterns in the market behavior that are predictable. It attempts to use past stock price and volume information to predict future price movements. In most cases, there are five time series for a single share or market index. These five series are open price, close price, highest price, lowest price and trading volume. Analysts monitor changes of these numbers to decide their trading. As long as past stock prices and trading volumes are not fully discounted by the market, technical analysis has its value on forecasting.

To maximize profits from the stock market, more and more "best" forecasting techniques are used by different traders. Nowadays, traders no longer rely on a single technique to provide information about the future of the markets but rather use a variety of techniques to obtain multiple signals. Neural networks are often trained by technical indicators to produce trading signals. Yao *et al*. (1999) used a technical method which takes not only the delayed time series data as inputs but also the technical indicators. The technical indicators used are moving average (MA), momentum (M), Relative Strength Index (RSI) and stochastics (%K), and moving average of stochastics (%D). Neural network were trained based on these technical indicators to forecast Kuala Lumpur Composite Index (KLCI). Chakraborty *et al*. (1992) and Refenes *et al*. (1993), Kimoto et al. (1990), Barr and Mani (1994) and Grudnitski and Osborn (1993) in their studies used historic data to train the neural network, which, consists of the moving average values. The aim is to discern some relationship between past and present behaviour, indicating some form of structure. Kamruzzaman and Sarkar (2004) used five technical indicators to predict six currency rates against Australian dollar using neural networks. The indicators are MA5, MA10, MA20, MA60, MA120 and X_i, namely, moving average of one week, two weeks, one month, one quarter, half year and last week's closing rate, respectively. Man-chung *et al.* used various technical indicators to gain insight into the direction that the Shanghai's Stock Exchange market may be going. Ten technical indicators was selected as inputs of the neural network: the lagging input of past 5 days' change in exponential moving average, relative strength index on day *t*-1, moving average convergence-divergence on day *t*-1, MACD Signal Line on day *t*-1, stochastic %K on day *t*-1, and stochastic %D on day *t*-1.

It is a well-established practice in the recent empirical finance literature to account for the predictability of stock market given the information set by using technical indicators. Stock returns predictability given aggregate variables in the technical indicators information set is a well-accepted fact. The question that remains is how to use the information set in an optimal way for forecasting and trading.

2.2. *Forecasting the direction of stock market*

Most trading practices adopted by financial analysts rely on accurate prediction of the price levels of financial instruments. However, some recent studies have suggested that trading strategies guided by forecasts on the direction of price change may be more effective and generate higher profits. Wu and Zhang (1997) investigate the predictability of the direction of change in the future spot exchange rate. Based on the S&P 500 futures, Maberly (1986) explores the relationship between the direction of interday and intraday price change. Finally, in their study on the All Ordinaries Index futures traded at the Australian Associated Stock Exchanges, Hodgson and Nicholls (1991) suggest to hold an evaluation of the economic significance of the direction of price changes in future research.

In summary, the findings in these studies are reasonable because accurate point estimation, as judged by its deviation from the actual observation, may not be a good predictor of the direction of change in the instrument's price level. Also, predicting the direction is a practical issue which usually affects a financial trader's decision to buy or sell an instrument.

2.3 Application of support vector machines and random forest in finance

The neural network has been widely used in the area of financial time series forecasting because of its broad applicability to many business problems and preeminent learning ability. However, the neural network has many disadvantages including the need for the determination of the value of controlling parameters and the number of processing elements in the layer, and the danger of over fitting problem.

Recently, a support vector machine (SVM), and random forest regression based machine learning methods have been used in finance. There are few studies for the application of SVM and random forest regression in financial time-series forecasting. Mukherjee *et al*. (1997) showed the applicability of SVM to time-series forecasting. Recently, Tay and Cao (2001) examined the predictability of financial time-series including five time series data with SVMs. The objective of this paper was to examine the feasibility of SVM in financial time series forecasting by comparing it with a multi-layer back-propagation (BP) neural network. The experiment shows that SVM outperforms the BP neural network based on the criteria of normalized mean square error (NMSE), mean absolute error (MAE), directional symmetry (DS) and weighted directional symmetry (WDS). Kim (2003) used SVM to forecast the direction of daily price change in the daily Korea composite stock price index (KOSPI). This study selects 12 technical indicators to make up the initial attributes. In addition, this study examines the feasibility of applying SVM in financial forecasting by comparing it with back-propagation neural networks and case-based reasoning. Analysis of the experimental results proved that it is advantageous to apply SVMs to forecast financial time series.

Creamer and Freund (2004) in their study used random forest regression technique for predicting performance and quantifying corporate governance risk in the case of Latin American markets. They conduct tenfold cross-validation experiments on one sample of Latin American Depository Receipts (ADRs), and on another sample of Latin American banks. They compared their results of random forest with logistic regression. Results were found in favor of random forest regression. Lariviere and Poel (2004) used random forest regression technique for investigating both customer retention and profitability outcomes. By means of random forests techniques they investigated a broad set of explanatory variables, including past customer behavior, observed customer heterogeneity and some typical variables related to intermediaries. The authors analyzed a real-life sample of 100,000 customers taken from the data warehouse of a large European financial services company. The research findings demonstrate that random forests techniques provide better fit for the estimation and validation sample compared to ordinary linear regression and logistic regression models.

