
Forecasting stock index returns using ARIMA-SVM, ARIMA-ANN, and ARIMA-random forest hybrid models

Manish Kumar*

CRISIL Global Research and Analytics,
Chennai: 600036, India
Fax: 91-44-2257-4552
E-mail: manishkumar_iitm@yahoo.co.in
*Corresponding author

M. Thenmozhi

Department of Management Studies,
Indian Institute of Technology Madras,
Chennai: 600036, India
Fax: 91-44-2257-4552
E-mail: mtm_iitm@yahoo.com

Abstract: The purpose of this paper is to develop and identify the best hybrid model to predict stock index returns. We develop three different hybrid models combining linear ARIMA and non-linear models such as support vector machines (SVM), artificial neural network (ANN) and random forest (RF) models to predict the stock index returns. The performance of ARIMA-SVM, ARIMA-ANN and ARIMA-RF are compared with performance of ARIMA, SVM, ANN and RF models. The various competing models are evaluated in terms of statistical metrics and trading performance criteria via a trading strategy. The analysis shows that the hybrid ARIMA-SVM model is the best forecasting model to achieve high forecast accuracy and better returns.

Keywords: hybrid models; ARIMA; artificial neural network; ANN; support vector machines; SVM; random forest; forecasting; stock market trading.

Reference to this paper should be made as follows: Kumar, M. and Thenmozhi, M. (2014) 'Forecasting stock index returns using ARIMA-SVM, ARIMA-ANN, and ARIMA-random forest hybrid models', *Int. J. Banking, Accounting and Finance*, Vol. 5, No. 3, pp.284–308.

Biographical notes: Manish Kumar received his BE in Mechanical Engineering from the Pandit Ravi Shankar Shukla University Raipur, India and MS by research and PhD in Finance from the Indian Institute of Technology Madras, India. Currently, he is working at CRISIL Global Research and Analytics, India. His area of interest is market microstructure, forecasting, financial time series analysis and artificial intelligence.

M. Thenmozhi is currently a Professor in Finance in the Department of Management Studies of Indian Institute of Technology Madras, India. She holds a doctorate in Commerce from the University of Madras and has been a teacher and researcher in finance for more than 18 years. She is a recipient of Fulbright-Nehru Fellowship, European Union Erasmus Mundus Scholarship, an

Executive Endeavour Award Fellow of the Australian Government and a National Award Winner for Developing Best Manufacturing Policy Draft for India from FICCI and WTI, India. Her research interest includes corporate finance, financial markets, computational finance, and financial time series forecasting.

1 Introduction

Financial time series such as stock price and exchange rate are dynamic, non-linear, complicated, and chaotic in nature. They are also affected by various factors such as political events, firm's policies, general economic conditions, investor's expectations, institutional investor's choices, movement of other financial market, and behavioural aspects of investors, etc. Hence, forecasting the movement of financial time series is considered to be a very challenging task for both academicians and market participants (Kumar and Thenmozhi, 2009).

Numerous studies have examined the predictability of financial time series using a wide range of linear and non-linear models. The use of traditional time series models such as ARIMA, logit/probit models, discriminant analysis, multiple regressions, vector autoregression and vector error correction model among others have been obvious choices in earlier studies to forecast financial time series such as stock price and exchange rates. The approximation of traditional time series methods to model financial time series is not always satisfactory, since, financial time series exhibit high non-linearity. With computational ability, the focus has shifted to the use of artificial intelligence and machine learning techniques such as artificial neural networks (ANN), support vector machines (SVM) and random forest (RF). Machine learning methods are aimed at identifying the elusive and ever-changing patterns in data and are concerned with the design and development of algorithms and techniques that allow computers to 'learn' (Kumar and Thenmozhi, 2009).

ANN, SVM and RF have been successfully employed to forecast a wide range of financial time series. While ANN is a very successful tool for time series modelling and forecasting, it has certain limitations in learning the pattern if the data is noisy and has complex dimensions (Zhang et al., 1998). ANN may not converge to global solutions and has problems of overfitting and thereby loose generalisability. Moreover, proper selection of parameters such as input variables, number of epochs and learning rate, is difficult in ANN (Kim, 2003). Researchers have attempted to evolve a simpler and more accurate model and SVM and RF are recently being used in various financial applications. SVM is widely used for pattern recognition and results in better generalisation than conventional techniques and ANN. SVM and RF have been used to solve estimation problems and they exhibit excellent performance, because, they avoid the question of modelling the underlying distribution and focus on making accurate prediction of certain variables, given other variables (Kim, 2003; Kumar and Thenmozhi, 2009). In other words, these techniques provide an analytical alternative to conventional techniques which are often limited by strict assumptions of normality, linearity, variable independence, etc. In addition, SVM and RF provide solutions that may be the global optimum while ANN models may tend

to fall into a local optimal solution. Thus, overfitting is unlikely to occur with SVM and RFM.

The use of non-linear techniques is widely accepted to examine stock market behaviour (Atsalakis and Valavanis, 2009) and in the recent years, the focus is on the use of a combination of linear and non-linear techniques to forecast. In addition, the combined model is more robust and flexible with regard to the possible structural change in the data. There have been studies which combine fuzzy logic with ANN (Yu and Huarng, 2008), ARIMA and SVM (Pai and Lin, 2005), ARIMA and ANN (Zhang, 2003), Kohonen self-organising map with ARIMA (Voort et al., 1996), ARIMA and ANN (Su et al., 1997), radial basis function and ARIMA (Wedding and Cios, 1996), regression and ANN (Luxhoj et al., 1996), and combining several feed forward neural network (Pelikan et al., 1992; Ginzburg and Horn, 1994) to improve the time series forecasting accuracy.

Existing literature has ample empirical evidence of ANN outperforming traditional models like multiple regression, discriminant analysis, and logit model (see Kara et al., 2011; for excellent review). There is also evidence of SVM outperforming ANN for stock market return series (see Kara et al., 2011; for review) and RFM outperforming logistic model (Kumar and Thenmozhi, 2009). In addition, it is also known that hybrid ARIMA-ANN model outperforms the ANN model and that hybrid ARIMA-SVM model outperforms SVM. However, there is no evidence on whether ARIMA-ANN is better than ARIMA-SVM or not. Moreover, to the best of our knowledge, there has been no attempt to develop ARIMA-RF hybrid model and to examine its applicability in financial time series forecasting. Hence, the performance of hybrid models viz., ARIMA-ANN, ARIMA-SVM and ARIMA-RF is yet to be compared.

An emerging economy like India could be a good testing ground for the viability of hybrid linear and non-linear forecasting models like ARIMA-ANN, ARIMA-SVM and ARIMA-RF, as testified by the study conducted by Harvey (1995). Since he documented a higher level of predictability in emerging market returns than in developed market returns. Hence, we develop three hybrid models namely ARIMA-ANN, ARIMA-SVM and ARIMA-RF to forecast the daily returns of S&P CNX Nifty. We also examine whether these hybrid models have the ability to outperform independent models such as ANN, SVM, RF and ARIMA.

The empirical results clearly suggest that the ARIMA-SVM hybrid model is able to outperform the independent models and the other two hybrid model in terms of MAE, RMSE and sign and directional accuracy (DA) test. Overall, we confirm that hybrid models outperform independent models. The finding is similar to results of Wedding and Cios (1996), Zhang (2003) and Pai and Lin (2005), but contradicts the findings of Tugba and Casey (2005). The result also shows that the ARIMA-SVM model outperforms ARIMA-ANN and ARIMA-RF hybrid models and other independent models in terms of various trading measures like annualised returns, Sharpe ratio, etc.

The remaining section of the paper is organised as follows. The literature review is discussed in Section 2. The data and methodology used to develop different hybrid models and the benchmark models are explained in Section 3. The empirical results from the real data sets are discussed in Section 4. Finally, Section 5 summarises the findings and brings out the implications of the study.

2 Literature review

In the last two decades, several studies have used a wide range of linear and non-linear models for forecasting financial time series. The most popular and traditional time series model is the linear Box-Jenkins or ARIMA model. The ARIMA approach is both simple and yields accurate results and is widely used. Many authors (Virtanen and Paavo, 1987; Pagan and Schwert, 1990; Lesseps and Morrel, 1997; Crawford and Fratanoni, 2003) have used the ARIMA model to forecast different time series such as stock index returns and exchange rates and have compared it with different models like Markov switching, regime switching, GARCH, etc. The results show that ARIMA models performed better than other models. Because of its popularity, the ARIMA model has been used as a benchmark to evaluate many new modelling approaches (Hwang and Ang, 2002).

