



Predicting stock market index using fusion of machine learning techniques



Jigar Patel, Sahil Shah, Priyank Thakkar*, K Kotecha

Computer Science & Engineering Department, Institute of Technology, Nirma University, Ahmedabad, Gujarat, India

ARTICLE INFO

Article history:

Available online 25 October 2014

Keywords:

Artificial Neural Networks
Support Vector Regression
Random Forest
Stock market
Hybrid models

ABSTRACT

The paper focuses on the task of predicting future values of stock market index. Two indices namely CNX Nifty and S&P Bombay Stock Exchange (BSE) Sensex from Indian stock markets are selected for experimental evaluation. Experiments are based on 10 years of historical data of these two indices. The predictions are made for 1–10, 15 and 30 days in advance. The paper proposes two stage fusion approach involving Support Vector Regression (SVR) in the first stage. The second stage of the fusion approach uses Artificial Neural Network (ANN), Random Forest (RF) and SVR resulting into SVR–ANN, SVR–RF and SVR–SVR fusion prediction models. The prediction performance of these hybrid models is compared with the single stage scenarios where ANN, RF and SVR are used single-handedly. Ten technical indicators are selected as the inputs to each of the prediction models.

© 2014 Elsevier Ltd. All rights reserved.

1. Introduction and literature review

Prediction of stock prices is a classic problem. Efficient market hypothesis states that it is not possible to predict stock prices and that stocks behave in random walk manner. But technical analysts believe that most information about the stocks are reflected in recent prices and so if trends in the movements are observed then prices can be easily predicted. In addition, stock market's movements are affected by many macro-economical factors such as political events, firms' policies, general economic conditions, commodity price index, bank rate, bank exchange rate, investors' expectations, institutional investors' choices, movements of other stock markets, psychology of investors, etc. (Miao, Chen, & Zhao, 2007). Value of stock indices are calculated based on stocks with high market capitalization. Various technical parameters are used to gain statistical information from value of stocks prices. Stock indices are derived from prices of stocks with high market capitalization and so they give an overall picture of economy and depends on various factors.

There are several different approaches to time series modeling. Traditional statistical models including moving average, exponential smoothing, and ARIMA are linear in their predictions of the future values (Bollerslev, 1986; Hsieh, 1991; Rao & Gabr, 1984). Extensive research has resulted in numerous prediction

applications using Artificial Neural Networks (ANN), fuzzy logic, Genetic Algorithms (GA) and other techniques (Hadavandi, Shavandi, & Ghanbari, 2010b; Lee & Tong, 2011; Zarandi, Hadavandi, & Turksen, 2012). Artificial Neural Networks (ANN) and Support Vector Regression (SVR) are two machine learning algorithms which have been most widely used for predicting stock price and stock market index values. Each algorithm has its own way to learn patterns. Zhang and Wu (2009) incorporated the backpropagation neural network with an Improved Bacterial Chemotaxis Optimization (IBCO). They demonstrated the ability of their proposed approach in predicting stock index for both short term (next day) and long term (15 days). Simulation Results exhibited the superior performance of proposed approach. A combination of data preprocessing methods, genetic algorithms and Levenberg–Marquardt (LM) algorithm for learning feed forward neural networks was proposed in Asadi, Hadavandi, Mehmanpazir, and Nakhostin (2012). They used data preprocessing methods such as data transformation and selection of input variables for improving the accuracy of the model. The results showed that the proposed approach was able to cope with the fluctuations of stock market values and also yielded good prediction accuracy. The Artificial Fish Swarm Algorithm (AFSA) was introduced by Shen, Guo, Wu, and Wu (2011) to train radial basis function neural network (RBFNN). Their experiments on the stock indices of the Shanghai Stock Exchange indicated that RBF optimized by AFSA was an easy-to-use algorithm with considerable accuracy. Ou and Wang (2009) used total ten data mining techniques to predict price movement of Hang Seng index of Hong Kong stock market. The approaches included Linear

* Corresponding author.

E-mail addresses: 10bce067@nirmauni.ac.in (J. Patel), 10bce089@nirmauni.ac.in (S. Shah), priyank.thakkar@nirmauni.ac.in (P. Thakkar), director.it@nirmauni.ac.in (K Kotecha).