With the introduction of SVM and random forest regression, these techniques have been used to solve estimation problems and they have been shown to exhibit excellent performance. Thus it appears that the SVM and random forest regression techniques can improve the predictability of S&P CNX NIFTY Index direction movement considering the complex nature of the emerging stock market. The objective of this study, as in all previous research, will focus on the out-of-sample performance of the machine learning techniques (SVM and random forest regression), neural network model, discriminant analysis and logistic regression in forecasting the direction of movement of S&P CNX NIFTY Index. Thus the major contribution of this study will be (1) to demonstrate and verify the predictability of S&P CNX NIFTY Index direction movement by applying the SVM and random forest regression model; (2) to find out the appropriate neural network, disriminant analysis and logistic model for NIFTY index series; (3) to compare the performance of the SVM and random forest regression model with that of neural network model, and other statistical techniques in terms of hit ratio.

The study is not based on any underlying economic behavior. The study uses models, which typically relate a dependent variable with the various technical indicators. The models developed will be especially suitable for short term forecasting of the direction of the stock market movement. The focus of this study is to give the fund manager, traders, borrowers, and treasurer' s better planning abilities by predicting the direction of future stock market movement patterns.

3. DATA AND METHODOLOGY

The research data used in this study is technical indicators and the direction of change in the daily closing prices for the S&P CNX NIFTY Index. The series spans from 1st January 2000 to 31st May 2005 totaling 1,360 trading days. The data is divided into two periods- the first period runs from 1st January, 2000, to 23rd February, 2004, which is used for model estimation and is classified as in-sample, while the second period runs from 24th February, 2004 to 31st May, 2005, is reserved for out-of-sample forecasting and evaluation. The division amounts to approximately 25 per cent being retained for out-of-sample purposes. Since this study attempt to forecast the direction of daily price change in the stock price index, technical indicators are used as input variables. This study selects 12 technical indicators to make up the initial attributes, as determined by prior research, Kim and Han (2000) and Kim (2003). The descriptions of initially selected attributes are presented in Appendix 1.

The purpose of this study is to predict the directions of daily change of the S&P CNX NIFTY Index. Direction is a categorical variable to indicate the movement direction of S&P CNX NIFTY Index at any time t. They are categorized as "0" or "1" in the research data. "0" means that the next day' s index is lower than today' s index, and "1" means that the next day' s index is higher than today' s index. The original data are scaled into the range of [1.0; 1.0] using max-min normalization formula, which is given by:

$$v' = \frac{v - \min A}{\max A - \min A}(new\_\max A - new\_\min A) + new\_\min A$$

Where min A, max A, is the minimum and maximum values of the attribute A. max-min normalization maps a value v of A to v' in the range {new_min A, new_max A} i.e. (-1.0; 1.0), in this case. The goal of linear scaling is to independently normalize each feature component to the specified range. It ensures the larger value input attributes do not overwhelm smaller value inputs, and then helps to reduce prediction errors.

The prediction performance P is evaluated using the following equation:

$$P = \frac{1}{m}\sum_{i=1}^{m} R_i$$

Where R_i is the prediction result for the ith trading day is defined by

$$R_i = \begin{cases} 1 \text{ if } PO_i = AO_i \\ 0 \text{ otherwise} \end{cases}$$

Where PO_i is the predicted output from the model for the ith trading day, and AO_i is the actual output for the ith trading day, m is the number of the test examples.

3.1 Forecasting by classification models

In this section, a brief summary of the classification models used in this comparative study are discussed. Although these models are based on different statistical techniques, they share a common trait - the ability to generate the probability of group membership. In other words, these models are able to estimate the probability of an upward (or downward) movement in the stock index.

Linear Disrciminant Analysis

Discriminant analysis is a multivariate statistical technique that investigates the differences between two or more groups of observations with respect to a set of independent (input) variables. These independent variables, called discriminator variables, are used to distinguish the characteristics among different groups. In the discriminant analysis, these discriminator variables are combined to form a set of mathematical equations, known as the classification functions. There is a classification function for each group of observations. The classification function which yields the highest *Z* score indicates the group membership of the input vector to be classified.

Logit Model

Logit models are appropriate to use when trying to model dependent variables that can take on only binary values (e.g. 0 or 1). The general form of the model is:
$P(Y_i = 1| X_i) = F(X_i \beta)$
where the dependent variable *Y* takes the value of either 0 or 1. The question hinges on the value of the parameter *P*, the probability that *Y* equals one. *X* is the set of explanatory variables and *F* (.) is a nonlinear function of the conditional mean. *F* (.) is the cumulative density function (CDF) of the logistic distribution of the logit model.

Discriminant analysis and logit model are particularly suited for this study, since this study attempts to predict the direction in the movement of a stock index. The direction of movement is binary in nature (up or down). Leung *et al.* (2000) tested various classification models, which predict direction of three globally traded broad market indices —S&P 500 for the US, FTSE 100 for the UK, and Nikkei 225 for Japan, based on probability, include linear discriminant analysis, logit, and neural network. Neural network performed better than the discriminant and logit model for US and UK market but in case of Japanese market discriminant analysis outperformed the other models. Huang *et al.* (2005) investigated the predictability of financial movement direction with SVM by forecasting the weekly movement direction of NIKKEI 225 index. To evaluate the forecasting ability of SVM, they compared its performance with those of Linear Discriminant Analysis, and Elman Backpropagation Neural Networks. The experiment results show that SVM outperforms the other classification methods. Yoon *et al.* (1993) used neural network for predicting the stock price performance. They compared the results of the neural network by benchmarking the results against the multiple discriminant analysis (MDA). The MDA achieved an accuracy of 65 %, while the neural network achieved an accuracy of 77%. Results confirm the predictability and superiority of neural networks. In most of the studies pertaining to bankruptcy prediction, corporate distress examination, bond rating, etc discriminant analysis and logit models are used as a benchmark (Adam and Lin, 2001). So it is reasonable to use these models for predicting the direction of S&P CNX NIFTY Index direction movement.