The major limitation of the ARIMA model is the pre-assumed linear form of the model. Financial time series are considered as highly non-linear where the mean and variance of the series changes over time. Grudnitski and Osborn (1993) stated that there is noisy non-linear process present in the prices. Refenes et al. (1994) indicated that traditional statistical techniques for forecasting have serious limitations with respect to applications with non-linearity in the data set such as stock indices. Hence, detecting hidden non-linear relationship and the application of non-linear models may help in improving forecasting accuracy. One of the popular and powerful tools in this area is ANN. The major advantage of ANN is their flexible non-linear modelling capability (Donaldson and Kamstra, 1996).

Several authors have found that non-linear models like ANN outperform the traditional models like ARIMA and generate better prediction of stock prices. They also perform better than regression (Refenes, 1993; Hill et al., 1996; Desai and Bharati, 1998; Kanas and Yannopoulos, 2001; Olson and Mossman, 2003; Altay and Satman, 2005), discriminant analysis (Yoon et al., 1993) and logit model (Olson and Mossman, 2003).

Various studies (Castigliano, 2001; Yao et al., 2002; Phua et al., 2003; Jasic and Wood, 2004; Weckman et al., 2008; Tilakaratne et al., 2009; Ahangar et al., 2010) have used the ANN model to capture the non-linear behaviour of stock prices in Brazil, Australia, Thailand, USA, Japan, UK, Tehran, China, India, Singapore and Hong Kong.

While ANN can be a very successful tool for time series modelling and forecasting, it has some limitations in learning the pattern (Kumar and Thenmozhi, 2009). Recently, SVM has gained popularity in forecasting the financial time series. SVM has better generalisation capability than conventional techniques and ANN and there is no increase in the number of parameters with the size of input dimension (Sivapragasam and Liong, 2000). SVM has been applied for forecasting stock prices, stock index, and futures prices (Mukherjee et al., 1997). Literature shows that SVM outperforms models like ANN in forecasting financial time series (Trafalis and Ince, 2000; Kim, 2003; Huang et al., 2005, 2005; Chen and Wang, 2007; Ding et al., 2008; Kumar and Thenmozhi, 2009). SVM outperforms discriminant analysis (Huang et al., 2005; Ding et al., 2008) and ARIMA (Chen and Wang, 2007). Literature also shows that there is evidence of SVM outperforming ANN and other benchmark models for the USA, Korea, Japan, Asian, Indian and Turkey stock markets (Trafalis and Ince, 2000; Kim, 2003; Huang et al., 2005; Chen et al., 2006; Kumar and Thenmozhi, 2009; Kara et al., 2011).

There has been an attempt to evolve a simpler and more accurate model and recently, RF method is being used in forecasting. RF model was developed by Breiman (2001), exhibits outstanding performance with regard to prediction error on a suite of benchmark

datasets. Creamer and Freund (2004) used RF technique for predicting performance and quantifying corporate governance risk in Latin American markets. Larivière and Poel (2004) used RF regression technique for investigating both customer retention and profitability outcomes. Segal (2004) used RF regression and investigated prediction performance on real-world and simulated datasets. These studies reveal that gains can be realised by additional tuning, to regulate tree size via limiting the number of splits and/or the size of nodes for which splitting is allowed.

Although ANN, RF and SVM are good forecasting models, that can overcome many of the drawbacks associated with traditional techniques, recent studies have focused on using hybrid model or combining various models of forecasting to improve the forecasting accuracy. The real world financial time series is not absolutely linear or non-linear. If the time series exhibits both linear and non-linear features, neither linear nor non-linear models individually can capture different patterns in the time series (Tugba and Casey, 2005). The idea behind the model combination is to use the unique features of each model to accurately analyse different patterns in the data and improve the forecasting accuracy (Makridakis, 1989; Clemen, 1989; Palm and Zellner, 1992; Makridakis et al., 1993). In addition, the combined model is more robust and flexible with regard to the possible structure change in the data. There have been studies which combine Kohonen self-organising map with ARIMA (Voort et al., 1996), ARIMA and ANN (Su et al., 1997), Radial basis function and ARIMA (Wedding and Cios, 1996), Econometrics and ANN (Luxhoj et al., 1996) and combining several feed forward neural network (Pelikan et al., 1992) to improve the time series forecasting accuracy. Zhang (2003) observed that most of the studies used autoregressive term as input to neural network, but the non-linear pattern will always be present in the residuals of the linear ARIMA model. Hence, he used the residuals of the ARIMA model as input to the neural network model and developed a hybrid ARIMA-ANN model. The performance of ARIMA-ANN hybrid model was compared with independent ARIMA and ANN models. Experimental results with real data sets indicated that the combined model could be effective in improving forecasting accuracy. Using Zhang's approach, Pai and Lin (2005) proposed a hybrid methodology that exploits the unique strength of the ARIMA model and the SVM model in forecasting stock prices and the results of the computational tests were promising.

On the contrary, Tugba and Casey (2005) used Zhang's (2003) approach to show that the combined forecast can underperform significantly, compared to its constituents' performances. Clemen (1989) and Granger and Ramanathan (1984) concluded that failure of the combination models may be attributed to the performance of benchmark models. If performance of the benchmark models was weaker than that of the ANN model, it is unlikely that combining relatively 'poor' models with an otherwise 'good' one will outperform the 'good' model alone. Besides, the performance of hybrid models have not been evaluated using trading performance measures using a trading strategy.

Although ARIMA-SVM (Pai and Lin, 2005) is found to outperform isolated models like ARIMA and SVM, and ARIMA-ANN (Zhang, 2003) is found to outperform isolated models like ARIMA and ANN, no attempt has been made to compare the performance of hybrid models such as hybrid ARIMA-ANN and ARIMA-SVM.

Recently, RF regression model has been used in financial forecasting. Kumar and Thenmozhi (2009) used RF model to forecast the direction of the movement of S&P CNX Nifty index and compared it with SVM, NN and other methods. They found that SVM was better than RF by only 1% in terms of hit ratio. However, there is no study

which uses RF regression to develop a hybrid models by integrating it with a linear models like ARIMA. Thus, the validity and relevance of hybrid ARIMA-ANN, ARIMA-SVM and ARIMA-RF models is yet to be explored for forecasting financial time series such as stock prices.

The other key problems associated with earlier studies are as follows. These studies (Wedding and Cio, 1996; Luxhoj et al., 1996; Voort et al., 1996; Su et al., 1997; Tseng et al., 2002; Zhang, 2003; Tugba and Casey, 2005; Pai and Lin, 2005; Chen and Wan, 2007; Maia and de Carvalho, 2011) used simulated or artificial data set for the analysis and the number of observation for training and the test data were very small. Further, in most of the studies (Zhang, 2003; Tugba and Casey, 2005; Pai and Lin, 2005; Chen and Wang, 2007; Maia and de Carvalho, 2011), the degree of accuracy and the acceptability of forecasting models are not examined by turning point forecast capability using sign and direction test. Leung et al. (2000) suggested that depending on the trading strategies adopted by investors, forecasting methods based on minimising forecast error may not be adequate to meet their objectives. Hence, competing models must be evaluated not only in terms of MAE, RMSE, etc., but also in terms of sign and direction tests. The other drawback of the earlier studies is that the models were not evaluated on the basis of trading performance. Evaluations of models using financial criteria through a trading experiment may be more appropriate.

Although previous studies focus on out-of-sample performance, most studies arbitrarily split the available data into a training (i.e., in-sample) set for model construction and a test (i.e., out-of-sample) set for model validation. This introduces bias into model selection and evaluation and the differences in model performances are likely to be a result of variations in the time frame and the number of observations used rather than assuring consistency in the results.

Although there are studies addressing the issue of forecasting financial time series such as stock market index, most of the empirical findings are associated with well-developed financial markets (UK, USA, and Japan). However, few studies exist in literature that predicts the financial time series of emerging 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.

Based on the gaps identified in literature, we seek to overcome the drawback by examining the applicability of hybrid ARIMA-ANN, ARIMA-SVM and ARIMA-RF models for predicting the daily returns of the S&P CNX Nifty Index of National Stock Exchange India, and compare the performance of three hybrid models with independent ARIMA, ANN, SVM and RF models. More specifically, we attempt to answer the following questions:

- 1 Among the machine learning methods – RF, SVM and ANN, and linear models such as ARIMA, which method is better for forecasting?
- 2 Among the hybrid ARIMA-ANN, ARIMA-SVM and ARIMA-RF models, which hybrid models is better for forecasting?
- 3 Can hybrid linear and non-linear models provide better forecast than independent models?