Discriminant Analysis (LDA), Quadratic Discriminant Analysis (QDA), K-nearest neighbor classification, Naive Bayes based on kernel estimation, Logit model, Tree based classification, neural network, Bayesian classification with Gaussian process, Support Vector Machine (SVM) and Least Squares Support Vector Machine (LS-SVM). Experimental results showed that the SVM and LS-SVM generated superior predictive performance among the other models. Hadavandi, Ghanbari, and Abbasian-Naghneh (2010a) proposed a hybrid artificial intelligence model for stock exchange index forecasting. The model was a combination of genetic algorithms and feed forward neural networks.

Recently, the support vector machine (SVM) (Vapnik, 1999) has gained popularity and is regarded as a state-of-the-art technique for regression and classification applications. Kazem, Sharifi, Hussain, Saberi, and Hussain (2013) proposed a forecasting model based on chaotic mapping, firefly algorithm, and Support Vector Regression (SVR) to predict stock market price. SVR-CFA model which was newly introduced in their study, was compared with SVR-GA (Genetic Algorithm), SVR-CGA (Chaotic Genetic Algorithm), SVR-FA (Firefly Algorithm), ANN and ANFIS models and the result showed that SVR-CFA model was performing better than other models. Pai, Lin, Lin, and Chang (2010) developed a Seasonal Support Vector Regression (SSVR) model to forecast seasonal time series data. Hybrid genetic algorithms and tabu search (GA/TS) algorithms were applied in order to select three parameters of SSVR models. They also applied two other forecasting models, Seasonal Autoregressive Integrated Moving Average (SARIMA) and SVR for forecasting on the same data sets. Empirical results indicated that the SSVR outperformed both SVR and SARIMA models in terms of forecasting accuracy. By integrating genetic algorithm based optimal time-scale feature extractions with support vector machines, Huang and Wu (2008) developed a novel hybrid prediction model that operated for multiple time-scale resolutions and utilized a flexible nonparametric regressor to predict future evolutions of various stock indices. In comparison with neural networks, pure SVMs and traditional GARCH models, the proposed model performed the best. The reduction in root-mean-squared error was significant. Financial time series prediction using ensemble learning algorithms in Cheng, Xu, and Wang (2012) suggested that ensemble algorithms were powerful in improving the performances of base learners. The study by Aldin, Dehnavr, and Entezari (2012) evaluated the effectiveness of using technical indicators, such as Moving Average, RSI, CCI, MACD, etc. in predicting movements of Tehran Exchange Price Index (TEPIX).

This paper focuses on the task of predicting future values of stock market indices. The predictions are made for 1–10, 15 and 30 days in advance. It has been noticed from the literature that the existing methods for the task under focus in this paper employ only one layer of prediction which takes the statistical parameters as inputs and gives the final output. In these existing methods, the statistical parameters' value of $(t)^{\text{th}}$ day is used as inputs to predict the $(t + n)^{\text{th}}$ day's closing price/value (t is a current day). It is felt that in such scenarios, as the value of n increases, predictions are based on increasingly older values of statistical parameters and thereby not accurate enough. It is clear from this discussions that there is a need to address this problem, and that motivated us about two stage prediction scheme which can bridge this gap and minimize the error stage wise. It was thought that success of the two stage proposed model could really be the significant contribution to the research as the approach can be generalized for other prediction tasks such as whether forecasting, energy consumption forecasting, GDP forecasting etc.

The two stage fusion approach proposed in this paper involves Support Vector Regression (SVR) in the first stage. The second stage of the fusion approach uses Artificial Neural Network (ANN), Random Forest (RF) and SVR resulting into SVR-ANN, SVR-RF and

SVR-SVR prediction models. The prediction performance of these hybrid models is compared with the single stage scenarios where ANN, RF and SVR are used single-handedly.

The remainder of the paper is organized into following sections. Section 2 describes single stage approach while focus of the Section 3 is proposed two stage approach. Section 4 addresses experimental results and discussions on these results. Section 5 ends with the concluding remarks.

2. Single stage approach

The basic idea of a single stage approach is illustrated in Fig. 1. It can be seen that for the prediction task of n^{th} day ahead of time, inputs to prediction models are ten technical indicators describing t^{th} day while the output is $(t + n)^{\text{th}}$ day's closing price. These technical indicators which are used as inputs are summarized in Table 1. The prediction models employed are described in the following sub-sections.

2.1. Artificial Neural Networks

Three layer feed forward back propagation ANN as shown in Fig. 2 is employed in this paper. Input layer has ten neurons, one for each of the selected technical parameters. The value of the index which is to be predicted is represented by the only neuron in the output layer. Adaptive gradient descent is used as the weight update algorithm. A tangent sigmoid is used as the transfer function of the neurons of the hidden layer while the neuron in the output layer uses linear transfer function. The output of the model is a continuous value signifying the predicted value of the index.