Neural Network Methodology

A neural network is a collection of interconnected simple processing elements. The most popular and successful one is the feedforward multilayer network or the backpropagation neural network (BPN). A BPN is typically composed of several layers of nodes. The first or the lowest layer is an input layer where external information is received. The last or the highest layer is an output layer where the problem solution is obtained. The input layer and output layer are separated by one or more intermediate layers called the hidden layers. The units in the network are connected in a feedforward manner, from the input layer to the output layer. Every connection in a neural network has a weight attached to it.

For the purpose of application of Artificial Neural Network (ANN), backpropagation algorithm is very popular in literature. In backpropagation algorithm input variables are passed forward to the hidden layer from the input layer and multiplied by their respective weights to compute a weighted sum of total input value to a neuron in the hidden layer. The weighted sum is modified by a transfer function and then sent as input to neurons in the next layer (hidden or output). The transfer function can take different forms like Sigmoid, Tanhyperbolic, Linear, Gaussian etc. They stand for the signals thus generated from earlier layers to later layers and the signal finally reaches the output layer. The output layer neuron re-calculates the weighted sum and applies the transfer function to produce the output value of the signal received by it. Finally, an error signal is backpropagated to the hidden layer in a sequence opposite to that of the input variable. The error signal is computed as the difference between the output value of the neural network and the actual output value (also called the target value of the neural network) The weights that connect two layers are adjusted proportionally according to the contribution of each neuron to the forecast error. This is done so as to minimize the mean squared error (MSE). This training process continues until an acceptable MSE target that is specified based on requirement is achieved. One typical method for training a network is to first separate the data series into two disjoint sets: the training set and the test set. The network is trained

(e.g., with backpropagation) directly on the training set (i.e. arrive at set of weights between two neurons). This trained network is used to forecast and its ability to forecast is measured on the test set.

Model Formulation

This study employs a three-layer backpropagated neural network to forecast NIFTY Index direction. The input variables are the twelve technical indicators, and are fed to the neural network model to forecast the next period direction in this model. For example, the inputs to a 12-*x*-1 neural network are NX_{12}, NX_{11}, NX_{10}, …..and NX_{1} and while the output of the neural network is the next day' s direction of NIFTY Index i.e. a binary value (1 or 0), where NX_{i} stands for the different technical indicators used in this study. The architecture of the neural network is denoted by X-Y-Z. The X-Y-Z stands for a neural network with X neurons in input layer, Y neurons in hidden layer, and Z neurons in output layer. Only one output node is deployed in the output layer since one-step-ahead forecast is made in this study. The number of hidden nodes is not specified *a priori*. This will be selected through experiment. This study uses tansigmoid function for the nodes in the input layer for backpropagated neural network, while tansigmoid function and pure linear function are used at hidden layers and output layers.

The number of input nodes is probably the most critical decision variable for a time series-forecasting problem since it contains important information about the data. In this study, the number of input nodes corresponds to the twelve technical indicators used to discover the underlying pattern in a time series and to make forecasts for future values. Currently, there is no theory suggesting the appropriate number of input nodes. But ideally it would be better to have a small number of essential nodes, since this can unveil the unique features embedded in the data. Too few or too many input nodes can affect either the learning or prediction capability of the network. This study selects 12 technical indicators to make up the initial attributes, as determined by prior research, Kim and Han (2000) and Kim (2003).

The number of hidden nodes plays a very important role too. These hidden neurons enable the network to detect the feature, to capture the pattern in the data, and to perform complicated nonlinear mapping between input and output variables. Hornik *et al.* (1989) in their theoretical work found that single hidden layer is sufficient for the network to approximate any complex non-linear function with any desired accuracy. Most authors use only one hidden layer for forecasting purposes. This study employs three-layer BPN to forecast the daily direction of NIFTY Index. This study varies the number of nodes in the hidden layer for training. In this study, 2, 4, 6, 8, 10, and 12 hidden nodes have been experimented because the neural network does not have a general rule for determining the optimal number of hidden nodes.

This study uses backpropagation algorithm to train the BPN. Backpropagation is the most widely used algorithm for supervised learning with neural networks. The study uses MATLAB 6.5 to build and train neural network. The MATLAB program works with default parameter values of weight, assigned by the MATLAB.

Random Forest

In random forest as proposed by Breiman (2001), a random vector θ_k is generated, independent of the past random vectors θ_1, ... , θ_{k-1} but with the same distribution; and a tree is grown using the training set and θ_k , resulting in a classifier $h(\mathbf{x}, \theta_k)$ where $\mathbf{x}$ is an input vector. In random selection θ consists of a number of independent random integers between 1 and K. The nature and dimensionality of θ depends on its use in tree construction. After a large number of trees are generated, they vote for the most popular class. This procedure is called random forests.