Our study is different from previous research in the following way. First, three different linear and non-linear hybrid models viz. ARIMA-ANN, ARIMA-SVM and ARIMA-RF

have been developed using the Zhang's approach. The performance of these hybrid models have been compared with those of the independent models such as ARIMA, RF, ANN and SVM models. Second, the seven different competing models (three hybrid and four independent models) have been rigorously compared using two approaches. We examine the out-of-sample forecasts generated by different competing models employing statistical criteria such as goodness of forecast measures [i.e., mean absolute error (MAE)], normalised mean squared error (NMSE), root mean squared error (RMSE) and proportion of times the signs of returns are correctly forecasted (Signs) and the DA test (e.g., Pesaran and Timmermann, 1992). The competing models have also been examined in terms of trading performance and economic criteria *via* a trading experiment. For example, we have used return forecasts from the different models in a simple trading strategy (buy when the forecast is positive and sell when forecast is negative) and has compared pay-offs to determine which model can serve as a useful forecasting tool. Finally, we have used a longer time-period for the time series in order to overcome the drawback of bias in the results.

3 Data and methodology

We use daily closing values of the S&P CNX Nifty index for the period 1st of January 2003 to 31st of December 2009. The data were obtained from National Stock Exchange of India Limited website (<http://www.nseindia.com>). S&P CNX Nifty is the leading index for large companies on the National Stock Exchange of India. Nifty is a diversified 50 stock index accounting for 23 sectors of the economy. It is used for a variety of purposes such as benchmarking fund portfolios, index-based derivatives and index funds. The index is a free float market capitalisation weighted index. Emerging economic, financial and foreign exchange markets are subject to more extreme external and internal shocks than their counter-parts. On the other hand emerging market economies are expected to continue experiencing rapid growth over the next few years. A robust forecasting model which can capture the pattern of the various time series is therefore essential. Hence, in this study, we examine the model for an emerging market.

The literature review suggests that there are no fixed rules to split the data into in-sample and out-of-sample. Earlier studies have used in-sample periods of 60%, 70%, 80% and 90% and the rest 40%, 30%, 20% and 10% for out-of-sample periods respectively. In this study, we used about 25% of the data for the out-of-sample period. Thus, data from 1st of January 2003 to 31st of March 2007 was used for model estimation, while the period from 1st of April 2008 to 31st of December 2009 was used for out-of-sample testing. This corresponds to the bull (2007 to March 2008) and bear period (March 2008 – Global crisis) and again the uptrend from June 2009 onwards. The out-of-sample periods can be viewed as the acid test for our proposed model. In order to overcome the problem of non-stationarity, the time series is transformed into continuously compounded returns ($r_t = \log(P_t / P_{t-1})$), where P_t and P_{t-1} represent the index price at time t and $t - 1$, respectively.

3.1 Forecasting methodology

In this section, we explain the non-linear models (ANN, SVM and RF), the linear model ARIMA and the three different hybrid models. We used only lagged returns as

explanatory variables, because the study is aimed at analysing the dynamic characteristics of returns from the stock index.

3.1.1 ARIMA methodology

ARIMA model assumes that the future values of a time series have a clear and definite functional relationship with current, past values and white noise. The mixed autoregressive model of order (p, q) denoted as ARMA (p, q) is defined as:

$$Z_t = \theta + \Phi_1 Z_{t-1} + \Phi_2 Z_{t-2} + \dots + \Phi_p Z_{t-p} + \Phi_0 a_t + \Phi_1 a_{t-1} + \Phi_2 a_{t-2} + \dots + \Phi_q a_{t-q} \quad (1)$$

where Z_t is the time series and a_t is an uncorrelated random error term with zero mean and constant variance and θ represents a constant term.

The time series models are based on the assumption that the time series involved are stationary. But time series are not always stationary, that is they are integrated. If a time series is integrated of order 1 [i.e., it is $I(1)$], and its first difference is $I(0)$, it is said to be stationary. Similarly, if a time series is $I(2)$, its second difference is $I(0)$. In general, if a time series is $I(d)$ after differencing it d times. Then $I(0)$ series will be obtained.

If a time series is differenced d times to make it stationary and then ARMA (p, q) model is applied to it, then the original time series is ARIMA (p, d, q) , that is an autoregressive integrated moving average time series.

We develop the ARIMA model using E-Views 4. The correlogram is used in identifying the significant ACFs and PACFs. The lags of ACF and PACF whose probability values are less than 5% are significant and are identified. The possible models are developed from these plots for the Nifty index returns series. The best model for forecasting is identified by considering the information criteria, i.e., Akaike Information Criteria (AIC) and Schwarz Bayesian Information Criteria (SBIC). It is also an accepted statistical paradigm that the correctly specified model for the historical data will also be the optimal model for forecasting. Hence, we compare the results of three hybrid model (ARIMA-ANN, ARIMA-SVM and ARIMA-RF) and the best SVM, ANN and RF with those of ARIMA model.

3.1.2 Neural network methodology

In this study, one of the widely used ANN models, the feed forward neural network was used for financial time series forecasting. Usually, the NN model consists of an input layer, an output layer and one or more hidden layers. The hidden layers can capture the non-linear relationship between variables. Each layer consists of multiple neurons that are connected to neurons in adjacent layers.

A neural network can be trained by the historical data of a time series in order to capture the non-linear characteristics of the specific time series. The model parameters (connection weights and node biases) will be adjusted iteratively by a process of minimising the forecasting errors. For time series forecasting, the final computational form of the ANN model is as:

$$Y_t = ao + \sum_{j=1}^q w_j f \left(a_j + \sum_{i=1}^p w_{ij} Y_{t-i} \right) + \varepsilon_t \quad (2)$$

where a_j ($j = 0, 1, 2, \dots, q$) is a bias on the j^{th} unit, and w_{ij} ($i = 1, 2, \dots, p$; $j = 1, 2, \dots, q$) is the connection weight between layers of the model, $f(\cdot)$ is the transfer function of the hidden layer, p is the number of input nodes and q is the number of hidden nodes. Actually, the ANN model in (2) performs a non-linear functional mapping from the past observation ($Y_{t-1}, Y_{t-2}, \dots, Y_{t-p}$) to the future value Y_t , i.e.,

$$Y_t = \varphi(Y_{t-1}, Y_{t-2}, Y_{t-3}, \dots, Y_{t-p}, v) + \xi_t \quad (3)$$

where v is a vector of all parameters and φ is a function determined by the network structure and connection weights. Thus, in some senses, the ANN model is equivalent to a non-linear autoregressive (NAR) model.

This study employed a three-layer feed-forward ANN to forecast daily returns of the Nifty. The most important aspect of the ANN model development is the determination of the architecture of ANN for a particular application. The choice of input variables (i.e., nodes) and hidden nodes, activation functions, the learning rate, the number of epochs, the training algorithm and termination criteria are very crucial in determining the model's performance. In this study, we used Levenberg-Marquardt algorithm to train the ANN. Theoretically, ANN models can approximate any complex function; in this context, the financial time series models can be fairly well approximated.

The experimental set-up

Kumar and Thenmozhi (2007) used ANN to forecast India's S&P CNX Nifty index. As presented below, an experimental set-up that follows their study was used to develop the ANN model.

Data

- 1 data set: daily returns on the Nifty index
- 2 selection of training and test data: as explained earlier
- 3 input variables: past one, two, three, four, five and six day returns
- 4 output variables: one-day ahead returns.

Network

- 5 Topology: The three time series use one hidden layer with the following number of nodes in each layer: (1-5) TS-(1-5) TS-1L.

Activation function: TS-Tan sigmoid, L-linear, with the number of levels input and hidden nodes experimented varying from one to five respectively. All nodes in the network are fully connected without shortcut connections.

- 6 Initialisation: Fixed randomly-generated initial weights between -1.0 and 1.0 with random distribution.
- 7 Learning rate: .01.

Training

- 8 training algorithm: Levenberg-Marquardt
- 9 termination criteria: 2,500 epochs that are fixed and linked to the data set, or the mean square error of $MSE = 0$
- 10 no. of runs of each simulation: fixed at 15.