The reason behind using adaptive gradient descent is to allow learning rate to change during the training process. It may improve the performance of the gradient descent algorithm. In adaptive gradient descent, first, the initial network output and error are calculated. The current learning rate is used to calculate new weights and biases at each epoch. Based on these new weights and biases, new outputs and errors are calculated. If the new error exceeds the old error by more than a predefined ratio (1.04, in this study), the new weights and biases are discarded. Also the learning rate is decreased (to 70% of its current value, in this study). Otherwise, new weights and biases are kept and the learning rate is increased (by 5% of the current value, in the experiments reported in this paper).

The procedure ensures that the learning rate is increased only to the extent that the network can learn without large increases in error. This allows to obtain near optimal learning rate for the local terrain. At the same time, as long as stable learning is assured, learning rate is increased. When it is too high to assure a decrease in error, it is decreased until stable learning resumes.

Number of neurons in the hidden layer and number of epochs are considered as the design parameters of the model. Comprehensive number of experiments are carried out by varying the parameter values as shown in Table 2.

2.2. Support Vector Regression

The SVR uses the same principles as the SVM for classification, with only a few minor differences. First of all, because output is a real number it becomes very difficult to predict the information at hand, which has infinite possibilities. In the case of regression, a margin of tolerance ϵ is set in approximation to the SVM. Up until the threshold ϵ , the error is considered 0. However, the main idea is always the same: to minimize error, individualizing the hyper plane which maximizes the margin, considering that part of the error is tolerated (Parrella, 2007).

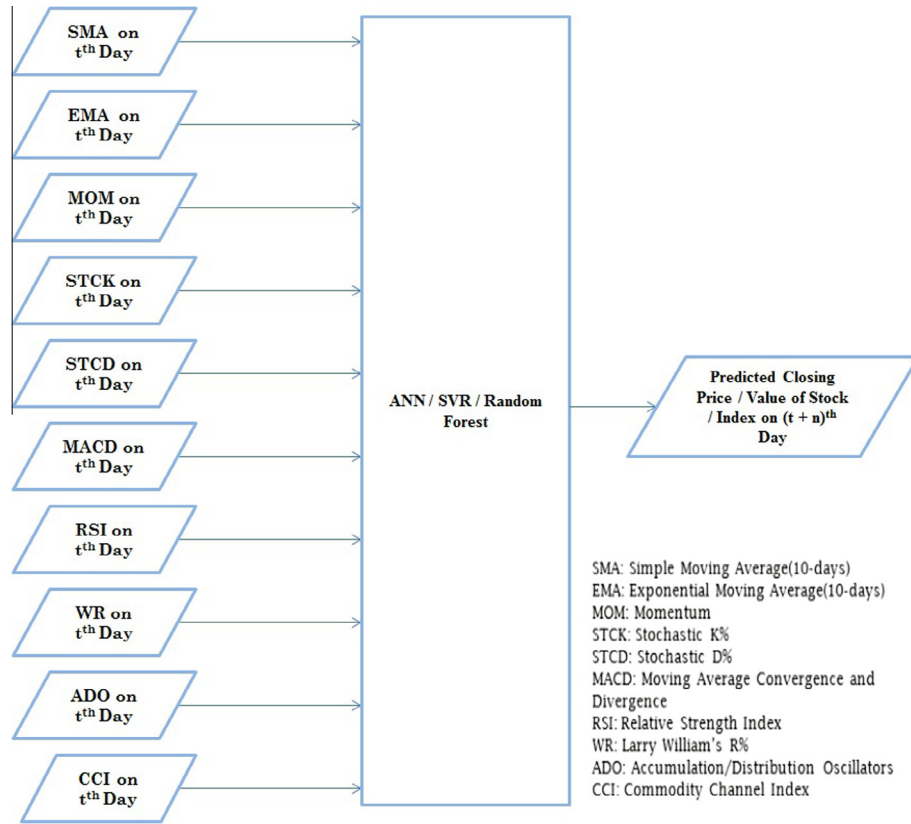


Fig. 1. General architecture of single stage approach for predicting n day ahead of time.

Table 1

Selected technical indicators & their formulas (Kara et al., 2011).