A random forest is a classifier consisting of a collection of tree structured classifiers $\{h(\mathbf{x}, \theta_k), k=1, ...\}$ where the $\{\theta_k\}$ are independent identically distributed random vectors and each tree casts a unit vote for the most popular class at input $\mathbf{x}$.

Given an ensemble of classifiers $h_1(\mathbf{x}), h_2(\mathbf{x}), ... , h_k(\mathbf{x})$, and with the training set drawn at random from the distribution of the random vector Y,$\mathbf{X}$, define the margin function as

$Mg(\mathbf{X}, Y) = av_k I(h_k(\mathbf{X}) = Y) \quad \max_{j \neq y} av_k I(h_k(\mathbf{X}) = j)$

where $I(\bullet)$ is the indicator function. The margin measures the extent to which the average number of votes at **X**, Y for the right class exceeds the average vote for any other class. The larger the margin, the more confidence in the classification. The generalization error is given by

$PE^* = P_{XY}(mg(\mathbf{X}, Y) < 0)$

where the subscripts **X**, *Y* indicate that the probability is over the **X**, *Y* space.

In random forests, $h_k(\mathbf{X}) = h(\mathbf{X}, \theta_k)$. For a large number of trees, it follows from the Strong Law of Large Numbers and the tree structure that: As the number of trees increases, for almost surely all sequences $\theta_1, \ldots$ PE^* converges to

$P_{XY}(P\theta(h(\mathbf{X}, \theta) = Y) \quad \max_{j \neq y} P\theta(h(\mathbf{X}, \theta) = j) < 0)$

The convergence of the above equation explains why random forests do not overfit as more trees are added, but produce a limiting value of the generalization error.
The strategy employed to achieve these ends is as follows:
1. To keep individual error low, grow trees to maximum depth.
2. To keep residual correlation low randomize via
a) Grow each tree on a bootstrap sample from the training data.
b) Specify $m << p$ (the number of covariates). At each node of every tree select *m* covariates and pick the best split of that node based on these covariates.

As operationalized by the random forest software, available from http://www.stat.Berkeley.EDU/users/breiman/rf.html, the size of the individual trees constituting the forest is controlled by a tuning parameter, nthsize. This specifies the number of cases in a node below which the tree will not split, and so determines maximal tree size. The user manual asserts that, in large datasets, larger values can be employed for memory and speed considerations with little loss of accuracy. In this study, an experiment is done to find out the optimal nthsize as well as the primary tuning parameter, *m*.

In summary, for binary outcomes, random forests construct an ensemble of classification trees. Each tree is built from a bootstrap sample of the data and at each split; a random sample of predictors is examined. In the end, classification is determined by a majority vote for each case over the ensemble of classification trees.

Support Vector Machines

SVM uses linear model to implement nonlinear class boundaries through some nonlinear mapping the input vectors x into the high-dimensional feature space. A linear model constructed in the new space can represent a nonlinear decision boundary in the original space. In the new space, an optimal separating hyperplane is constructed. The points on either side of the separating hyperplane have distances to the hyperplane. The smallest distance is called the margin of separation. Let q be the margin of the optimal hyperplane. The points that are distance q away from the OSH are called the support vectors. All other training examples are irrelevant for defining the binary class boundaries.

Consider the problem of separating the set of training vector belonging to two separate classes, Given a set of data points $G = \{(x_i, d_i)\}_i^n$ (x_i is the input vector, d_i is the desired value ($d_i \in \{0, 1\}$ is known as binary target) and n is the total number of data patterns), SVMs approximate the function using the following:

$y = f(x) = w \Phi(x) + b;$

where $\Phi(x)$ is the high dimensional feature space which is non-linearly mapped from the input space x. The coefficients w and b are estimated by minimizing

$$R_{SVM_S} = C\frac{1}{n}\sum_{i=1}^{n} L_\varepsilon(d_i y_i) + \frac{1}{2}\| w \|^2$$

$$L_\varepsilon(d, y) = \begin{cases} |d-y| - \varepsilon, & |d-y| \quad \varepsilon \\ 0 & \text{otherwise} \end{cases}$$

The first term $C\frac{1}{n}\sum_{i=1}^{n} L_\varepsilon(d_i y_i)$ is the so-called empirical error (risk), which is measured by the ε-insensitive loss function. The second term $\frac{1}{2}\| w \|^2$ on the other hand, is called the regularized term. ε is called the tube size of SVMs, and C is the regularization constant determining the trade-off between the empirical error and the regularized term. They are both user-prescribed parameters and are selected empirically.

As mentioned above, SVM constructs linear model to implement nonlinear class boundaries through the transforming the inputs into the high-dimensional feature space. For the nonlinearly separable case, a high-dimensional version of is simply represented as follows:

$$Y = b + \sum \alpha_i y_i K(x_i, x_j)$$

The function $K(x_i, x_j)$ is defined as the kernel function. The value is equal to the inner product of two vectors x_i and x_j in the feature space $\Phi(x_i)$ and $\Phi(x_j)$. That is, $K(x_i; x_j) = \Phi(x_i).\Phi(x_j)$. There are some different kernels for generating the inner products to construct machines with different types of nonlinear decision surfaces in the input space. Choosing among different kernels the model that minimizes the estimate, one chooses the best model. Common examples of the kernel function are the polynomial kernel $K(x, y) = (xy+1)^d$ and the Gaussian radial basis function $K(x, y) = \exp(-1/\delta^2 (x-y)^2)$, where d is the degree of polynomial kernel and δ^2 is the bandwidth of the Gaussian radial basis function kernel.