Analysis

- 11 Performance metrics: MAE, root mean square error (RMSE), normalised mean square error (NMSE), sign and directional ratio.

In the present study, we used MATLAB 6.5 to build and train the neural network. The MATLAB program works with default parameter values of weight, assigned by the MATLAB.

3.1.3 RF regression

The RF algorithm (for both classification and regression) as proposed by Breiman (2001) was as follows: a random vector θ_k was generated, independent of the past random vectors $\theta_1, \dots, \theta_{k-1}$ but with the same distribution; and a tree was grown using the training set and θ_k , resulting in a classifier $h(x, \theta_k)$ where 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 were generated, they voted for the most popular class. This procedure is called RFs.

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

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

$$Mg(X, Y) = av_k I(h_k(X) = Y) - \max_{j \neq Y} av_k I(h_k(X) = j) \quad (4)$$

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 generalisation error is given by:

$$PE^* = P_{XY}(mg(X, Y) < 0) \quad (5)$$

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

In RFs, $h_k(X) = h(X, \theta_k)$. 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 θ_1, \dots PE^* converges to:

$$P_{XY}(P\theta(h(X, \theta) = Y) - \max_{j \neq Y} P\theta(h(X, \theta) = j) < 0) \quad (6)$$

The convergence of the above equation explains why RFs do not overfit as more trees are added, but produce a limiting value of the generalisation 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 randomise via
 - a Grow each tree on a bootstrap sample from the training data.
 - b Specify $m \ll p$ (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.

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

Breiman also extended the concept of RFs to regression cases. RFs for regression were formed by growing trees depending on a random vector such that the tree predictor takes on numerical values as opposed to class labels. The RFs predictor is formed by taking the average over a number of the trees specified by the user.

The study explored different set of initial attributes varying from 1 to 6 as an input to RF regression. A large number of trees were grown (1,500 in this case). The codes were written in MATLAB 6.5 for the implementation of RF regression.

3.1.4 Support vector machines

SVM uses linear model to implement non-linear class boundaries through some non-linear mapping of the input vectors x into the high-dimensional feature space. A linear model constructed in the new space can represent a non-linear 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. If q is 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:

$$y = f(x) = w \Phi(x) + b; \quad (7)$$

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 minimising:

$$R_{SVM_S} = C \frac{1}{n} \sum_{i=1}^n L_\varepsilon(d_i y_i) + \frac{1}{2} \|w\|^2 \quad (8)$$

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

where both C and ε are prescribed parameters. This function indicates that errors below ε are not penalised. 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 regularised term. E is called the tube size of SVMs, and C is the regularisation constant determining the trade-off between the empirical error and the regularised term. They are both user-prescribed parameters and are selected empirically and d_i is the actual stock price in the i^{th} period. Introducing the positive slack variables ζ and ζ^* which represent the distance from the actual values to the corresponding boundary values of ε tube. The equation (4) is transformed to the following constrained formation:

Minimise:

$$R(w, \zeta, \zeta^*) = \frac{1}{2} ww^T + C^* \left(\sum_{i=1}^N (\zeta + \zeta^*) \right) \quad (9)$$

Subjected to:

$$w\phi(x_i) + b_i - d_i \leq \varepsilon + \zeta_i^* \quad (10)$$

$$d_i - w\phi(x_i) - b_i \leq \varepsilon + \zeta_i \quad (11)$$

$$\begin{aligned} \zeta_i, \zeta_i^* &\geq 0 \\ i &= 1, 2, \dots, N \end{aligned} \quad (12)$$

Equation (9) can be solved by introducing a Lagrangian multiplier and maximising the dual function of equation (9).

The function $K(x_i, x_j)$ in the modified form of equation (9) 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, $K(x_i, x_j) = \Phi(x_i) \cdot \Phi(x_j)$. There are some different kernels for generating the inner products to construct machines with different types of non-linear decision surfaces in the input space. Choosing the model that minimises the estimate, among different kernels 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.

3.1.5 The hybrid methodology

Zhang (2003) considers a time series to be composed of a linear autocorrelation structure and a non-linear component. A hybrid model comprising a linear and a non-linear component as represented below was employed in the experiments (Zhang, 2003).

$$Y_t = L_t + N_t \quad (13)$$

where L_t denotes the linear component and N_t denotes the non-linear component. These two components have to be estimated from the data. These data then enter the first stage of the ARIMA to account for a linear component; hence the residuals from the linear model will contain only the non-linear relationship. If e_t denote the residual components at time t from the linear model, then

$$e_t = Y_t - \hat{L}_t \quad (14)$$

where \hat{L}_t is the forecast value for time t . Any significant non-linear pattern in the residuals will indicate the limitation of the ARIMA. By modelling residuals using ANN, non-linear relationships can be discovered. With n input nodes, the ANN model for the residuals will be:

$$e_{tANN} = f_{ANN}(e_{t-1}, e_{t-2}, \dots, e_{t-n}) + \varepsilon_{tANN} \quad (15)$$

where f_{ANN} is non-linear functions determined by the neural network and ε_{tANN} is the random error term. If the forecast from ANN is denoted as \hat{N}_{tANN} , the combined forecast will be:

$$\hat{Y}_{tANN} = \hat{L}_t + \hat{N}_{tANN} \quad (16)$$

Thus, the methodology of the hybrid system by Zhang (2003) consists of two stages. In the first stage, an ARIMA model is fitted to the time series data to capture the linear part of the problem. In the second stage, an appropriate ANN model is developed to forecast the residuals from the ARIMA model. This hybrid model exploits the unique features and strength of ARIMA model as well as ANN model in determining different patterns (Zhang, 2003). The above hybrid ARIMA-ANN model uses the following:

- a forecast residuals \hat{N}_t (results of ARIMA model) of neural network
- b the forecast \hat{L}_t (results of ARIMA model).

The optimal architecture of hybrid model that captures the non-linear patterns of residuals of ARIMA model is formed in the same way as discussed in the model formulation of neural network methodology.

Next, we developed a hybrid ARIMA-SVM model using the same principle. A SVM model was used to model the non-linearity in the residuals of the ARIMA Model. With n input nodes, the SVM model for the residuals would be:

$$e_{tSVM} = f_{SVM}(e_{t-1}, e_{t-2}, \dots, e_{t-n}) + \mu_{tSVM} \quad (17)$$

where f_{SVM} is the non-linear function determined by the SVM model and μ_{tSVM} is the random error term. If the forecast from SVM is denoted as \hat{N}_{tSVM} , the combined forecast would be:

$$\hat{F}_{tSVM} = \hat{L}_t + \hat{N}_{tSVM} \quad (18)$$

In next step, we developed ARIMA-RF model using the Zhang's approach. A RF model is used to model the non-linearity in the residuals of the ARIMA Model. With n input nodes, the RF model for the residuals would be:

$$e_{tRF} = f_{RF}(e_{t-1}, e_{t-2}, \dots, e_{t-n}) + \mu_{tRF} \quad (19)$$

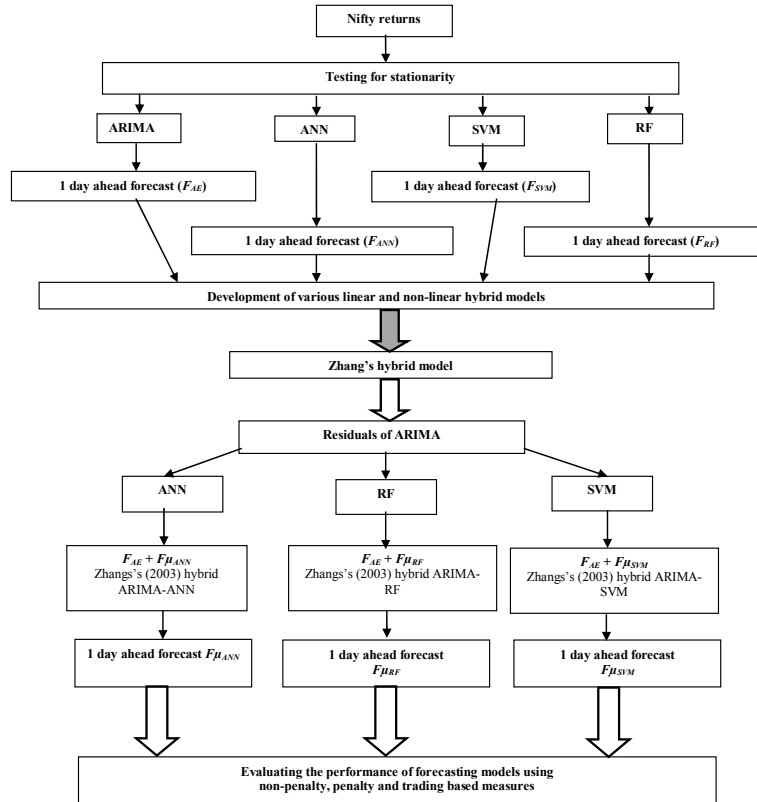
where f_{RF} is non-linear functions determined by the RF model and μ_{tRF} is the random error term. If the forecast from RF is denoted as \hat{N}_{tRF} , the combined forecast would be:

$$\hat{F}_{tRF} = \hat{L}_t + \hat{N}_{tRF} \quad (20)$$

Thus, in this study, we developed three different hybrid models to forecast the S&P CNX Nifty index return.