From the implementation point of view, training SVMs is equivalent to solving a linearly constrained quadratic programming (QP) problem with the number of variables equal to the number of training data points.

In this study, the Gaussian radial basis function is used as the kernel function of SVM. Tay and Cao (2001) showed that the upper bound C (regularization constant) and the kernel parameter δ^2 play an important role in the performance of SVMs. Improper selection of these two parameters can cause the over fitting or the under fitting problems. Since there is few general guidance to determine the parameters of SVM, this study varies the parameters to select optimal values for the best prediction performance. This study uses SVMtorch software system to perform experiments.

4. RESULTS

Each of the classification models described in the last section is estimated by in-sample data i.e. (1st January, 2000 to 23rd February). The model estimation selection process is then followed by an empirical evaluation which is based on the out-sample data i.e. (24th February to 31st May, 2005). At this stage, the relative performance of the models is measured by hit ratio.

One of the advantages of linear SVM is that there is no parameter to tune except the constant C. For the nonlinear SVM, there is an additional parameter, the kernel parameter, to tune. Kim (2003) found that the polynomial function kernel function takes a longer time in the training of SVM and provides worse results than the Gaussian radial basis function in preliminary test. Thus, this study uses the Gaussian radial basis function as the kernel function of SVMs.

This study used different set of experiments to find out the best performance of SVM with respect to various kernel parameters and constants. The value of δ^2 experimented within a range of 1 and 140. In

addition, the parameter C was experimented between 1 and 15. Table 1 presents the some of the best results of prediction performance of SVMs with various parameters, where C varies from 1 to 5 and δ^2 varies from 40 to 140.

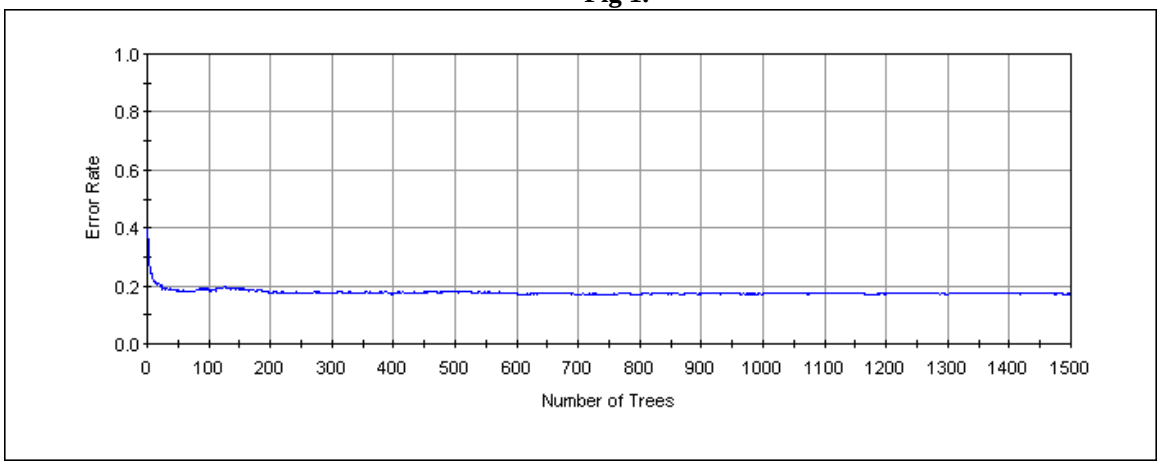
Table 1
The prediction performance of various parameters in SVMs

Hit Ratio											
C/δ^2	**40**	**50**	**60**	**70**	**80**	**90**	**100**	**110**	**120**	**130**	**140**
1	62.46	63.79	65.78	**68.44**	66.78	64.78	63.79	63.12	63.12	63.12	63.46
2	61.79	63.46	64.12	64.78	64.78	**67.11**	64.78	65.78	63.79	62.46	61.79
3	59.14	61.79	60.8	61.79	64.12	65.78	**65.78**	63.79	62.46	61.46	62.13
4	58.47	61.13	60.47	60.13	61.79	63.46	62.13	**64.45**	63.79	62.13	61.13
5	59.14	60.8	60.47	60.13	61.46	62.46	61.79	62.13	**63.14**	62.13	62.13

The best prediction performance of the out-of-sample data is recorded when δ^2 is 70 and C is 1. The prediction performance on the holdout data decreases when C increases from 1 to 15. It can be observed that predictive performance of SVM increases with the increase in δ^2 upto a certain value for a given value of C and then start decreasing. The predictive performance of SVM is sensitive for the given range of δ^2. The prediction performance for the out-of-sample data for SVM is 68.44% when C is 1 and δ^2 is 70.

In addition, this study compares the best SVM model with random forest, neural network, discriminant analysis and logit model. The S&P CNX NIFTY index direction has been forecasted using random forest method. Fig 1. Shows the graph of the error curve of random forest model as the number of tree grow. This graph displays the running relative rate of error for the random forest model. It begins with the first tree having an average error rate of about .4 as trees grow from 0 to 100 the error rate decreases exponentially. When the number of trees is 100 the average error rate is almost .2 after that it started decreasing and ends with an average error rate of .18.

Fig 1.



The number of predictors considered for each node is 3 and minimum cases of parent node are selected 2. These parameters are by default set in the software. This experimented with different number of trees which varies from 200 to 1500, but there is no significant effect on the reduction of error rate for the out-of-sample data. The prediction performance for the out-of-sample data in case of random forest 67.40%.

In addition to the SVM and random forest, backpropagation neural network has been used in this to predict the direction of stock index. This study has used twelve technical indicators as input to the neural network model and the number of nodes in the hidden layer varies from 2 to 12 in a step of 2 because the neural network does not have a general rule for determining the optimal number of hidden nodes. The combination of twelve input nodes and six hidden nodes (2, 4, 6, 8, 10, 12) yields a total of 6 different neural network architecture which are being considered for each in-sample training set for forecasting the

direction of S&P CNX NIFTY Index. Table 2 gives the prediction performance of various neural network models.

Table 2
Results of various Neural Network Model

	No of Hidden Nodes					
Hit Ratio	2	4	6	8	10	12
	59.13	62.79	**62.93**	62.45	61.12	60.48

This study experimented with different levels of hidden nodes, which varies from 2 to 12 in a step of 2. The best prediction performance for the out-of-sample data is produced when the numbers of hidden processing elements are 6. The prediction performance of the out-of-sample data is 62.93%. The Neural network model 12 – 6 – 1 provides better fit for forecasting the direction of NIFTY Index

For the NIFTY Index series one-period-ahead forecast of the direction were produced by the five classification models namely logit, discriminant, neural network, random forest and SVM. The predictive performance of the five models measured in terms of hit ratio is summarized in Table 3. Table 3 compares the best prediction performances of SVM, random forest, neural network, discriminant analysis and logit model

Table 3
Forecasting performance of different classi1cation methods

Classification Method	Hit ratio
Discriminant Analysis	56.34%
Logit Model	59.60%
Neural Network	62.93%
Random Forest	67.40%
Support Vector Machines	**68.44%**

Discriminant analysis performs worst, with a hit ratio of 56.34%. Logit model outperforms discriminant analysis in term of hit ratio, because discriminant analysis assumes that all the classes have equal covariance matrices, which is not consistent with the properties of input variable belonging to different classes. In fact, the two classes have different covariance matrices.

SVM outperforms random forest and neural network by 1.04% and 5.51 % respectively. Although SVM is only marginally better than the random forest i.e. 1.04 %, SVM has the highest forecasting accuracy among the individual forecasting methods. While comparing random forest method with neural network, logit and discriminant analysis, one can note that random forest perform better than the other three models in terms of hit ratio. While comparing neural network with logit and discriminant analysis, neural network model outperforms the other two with a hit ratio of 62.93%.

The results indicate the feasibility of SVM in forecasting the direction of financial time series and are compatible with the findings of Kim (2003) and Tay and Cao (2001).

5. CONCLUSION

This study used SVM and random forest to predict the daily movement of direction of S&P CNX NIFTY Index and compared the results with that of traditional discriminant and logit model and artificial techniques like neural network. The experimental results showed that SVM outperformed random forest, neural network and other traditional models used in this study. The superior performance of SVMs over the other models is due to the reason that SVMs implement the structural risk minimization principle which minimizes an upper bound of the generalization error rather than minimizes the training error. This eventually leads to better generalization than the neural network and random forest which implements the empirical risk minimization principle.

Although SVM was slightly better than the random forest, the two models can further be evaluated for different financial time series such as exchange rates, and different stock market index, to find out a conclusive result. The machine learning methods like support vector machines and random forest will help traders, borrowers, FII etc, to make better investment decision. Financial forecaster, dealers, and traders, can use different trading approaches based on the machine learning techniques, which can lead to financial gain.

However, each method has its own strengths and weaknesses. Future research can also be done by combining models by integrating SVM with other classification models. The weakness of one method can be balanced by the strengths of another by achieving a systematic effect. There is also scope to assess the direction of stock market index, taking into account the set of potential macroeconomic input variables such as interest rates, consumer price index, industrial production etc. Thus to conclude, SVM is a useful tool for economists and practitioners dealing with the forecasting stock market exchange rate vis-à-vis neural network and other linear models.

REFERENCES

Adam, F., Lin, L. H., "An Analysis of the Applications of Neural Networks in Finance", Interfaces, 31 (4), 2001, 112–122.

Bansal, R.., Viswanathan, S., " No Arbitrage and Arbitrage Pricing: A new Approach", *Journal of Finance*. 48(4), 1993, 1231-1262.

Banz R, Breen W., "Sample-dependent results using accounting and market data: some evidence", *Journal of Finance*, 41, 1986, 779–93.

Barr, D., Mani, G., "Using Neural Nets to Manage Investments", *AI Expert*, February, 1994, 16 –21.

Breiman, L., "Random Forests", *Machine Learning*, 45, 2001, 5-32.

Campbell J., "Stock returns and the term structure", *Journal of Financial Economics*, 18, 1987, 373–99.