A flowchart explaining the development of the various models is explained in Figure 1.

Figure 1 Framework of model development



3.2 Forecasting accuracy and trading simulation

To compare the performance of the models, it is necessary to evaluate them on previously unseen data. This situation is likely to be the closest to a true forecasting or trading situation. To achieve this, all models were maintained with an identical out-of-sample period allowing a direct comparison of their forecasting accuracy and trading performance. The details of the two approaches are as follows:

3.2.1 Out-of-sample forecasting accuracy measures

We used four statistical metrics such as NMSE, MAE, RMSE and the Pesaran and Timmermann (1992) (DA) test to evaluate the forecasting capabilities among the seven models.

RMSE, NMSE and MAE measure the deviation between actual and forecast value. The smaller the values of MAE, NMSE and RMSE, the closer are the predicted time

series values to that of the actual value. It is observed that RMSE, MAE or NMSE functions that are used for financial forecasting models may not make sense in the financial context.

In financial time series, this is particularly important if decision maker is more concerned about the sign of stock return predictions than the exact value of the returns. Out-of-sample forecasts can also be evaluated by comparing the signs of the out-of-sample predictions with the true sample. Leung et al. (2000) examined the predictability of direction and sign of stock index and corroborate the importance of correct prediction of sign and direction. Thus, the correct sign forecast reflects the market timing ability of the forecasting model. The Pesaran and Timmermann (1992) developed the test of DA for out-of-sample predictions. The DA statistic was approximately distributed as standard normal, under the null hypothesis that the signs of the forecasts and the signs of the actual variables were independent. So we use the sign and directional change metrics of the Pesaran and Timmermann (1992) test to examine the directional prediction accuracy of changes, or statistical significance of market-timing ability of a model. In the present study, we used the codes written in MATLAB 6.5 for Pesaran and Timmermann test.

3.2.2 Out-of-sample trading performance measures

Statistical performance measures are often inappropriate for financial applications. Typically, modelling techniques are optimised using a mathematical criterion, but ultimately the results are analysed on a financial criterion upon which it is not optimised. In other words, the forecast error may have been minimised during model estimation, but the evaluation of the true merit should be based on the performance of a trading strategy. So, we use a simulated trading strategy to evaluate the model performance. The operational detail of the trading strategy is as follows. In the simulated market set up for experimenting the proposed methodology, a virtual trader can buy or sell stock index fund on the stock index concerned, and both short and long positions can be taken over the index.

Assume that a certain amount of seed money is used in this trading experiment. The seed money is used to buy stock index funds when the prediction shows a rise in the stock index price. To calculate the profit, the stock index funds are bought or sold at the same time. It should be noted that the price of the stock index fund is directly proportional to the index level so that the virtual investor can gain from both a fall and rise of the stock index price. The trading strategy is to go long when the model predicts that the stock index price would rise, i.e., the forecast is positive and a sell otherwise. Then the stock index funds are being held at hand until the next turning point predicted by the model.

Some of the more important trading measures include the annualised return, annualised volatility, Sharpe ratio, maximum drawdown and average gain/loss ratio. The Sharpe ratio is a risk-adjusted measure of return, with higher ratios preferred to those that are lower; the maximum drawdown is a measure of downside risk and the average gain/loss ratio is a measure of overall gain, for which a value above one is preferred.

The application of these measures may be a better standard for determining the quality of the forecasts. After all, the financial gain from a given strategy depends on trading performance, not on forecast accuracy.

4 Results

The ARIMA, ANN, RF, SVM, ARIMA-ANN, ARIMA-RF and ARIMA-SVM models are estimated for the in-sample data. The model estimation selection process is then followed by an empirical evaluation that is based on the out-sample data. The relative performance of the models is measured by four statistical measures and in terms of trading measures of performance.

4.1 ARIMA model

Before developing the ARIMA models, it is necessary to test the stationarity of the time series. ADF and Philip-Perron test statistics indicates that the daily returns for Nifty index follow a stationary series. We used the graph of the sample autocorrelation function (ACF) and the sample partial autocorrelation function (PACF) to determine the ARIMA model. However, the ARIMA model building process is more of an art rather than science. So we identify the appropriate ARIMA model for the Nifty, through experiments using the AIC and SBIC values. After considering all possible models, and looking at AIC and SBIC, the best-fit models identified for Nifty is ARIMA (2 1 3).

In order to verify the adequacy of the ARIMA model, we used one of the popular diagnostics test known as Breusch-Godfrey LM Test. Here, the test was used to check the presence of serial correlation in the residuals. The Breusch-Godfrey LM test for serial correlation of residuals as shown in Table 1 suggests that, in case of Nifty return the ARIMA model captures the entire serial correlation and the residual do not exhibit any serial correlation.

Table 1 Breusch-Godfrey serial correlation LM test for the Nifty return

<i>F-statistics</i>	<i>Probability</i>	<i>Obs. * R-square</i>	<i>Probability</i>
1.004433	0.366624	2.016921	0.364780

4.2 Neural network model

A combination of six input nodes and five hidden nodes yields a total of 30 different neural network architectures which were being considered for each in-sample training set for Nifty returns. The best network architecture thus obtained from this experiment for Nifty return on the basis of least error (MSE) associated with the model was 3-5-1, i.e., three input nodes in input layer, five nodes in hidden layer and one node in output layer. The neural network model 3-5-1 provided better fit to the Nifty returns series.

4.3 Support vector machines

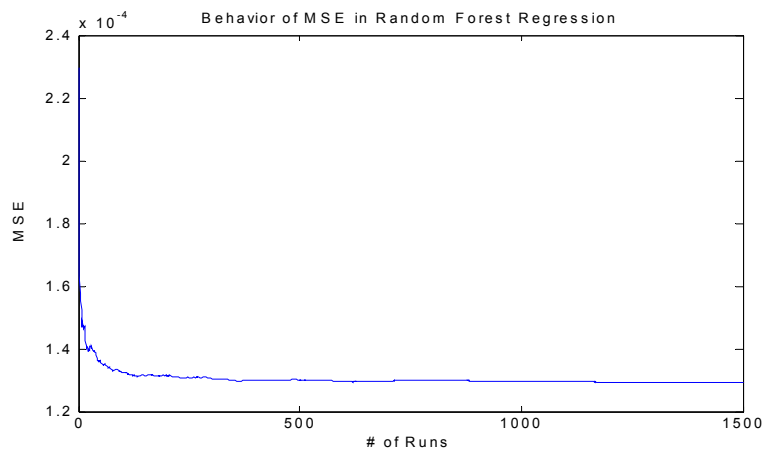
One of the advantages of linear SVM model is that there are no free parameters to tune except the constant C. For the non-linear SVM model, there is an additional parameter to tune the kernel parameter. Kim (2003) found in a preliminary test that the polynomial kernel function takes longer time to train the SVM model and provides worse results than the Gaussian radial basis function. Hence, in the present study, we use the Gaussian function as the kernel function of the SVMs similar to Tay and Cao (2001) and Kumar and Thenmozhi (2009).