Chakraborty, K., Mehrotra, K., Mohan, C. K., Ranka, S., "Forecasting the Behaviour of Multivariate Time Series Using Neural Networks", *Neural Networks*, 5(2), 1992, 961 – 970

Cheng W, Wanger L, Lin CH., "Forecasting the 30-year US treasury bond with a system of neural networks", *Journal of Computational Intelligence in Finance*, 4, 1996,10–6.

Donaldons, R. G., Kamstra, M., "A New Dividend Forecasting Procedure Rejects Bubbles in Asset Prices: The Case of 1929 Stock Crash", *Review of Financial Studies*, 9(2), 1996, 333-383.

Fama E, Bliss R., "The information in long-maturity forward rates", *American Economic Review*, 77, 1987, 680–92.

Fama E, French K., "Permanent and temporary components of stock prices", *Journal of Political Economy*, 96, 1988, 246–73.

Fama E, French K., "Dividend yields and expected stock returns", Journal of Financial Economics, 22, 1988, 3–25.

Fama E, French K., "Business conditions and expected returns on stocks and bonds", Journal of Financial Economics, 25, 1990, 23–49.

Fama E, French K., "The cross-section of expected stock returns", *Journal of Finance*, 47, 1992, 427–65.

Fama E, Schwert W., "Asset returns and inflation", *Journal of Financial Economics*, 5, 1977,115–46.

Ferson W. E., Harvey C. R.., "The risk and predictability of international equity returns", *Review of Financial Studies*, 6, 1993, 527–66.

Grudnitski, G., Osborn, L., "Forecasting S & P and Gold Futures Prices: An Application of Neural Networks", *The Journal of Futures Markets*, 13 (6), 1993,631 – 643, 1993.

Harvey, C. R., "Predictable risk and returns in emerging markets", *The Review of Financial Studies*, 8, 1995, 773–816.

Hodgson, A., Nicholls, D., "The impact of index futures markets on Australian share market volatility", *Journal of Business Finance and Accounting*, 8, 1991, 267–80.

Hornik K., Stinchcombe M., White H., "Multilayer feedforward networks are universal approximators", *Neural Networks*, 2, 1989, 359–366.

Jaffe, J, Keim, D., Westerfield, R., "Earnings yields, market values and stock returns", *Journal of Finance*, 44, 1989, 135–48.

Jaffe, J., Westerfield, R., "Patterns in Japanese common stock returns: day of the week and turn of the year effects", *Journal of Financial and Quantitative Analysis*, 20, 1985, 261–72.

Jung, C., Boyd, R., "Forecasting UK stock prices". *Applied Financial Economics*, 6, 1996, 279–86.
Kamruzzaman, J., Sarker, R. A., "ANN-Based Forecasting of Foreign Currency Exchange Rates", *Neural Information Processing - Letters and Reviews*, 3 (2), 2004.
Kato, K., Ziemba, W., Schwartz, S., Day of the week e4ects in Japanese stocks. In: Elton E, Grubber M, editors. Japanese capital markets. New York: Harper & Row, 1990.
Keim, D., Stambaugh, R., "Predicting returns in the stock and bond markets", *Journal of Financial Economics*, 17, 1986, 357–90.
Kim, K. J., "Financial time series forecasting using support vector machines", *Neurocomputing*, 55, 2003, 307 – 319
Kim, K., Han, I., "Genetic algorithms approach to feature discretization in artificial neural networks for the prediction of stock price index", Expert System Application, 19 (2), 2000, 125–132.
Kimoto, T., Asakawa, K., Takeoka, M., "Stock Market prediction system with modular neural networks", In: proceedings of the IEEE International Joint Conference on Neural Networks. San Diego, California, 2, 1990, 11-16.
Kryzanowski, L., Galler, M., Wright, D.W., "Using Artificial Networks to pick stocks", *Financial Analyst Journal*, August, 1993, 21-27.
Larivière, B., Poel, D. V. D., "Predicting Customer Retention and Profitability by Using Random Forests and Regression Forests Techniques", Working Paper, Department of Marketing, Hoveniersberg 24, 9000 Gent, Belgium, 2004.
Leung, M. T., Daouk, H., Chen, A. S., "Forecasting stock indices: a comparison of classification and level estimation models", *International Journal of Forecasting*, 16, 2000, 173–190.
Maberly, E. D., "The informational content of the interday price change with respect to stock index futures", *Journal of Futures Markets*, 6, 1986, 385–395.
McCluskey, P. C., "Feedforward and Recurrent Neural Networks and Genetic Programming for Stock Market and Time Series Forecasting", Department of Computer Sciences, Brown University, 1993.
Mukherjee, S., Osuna, E., Girosi, F., "Nonlinear prediction of chaotic time series using support vector machines", in: Proceedings of the IEEE Workshop on Neural Networks for Signal Processing, Amelia Island, FL, 1997,511–520.
O' Connor, M., Remus, W., & Griggs, K., Going up-going down: "How good are people at forecasting trends and changes in trends?", *Journal of* Forecasting, 16, 1997, 165–176.
Refenes, A. N., Bentz, Y., Bunn, D. E., Burgess, A. N., Zapranis, A. D., "Financial Time Series Modeling with Discounted Least Square Backpropagation", *Neuro Computing*, 14, 1987, 123-138.
Refenes, N., Azema-Barac, M., Chen, L., and Karoussos, S. A., "Currency Exchange Rate Prediction and Neural Network Design Strategies" *Neural Computing and Applications*, 1, 1993, 46 – 58.
Refenes, A. N., Zapranis, A. S., Francis, G., "Stock Performance Modeling Using Neural Networks: Comparative Study With Regressive Models", *Neural Networks*, 7(2), 1994, 375-388.
Rozeff M., "Dividend yields are equity risk premiums", *Journal of Portfolio Management*, 11, 1984, 68–75.
Sharda R, Patil RB. A connectionist approach to time series prediction: an empirical test. In: Trippi, RR, Turban, E, (Eds.), Neural Networks in Finance and Investing, Chicago: Probus Publishing Co., 1994, pp. 451–64.
Takashi, K., Kazuo, A., "Stock Market Prediction System with Modular Neural Network", *International Joint Conference on Neural Networks*, 1, 1990, 1-6.
Tay, F. E. H., Cao, L., "Application of support vector machines in financial time series forecasting", Omega, 29, 2001, 309–317.
Van E, Robert J. The application of neural networks in the forecasting of share prices. Haymarket, VA, USA: Finance & Technology Publishing, 1997.
V.N. Vapnik, Statistical Learning Theory, Wiley, New York, 1998.
Wu, Y., & Zhang, H., "Forward premiums as unbiased predictors of future currency depreciation: A non-parametric analysis", *Journal of International Money and Finance* 16, 1997, 609–623.
Yao, J., Chew, L. T., and Poh, H. L., "Neural networks for technical analysis: a study on KLCI", *International Journal of Theoretical and Applied Finance*, 2(2), 1999, 221-241
Yoon, Y., Swales, G., Margavio, T. M., 1993, "A comparison of discriminant-analysis versus artificial neural networks," *Journal of the Operational Research Society*, 44 (1), 1993, 51–60.
Zirilli, Josephs., Financial Prediction Using Neural Network, London, International Thompson Computer Press, 1997.

APPENDIX 1

Attribute Name	Description	Formula
% K	Stochastic %K. It compares here a security's price closed elative to its price range over a given time period.	$\frac{C_t - LL_{t-n}}{HH_{t-n} - LL_{t-n}} * 100$, where LL_t and HH_t are mean lowest low and highest high in the last t days, respectively.
% D	Stochastic %D. Moving average of %K.	$\frac{\sum_{i=0}^{n-1} \%K_{t-i}}{n}$
Slow % D	Stochastic slow %D. Moving average of %D.	$\frac{\sum_{i=0}^{n-1} \%D_{t-i}}{n}$
Momentum	It measures the amount that a security's price has changed over a given time span.	$C_t - C_{t-4}$
ROC	Price rate-of-change. It displays the difference between the current price and the price n days ago.	$\frac{C_t}{C_{t-n}} * 100$
William's % R	Larry William's %R. It is a momentum indicator that measures overbought/oversold levels.	$\frac{H_n - C_t}{H_n - L_n} * 100$
A/D Oscillator	Accumulation/distribution oscillator. It is a momentum indicator that associates changes in price.	$\frac{H_t - C_{t-1}}{H_t - L_t}$
Disparity 5	5-day disparity. It means the distance of current price and the moving average of 5 days.	$\frac{C_t}{MA_5} * 100$
Disparity 10	10-day disparity.	$\frac{C_t}{MA_{10}} * 100$
OSCP	Price oscillator. It displays the difference between two moving averages of a security's price.	$\frac{MA_5 - MA_{10}}{MA_5}$
CCI	Commodity channel index. It measures the variation of a security's price from its statistical mean.	$\frac{M_t - SM_t}{0.015 D_t}$ where $M_t = (H_t + L_t + C_t)/3$ $SM_t = \frac{\sum_{i=1}^{n} M_{t-i+1}}{n}$, and $D_t = \frac{\sum_{i=1}^{n} \lvert M_{t-i+1} - SM_t \rvert}{n}$
RSI	Relative strength index. It is a price following an oscillator that ranges from 0 to 100.	$100 - \frac{100}{1 + (\sum_{i=1}^{0} Up_{t-i}/n) / (\sum_{i=0}^{n-1} Dw_{t-i}/n)}$ where Upt means upward-price-change and Dwt

		means downward-price-change at time t.

C_t is the closing price at time t, L_t the low price at time t, H_t the high price at time t and, MA_t the moving average of t days.

Author' s Profile:

Manish Kumar received his B. E degree in Mechanical Engineering from Pt. Ravi Shankar Shukla University Raipur, and M. S by Research in Finance from Indian Institute of Technology Madras, India. He is currently doing Ph. D. at Department of Management Studies, IIT Madras, India. His research interest includes Forecasting, Artificial Intelligence, Meta heuristics and Time Series Analysis.

M. Thenmozhi received her M. Com, M. Phil and Ph. D in Finance from Madras University. She is currently Associate Professor of Department of Management Studies, IIT Madras, India. Her research area includes, Financial Management, Strategic Management, Econometrics, Financial Time Series Forecasting.

Address:

Manish Kumar *
Ph. D. Research Scholar
Department of Management Studies,
Indian Institute of Technology, Madras
Chennai – 600 036, India
ms04d001@iitm.ac.in

Dr. M. Thnmozhi
Associate Professor
Department of Management Studies,
Indian Institute of Technology, Madras
Chennai – 600 036, India
mtm_iitm@yahoo.com

* Corresponding Author