To determine the parameters of SVM, we vary the parameters in spirit of Tay and Cao (2001) to select optimal values for the best prediction performance. We used different sets of experiments to find out the best performing SVM with respect to the various kernel parameters and constants. Tay and Cao (2001) suggested that small values of C caused under-fitting of the training data while too large a value of C caused over-fitting the training data. Moreover, a small value of σ^2 would overfit the training data while a large value of σ^2 would underfit the training data. Hence, we experimented with value of σ^2 and C experimented within a range of 1 to 100 for both the Nifty return series. Tay and Cao (2001) found that the performance of the SVMs is insensitive to ε when a reasonable value is selected. In their study, the value of ε was .001. So, we used the same value for ε as denoted by Tay and Cao (2001) and compared the prediction performance with respect to various kernel parameters and constants. The results are consistent with the results of Tay and Cao (2001). We solved the SVM regression problem for forecasting the returns of Nifty index and residuals using the MATLAB programme.

4.4 RF regression

The S&P CNX Nifty index return was been forecasted using the RF regression method. The RFs as proposed by Breiman (2001) are easy to use; the only two parameters that have to be determined are the number of trees to be used and the number of variables (m) to be randomly selected from the available set of variables. In both cases the study follows Breiman's recommendation to pick a large number of trees (1,500 in this case) to be used, as well as the square root of the number of variables for the latter parameter. The study experimented with different set of available variables which range from 1 to 6, and found that 3 input variables are suitable for forecasting the returns of Nifty index. Since, the number of explanatory variables equalled to 3 in this study, the number of variables randomly selected from the available set of variables was two (square root of 3 \approx 2). Figure 2 shows the graph of the error curve of RF model as the number of tree grows.

Figure 2 Behaviour of MSE in RF regression (see online version for colours)



It begins with the first tree having an MSE value of about $2.3 * 10^{-4}$. As trees grew from 0 to 200 the error rate decreases exponentially. When the number of trees was around 200 the MSE value was almost $1.35 * 10^{-4}$ and beyond that value it started decreasing. The MSE value after 900 trees was almost constant and end with an MSE value of $1.3 * 10^{-4}$.

4.5 Hybrid ARIMA-ANN model

The input to the ANN was the residuals of the fitted ARIMA model. The study experimented with input nodes which vary from 1 to 6. One hidden layer was used to develop the hybrid model. The study experimented with different nodes/neurons in hidden layer, which varies from one to five. The output layer had one neuron or node, which is the one step-ahead forecast value of the residuals. We used the LM algorithm to train the ANN and the best architecture was selected based on the MSE criteria. The best neural network thus obtained on the training set was 3-2-1, i.e., three input nodes in input layer, two nodes in hidden layer and one node in output layer. Thus, the forecast residuals obtained from ANN were added with the forecast values obtained from ARIMA model to obtain the final forecast.

4.6 Hybrid ARIMA-RF model

The input to the RF model was the residuals of the fitted ARIMA model. The study experimented with past six lags of the residuals of ARIMA model to develop ARIMA-RF model. The results suggest that 3 input variables are suitable for forecasting the residuals of the fitted ARIMA model. Here too, the number of trees grown was 1,500. The forecast residuals obtained from the RF model was added with the forecast obtained from ARIMA model to get the hybrid forecast.

4.7 Hybrid ARIMA-SVM model

The input to the SVM model was the residuals of the fitted ARIMA model. We experimented with previous six lags of residuals of ARIMA model to develop different ARIMA-SVM models for Nifty. Moreover, this study used the value of σ^2 and C experimented within a range of 1 and 100 for Nifty returns series. Further, a Gaussian function was used as kernel function of the SVM models. The least MSE was obtained for C at 12 and σ^2 at 75 for training set (in-sample data) Nifty.

4.8 Forecast evaluation

4.8.1 Out-of-sample forecasting accuracy results

For the Nifty return series one-period-ahead forecast were produced by the seven models, namely, three hybrid models, ARIMA, ANN, RF and SVM models. The predictive performance of the seven models is summarised in Table 2.

The main purpose of any financial time series modelling is to determine how well forecasts from estimated models perform based on the non-penalty-based measures of performance such as MAE, RMSE and NMSE. The forecasting accuracy statistics provide very conclusive results. These values show the superiority of ARIMA-SVM hybrid model over the two other hybrid models and the ARIMA, RF, ANN and SVM

models. Moreover, MAE, RMSE and NMSE achieved by the ARIMA-SVM hybrid model is quite low indicating that there is a small deviation between the actual and predicted values.

Table 2 Out-of-sample prediction accuracy

<i>Model</i>	<i>Performance metrics</i>			
	<i>MAE</i>	<i>RMSE</i>	<i>NMSE</i>	<i>SR (DA Stat)</i>
ARIMA-SVM	0.012158	0.017160	0.996321	0.5515***
ARIMA-NN	0.012362	0.017282	0.997689	0.5455***
ARIMA-RF	0.012408	0.017322	0.998215	0.5389***
SVM	0.012612	0.017417	0.998467	0.5363***
NN	0.012728	0.017829	1.009718	0.5263
RF	0.013645	0.018783	1.009934	0.5096
ARIMA	0.013282	0.018649	1.009861	0.5155

Note: ***Significant at 10% level.

Between the other two hybrid models, i.e., ARIMA-NN and ARIMA-RF the ARIMA-NN performs better. It is also observed that ARIMA-RF hybrid model is better than the independent NN, SVM, RF and ARIMA models.

Among the independent models, namely, ANN, SVM, RF and ARIM, the SVM model perform better and among the ANN, RF and ARIMA models, ANN model outperforms the other two. It is surprising to see that ARIMA model outperforms the non-linear RF model using these criteria. The MAE, RMSE, and NMSE of the RF regression model are the highest among the seven test models. One of the reasons for the worst performance of RF regression model may be the small number of initial attributes, three in this case. The number of variables randomly selected from the available set of variables is two (square root of $3 \approx 2$) which may be less for the RF regression problem.

In terms of other performance metrics like market timing ability (signs ratio), the ARIMA-SVM hybrid model yields better performance than the other models. The success ratio (SR) for signs of returns is correctly forecasted 55.15% in the case of one-step-ahead forecasts and is significant at 10% level. Again, RF is worst among the competing models. ARIMA-ANN outperforms ARIMA-RF and other independent models in terms of sign test and is again significant at 10% level. Among the independent models, SVM outperforms the other models in terms of direction and sign test. The results show that by combining two models together, the overall forecasting errors can be reduced considerably. A majority decision rule would therefore select the ARIMA-SVM hybrid model as the overall 'best' model followed by ARIMA-ANN, ARIMA-RF and SVM.

4.8.2 Out-of-sample trading performance results

The performance of the ARIMA-SVM model is encouraging as it has the highest ranking. However, predictability does not necessarily imply profitability. In other words, the model which maximises the direction or minimises the error may not be always profitable while using a particular trading strategy. Moreover, the ranking of forecasting models on trading performance alone will vary with the trading strategy used by the traders. Hence, in this study, an attempt was made to evaluate the performance of ARIMA-SVM which

has ranked highest vis-à-vis ARIMA-ANN, ARIMA-RF, ANN, SVM, RF and ARIMA models. A comparison of the trading performance results of the seven models is presented in Table 3.

Table 3 Trading performance results

<i>Trading measure of performance</i>	<i>ARIMA-SVM</i>	<i>ARIMA-NN</i>	<i>ARIMA-RF</i>	<i>SVM</i>	<i>NN</i>	<i>RF</i>	<i>ARIMA</i>
Annualised return (%)	22.45	20.68	17.16	16.78	12.65	4.43	7.00
Annualised volatility (%)	14.03	14.05	14.06	14.06	15.79	17.28	16.77
Sharpe ratio	1.63	1.46	1.22	1.17	1.11	0.84	0.90
Maximum drawdown (%)	-5.69	-5.65	-5.60	-5.60	-6.36	-7.14	-7.14
Avg. gain/loss ratio	1.056	1.049	1.045	1.040	1.016	1.010	1.012
Profits T-statistics	30.87	27.68	24.72	22.58	14.45	6.92	8.67
% winning up periods	54.55	53.28	51.25	47.78	52.25	45.20	51.19
% winning down periods	60.83	57.68	55.17	64.51	55.54	55.34	53.30
Buy and hold returns (%)	9.23	9.23	9.23	9.23	9.23	9.23	9.23

The result of the ARIMA-SVM hybrid model is quite impressive. It generally outperforms the other two hybrid models and different independent models in terms of overall profitability with annualised return of 22.45% annualised volatility of 14.03% and in terms of risk-adjusted performance with a Sharpe ratio of 1.63. RF regression model performs the worst both in terms of overall profitability with annualised return and in terms of risk-adjusted performance with a Sharpe ratio. All the hybrid models have the lowest downside risk as measured by maximum drawdown, while ARIMA and RF model have the highest downside risk. The ARIMA-NN model is the first runner up while ARIMA-RF is the second runner up in terms of annualised return, cumulative profit, annualised volatility and Sharpe ratio.

The SVM hybrid model predicted the highest percent of winning down periods at 64.51%, however, ARIMA-SVM predicts the highest percent winning up periods at 54.55%. Interestingly, all models are more successful in forecasting a down in the Nifty returns series, as indicated by a greater percentage of winning down periods to winning up periods. In addition, the hybrid models namely ARIMA-SVM, ARIMA-ANN and ARIMA-RF have the highest average gain/loss ratio at 1.056, 1.049, and 1.045 respectively. However, the independent models like ARIMA and RF have lowest average gain/loss ratio. The simple SVM and ANN models have the average gain/loss at 1.040 and 1.016 respectively. As with statistical performance measures, financial criteria clearly single out the ARIMA-SVM hybrid model as the one with the most consistent performance: it is therefore considered the 'best' model for this particular application.

Zhang (2003) found that hybrid ARIMA-ANN model outperforms the individual ANN and ARIMA model while Pai and Lin (2005) showed that hybrid ARIMA-SVM

model outperforms the independent ARIMA and SVM model. However, these studies use only non-penalty-based criteria (MAE, RMSE, etc.) to evaluate the forecast model. The turning point forecast capability using sign and direction test were not considered. Moreover, these studies did not evaluate their models based on the performance of trading. The present study generally supports the finding of the Zhang (2003) and Pai and Lin (2005) and contradicts the findings of Tugba and Casey (2005). The results validate the findings with real financial time series data and also by evaluating the performance of models using a trading strategy.

5 Conclusions

We developed three hybrid models (ARIMA-SVM, ARIMA-ANN and ARIMA-RF) for forecasting and trading the S&P CNX Nifty index returns. The linear ARIMA model and the non-linear ANN, SVM and RF regression model were used in combination, aiming to capture different forms of relationship in the time series data. The three hybrid models were benchmarked against traditional forecasting techniques such as ARIMA and non-linear technique like ANN, SVM and RF to determine any added value to the forecasting process. The performance of the three hybrid models and the other four independent models were measured statistically and financially via a trading experiment.

The empirical results with the Nifty returns clearly suggest that the ARIMA-SVM hybrid model is able to outperform both independent model used in isolation and the other two hybrid models used in this study in terms of MAE, RMSE and sign and DA test (Pesaran and Timmermann, 1992). It also shows that the ARIMA-SVM model outperforms the other two hybrid models and different independent models in terms of overall profitability, annualised volatility and risk-adjusted performance (Sharpe ratio). Overall, the results indicate that the hybrid ARIMA-SVM model is important in out-of-sample forecasting and trading performance, and are in line with Wedding and Cios (1996), Zhang (2003) and Pai and Lin (2005) who found that hybrid model works well and outperforms the independent models. The results contradict the findings of Tugba and Casey (2005). The superior performance of ARIMA-SVM over the other models is because SVM implements the structural risk minimisation principle which minimises an upper bound of the generalisation error rather than minimising the training error. This eventually leads to better generalisation than the neural network and RF which implements the empirical risk minimisation principle.

The present study contributes to the existing literature in several ways. First, the study uses RF regression to develop hybrid models by integrating it with a linear model like ARIMA. Secondly, the study validates the relevance of hybrid ARIMA-ANN, ARIMA-SVM and ARIMA-RF models by comparing these models for forecasting financial time series. Thirdly, the study uses different measures of performance like sign and DA to compare the performance of different models. Moreover, the present study also evaluates the different models using several financial criteria through a simple trading experiment.

The results show that there is practical implication of the forecasted index return for investing and trading on index linked funds or for the index derivatives to be traded in the derivative market. The ARIMA-SVM hybrid model could probably be used by policy makers in forecasting financial and economic data, apart from traders, investment bankers, mutual funds, FIIs and arbitrageurs. Different trading strategies can be used

based on the hybrid models and the one with the best profit/risk combination can be selected. Thus, the hybrid ARIMA-SVM model can add value to the forecasting process and assure profitable returns.

Acknowledgements

The authors would like to thank anonymous reviewers for going through the manuscript patiently and critically and sharing his/her views in improving the article. We hereby would also like to thank the editor's of the *IBJAIF* for his time, thoughtful comments and suggestions on our revised manuscript.

References

- Ahangar, R.G., Yahyazadehfar, M. and Pournaghshband, H. (2010) 'The comparison of methods artificial neural network with linear regression using specific variables for prediction stock price in Tehran Price Stock Exchange', *International Journal of Computer Science and Information Security*, Vol. 7, No. 2, pp.38–46.
- Altay, E. and Satman, M.H. (2005) 'Stock market forecasting: artificial neural network and linear regression comparison in emerging market', *Journal of Financial Management and Analysis*, Vol. 18, No. 2, pp.18–33.
- Atsalakis, G.S. and Valavanis, K.P. (2009) 'Forecasting stock market short-term trends using neuro-fuzzy based methodology', *Expert Systems with Applications*, Vol. 36, pp.10696–10707.
- Breiman, L. (2001) 'Random forests', *Machine Learning*, Vol. 45, pp.5–32.
- Castigliano, F. (2001) 'Forecasting price increments using an artificial neural networks', *Advances in Complex Systems*, Vol. 4, No. 1, pp.45–56.
- Chen, K.Y. and Wang, C.H. (2007) 'A hybrid SARIMA and support vector machines in forecasting the production values of the machinery industry in Taiwan', *Expert Systems with Applications*, Vol. 32, No. 1, pp.54–264.
- Chen, W.-H., Shih, J.-Y. and Soushan, W. (2006) 'Comparison of support-vector machines and back propagation neural networks in forecasting the six major Asian stock markets', *International Journal of Electronic Finance*, Vol. 1, No. 1, pp.49–67.
- Clemen, R. (1989) 'Combining forecasts: a review and annotated bibliography', *International Journal of Forecasting*, Vol. 5, pp.559–583.
- Crawford G.W. and Fratantoni, M.C. (2003) 'Assessing the forecasting performance of regime-switching, ARIMA and GARCH models of house prices', *Real Estate Economics*, Vol. 31, No. 2, pp.223–243.
- Creamer, G. and Freund, Y. (2004) 'Predicting performance and quantifying corporate governance risk for Latin American ADR's and banks', *Proceedings of Financial Engineering and Application Conference*, Paper No. 437-045 [online] <http://www.actapress.com> (accessed 15 April 2012).
- Desai, V.S. and Bharati, R. (1998) 'A comparison of linear regression and neural network methods for predicting excess returns on large stocks', *Annals of Operation Research*, Vol. 78, pp.127–163.
- Ding, Y.S., Song, X.P. and Zen, Y.M. (2008) 'Forecasting financial condition of Chinese listed companies based on support vector machine', *Expert Systems with Applications*, Vol. 34, No. 4, pp.3081–3089.
- Donaldson, R.G. and Kamstra, M. (1999) 'Neural network forecast combining with interaction effects', *Journal of the Franklin Institute*, Vol. 336, pp.227–236.

- Ginzburg, I. and Horn, D. (1994) 'Combined neural networks for time series analysis', *Advances in Neural Information Process. Systems*, Vol. 6, pp.224–231.
- Granger, C.W.J. and Ramanathan, R. (1984) 'Improved methods of combining forecasts', *Journal of Forecasting*, Vol. 3, pp.197–204.
- Grudnitski, G. and Osborn, L. (1993) 'Forecasting S&P and gold futures prices: an application of neural networks', *The Journal of Futures Markets*, Vol. 13, No. 6, pp.631–643.
- Harvey, C.R. (1995) 'Predictable risk and returns in emerging markets', *Review of Financial Studies*, Vol. 8, No. 3, pp.773–816.
- Hill, T., O'Connor, M. and Remus, W. (1996) 'Neural network models for time series forecasts', *Management Science*, Vol. 42, pp.1082–1092.
- Huang, W., Nakamori, Y. and Wang, S-Y. (2005) 'Forecasting stock market movement direction with support vector machine', *Computers & Operations Research*, Vol. 32, pp.2513–2522.
- Hwarng, H.B. and Ang, H.T. (2002) 'A simple neural network for ARMA (p,q) time series', *OMEGA: International Journal of Management Science*, Vol. 29, pp.319–333.
- Jasic, T. and Wood, D. (2004) 'The profitability of daily stock market indices trades based on neural network predictions: case study for the S&P 500, the DAX, the TOPIX, and the FTSE in the period 1965–1999', *Applied Financial Economics*, Vol. 14, No. 4, pp.285–297.
- Kanas, A. and Yannopoulos, A. (2001) 'Comparing linear and nonlinear forecasts for stock returns', *International Review of Economics and Finance*, Vol. 10, pp.383–398.
- Kara, Y., Boyacioglu, M.C. and Baykan, O.K. (2011) 'Predicting direction of stock price index movement using artificial neural networks and support vector machines: the sample of the Istanbul stock exchange', *Expert Systems with Applications*, Vol. 38, No. 5, pp.5311–5319.
- Kim, K.J. (2003) 'Financial time series forecasting using support vector machines', *Neurocomputing*, Vol. 55, pp.307–319.
- Kumar, M. and Thenmozhi, M. (2007) 'An experimental analysis of selected training algorithms for artificial neural networks in financial time series prediction', *International Journal of Operations and Quantitative Management*, Vol. 3, No. 1, pp.15–32.
- Kumar, M. and Thenmozhi, M. (2009) 'Forecasting stock index movement: a comparison of support vector machines and random forest', *IIMB Management Review*, Vol. 21, No. 1, pp.42–51.
- Larivière, B. and Poel, D.V.D. (2004) *Predicting Customer Retention and Profitability By Using Random Forests and Regression Forests Techniques*, Working paper, Department of Marketing, Hoveniersberg 24, 9000 Gent, Belgium.
- Lesseps, M. and Morrell, J.G. (1997) *Forecasting Exchange Rate: Theory and Practice*, The Henley Centre for Forecasting, London.
- Leung, M.T., Daouk, H. and Chen, A.S. (2000) 'Forecasting stock indices: a comparison of classification and level estimation models', *International Journal of Forecasting*, Vol. 16, pp.173–190.
- Luxhoj, J.T., Riss J.O. and Stensballe, B. (1996) 'A hybrid econometric neural network modelling approach for sales, forecasting', *International Journal of Production Economy*, Vol. 43, pp.175–192.
- Maia, A.L.S. and de Carvalho, F.A.T. (2011) 'Holt's exponential smoothing and neural network models for forecasting interval-valued time series', *International Journal of Forecasting*, Vol. 27, pp.740–759.
- Makridakis, S. (1989) 'Why combining works?', *International Journal of Forecasting*, Vol. 5, pp.601–603.
- Makridakis, S., Chateld, C., Hibdon, M., Lawrence, M., Millers, T., Ord, K. and Simmons, L.F. (1993) 'The M-2 competition: a real-life judgmentally based forecasting study', *International Journal of Forecasting*, Vol. 9, pp.5–29.

- Mukherjee, S., Osuna, E. and Girosi, F. (1997) 'Non-linear prediction of chaotic time series using support vector machines', *Proceedings of the IEEE Workshop on Neural Networks for Signal Processing*, Amelia Island, FL, pp.511–520.
- Olson, D. and Mossman, C. (2003) 'Neural network forecasts of Canadian stock returns using accounting ratios', *International Journal of Forecasting*, Vol. 19, No. 3, pp.453–465.
- Pagan, A. and Schwert, G.W. (1990) 'Alternative models for conditional stock volatility', *Journal of Econometrics*, Vol. 45, pp.267–290.
- Pai, P-F. and Lin, C-S. (2005) 'A hybrid ARIMA and support vector machines model in stock price forecasting', *Omega*, Vol. 33, No. 6, pp.497–505.
- Palm, F.C. and Zellner, A. (1992) 'To combine or not to combine? Issues of combining forecasts', *Journal of Forecasting*, Vol. 11, pp.687–701.
- Pelikan, E., de Groot, C. and Wurtz, D. (1992) 'Power consumption in West-Bohemia: improved forecasts with decorrelating connectionist networks', *Neural Network World*, Vol. 2, pp.701–712.
- Pesaran, M.H. and Timmermann, P. (1992) 'A simple nonparametric test of predictive performance', *Journal of Business and Economic Statistics*, Vol. 10, pp.461–465.
- Phua, P.K.H.; Zhu, X. and Koh, C.H. (2003) 'Forecasting stock index increments using neural networks with trust region methods', *Proceedings of the International Joint Conference on Neural Networks*, Vol. 1, pp.260–265.
- Refenes, A.N. (1993) 'Constructive learning and its application to currency exchange rate forecasting', in Trippi, R.R. and Turban, E. (Eds.): *Neural Networks in Finance and Investing: Using Artificial Intelligence to Improve Real-World Performance*, Probus Publishing Company, Chicago.
- Refenes, A.N., Zapanis, A.S. and Francis, G. (1994) 'Stock performance modeling using neural networks: comparative study with regressive models', *Neural Networks*, Vol. 7, No. 2, pp.375–388.
- Segal, M.R. (2004) *Machine Learning Benchmarks and Random Forest Regression*, Technical Report, Center for Bioinformatics & Molecular Biostatistics, University of California, San Francisco.
- Sivapragasam, C. and Liong, S.Y. (2000) 'Improving higher lead period runoff forecasting accuracy: the SVM approach', *12th Congress of the Asia and Pacific Regional Division of the IAHR*, Vol. 3, pp.1–22.
- Su, C.T., Tong, L.I. and Leou, C.M. (1997) 'Combination of time series and neural network for reliability forecasting modeling', *J. Chin. Inst. Ind. Eng.*, Vol. 14, No. 4, pp.419–429.
- Tay, F.E.H. and Cao, L. (2001) 'Application of support vector machines in financial time series forecasting', *Omega*, Vol. 29, pp.309–317.
- Tilakaratne, C.D., Mammaov, M.A. and Morris, S.A. (2009) 'Modified neural network algorithms for predicting trading signals of stock market indices', *Journal of Applied Mathematics and Decision Sciences*, No. 3, pp.1–22.
- Trafalis, T.B. and Ince, H. (2000) 'Support vector machine for regression applications to financial forecasting', *IEEE – INNS – ENNS International Joint Conference on Neural Networks*, Italy.
- Tseng, F.M. Yu, H.C. and Tzeng, G.H. (2002) 'Combining neural network model with seasonal time series ARIMA model', *Technological Forecasting and Social Change*, Vol. 69, pp.71–87.
- Tugba, T.T. and Casey, M.C. (2005) 'A comparative study of autoregressive neural network hybrids', *Neural Networks*, Vol. 18, pp.781–189.
- Virtanen, I. and Paavo, Y.O. (1987) 'Forecasting stock market prices in a thin security market', *OMEGA: International Journal of Management Science*, Vol. 15, No. 2, pp.145–155.
- Voort, V.D., Dougherty, M. and Watson, M. (1996) 'Combining Kohonen maps with ARIMA time series models to forecast traffic flow', *Transport. Research Circular*, Vol. 4C, No. 5, pp.307–318.

- Weckman, G.R., Lakshminarayanan, S., Marvel, J.H. and Snow, A. (2008) 'An integrated stock market forecasting model using neural networks', *International Journal of Business Intelligence and Management*, Vol. 1, No. 1, pp.30–49.
- Wedding, D.K. and Cios, K.J. (1996) 'Time series forecasting by combining RBF networks, certainty factors, and the Box-Jenkins model', *Neurocomputing*, Vol. 10 pp.149–168.
- Yao, J.T., Li, Y. and Tan, C.L. (2002) 'Option price forecasting using neural networks', *Omega*, Vol. 28, pp.455–466.
- Yoon, Y., Swales, G. and Margavio, T.M. (1993) 'A comparison of discriminant analysis vs artificial neural networks', *Journal of Operational Research Society*, Vol. 4, No. 1, pp.51–60.
- Yu, T.H.K. and Huarng, K.H. (2008) 'A bivariate fuzzy time series model to forecast the TAIEX', *Expert Systems with Applications*, Vol. 34, No. 4, pp.2945–2952.
- Zhang, G.P. (2003) 'Time series forecasting using a hybrid ARIMA and neural network model', *Neurocomputing*, Vol. 50, pp.59–175.
- Zhang, G., Patuwo, B.E. and Hu., M.Y. (1998) 'Forecasting with artificial neural networks: the state of the art', *International Journal of Forecasting*, Vol. 14, No. 1, pp.35–62.