Name of indicators	Formulas
Simple n (10 here)-day moving average	$\frac{C_t + C_{t-1} + \dots + C_{t-n}}{n}$
Weighted 10-day moving average	$\frac{(10)C_t + (9)C_{t-1} + \dots + C_{t-9}}{n + (n-1) + \dots + 1}$
Momentum	$C_t - C_{t-9}$
Stochastic K%	$\frac{C_t - LL_{t-(n-1)}}{HH_{t-(n-1)} - LL_{t-(n-1)}} \times 100$
Stochastic D%	$\frac{\sum_{i=0}^{n-1} K_{t-i}}{10} \%$
Relative strength index (RSI)	$100 - \frac{100}{1 + (\sum_{i=0}^{n-1} UP_{t-i}/n) / (\sum_{i=0}^{n-1} DW_{t-i}/n)}$
Moving average convergence divergence (MACD)	$MACD(n)_{t-1} + \frac{2}{n+1} \times (DIFF_t - MACD(n)_{t-1})$
Larry William's R%	$\frac{H_n - C_t}{H_n - L_n} \times 100$
A/D (Accumulation/Distribution) oscillator	$\frac{H_t - C_{t-1}}{H_t - L_t}$
CCI (Commodity channel index)	$\frac{M_t - SM_t}{0.015D_t}$

C_t is the closing price, L_t is the low price and H_t the high price at time t , $DIFF_t = EMA(12)_t - EMA(26)_t$, EMA is exponential moving average, $EMA(k)_t = EMA(k)_{t-1} + \alpha \times (C_t - EMA(k)_{t-1})$, α is a smoothing factor which is equal to $\frac{2}{k+1}$, k is the time period of k -day exponential moving average, LL_t and HH_t implies lowest low and highest high in the last t days, respectively. $M_t = \frac{H_t + L_t + C_t}{3}$, $SM_t = \frac{(\sum_{i=1}^n M_{t-i+1})}{n}$, $D_t = \frac{(\sum_{i=1}^n |M_{t-i+1} - SM_t|)}{n}$, UP_t means upward price change while DW_t is the downward price change at time t .

The basic concepts of SVR which are discussed here can also be found in Cristianini and Shawe-Taylor (2000), Kecman (2001) and Huang and Tsai (2009). Assume that $x_i \in R^d$, $i = 1, 2, \dots, m$ forms a set of input vectors with corresponding response variable $y_i \in R$, $i = 1, 2, \dots, m$. SVR builds the linear regression function as shown in Eq. (1).

$$f(x, w) = w^T x + b \quad (1)$$

Eq. (2) shows Vapnik's linear ε -Insensitivity loss function.

$$|y - f(x, w)|_\varepsilon = \begin{cases} 0, & \text{if } |y - f(x, w)| \leq \varepsilon \\ |y - f(x_i, w)| - \varepsilon, & \text{otherwise} \end{cases} \quad (2)$$

Based on this, linear regression $f(x, w)$ is estimated by simultaneously minimizing $\|w\|^2$ and the sum of the linear ε -Insensitivity losses as shown in Eq. (4). The constant c controls a trade-off between an approximation error and the weight vector norm $\|w\|$.

$$R = \frac{1}{2} \|w\|^2 + c \left(\sum_{i=1}^m |y - f(x, w)|_\varepsilon \right) \quad (3)$$

Minimizing the risk R is equivalent to minimizing the risk shown in Eq. (4) under the constraints illustrated in Eq. (5)–(7). Here, ξ_i and ζ_i^* are slack variables, one for exceeding the target value by more than ε and other for being more than ε below the target.

$$R = \frac{1}{2} \|w\|^2 + c \sum_{i=1}^m (\xi_i + \zeta_i^*) \quad (4)$$

$$(w^T x_i + b) - y_i \leq \varepsilon + \xi_i \quad (5)$$

$$y_i - (w^T x_i + b) \leq \varepsilon + \zeta_i^* \quad (6)$$

$$\xi_i, \zeta_i^* \geq 0, \quad i = 1, 2, \dots, m \quad (7)$$

Similar to SVM, above constrained optimization problem is solved using Lagrangian theory and the Karush–Kuhn–Tucker conditions to obtain the desired weight vector of the regression function.

SVR can map the input vectors $x_i \in R^d$ into a high dimensional feature space $\Phi(x_i) \in H$. A kernel function $K(x_i, x_j)$ performs the mapping $\phi(\cdot)$. The polynomial and radial basis kernel functions are used in this study and they are shown in Eqs. (8) and (9)

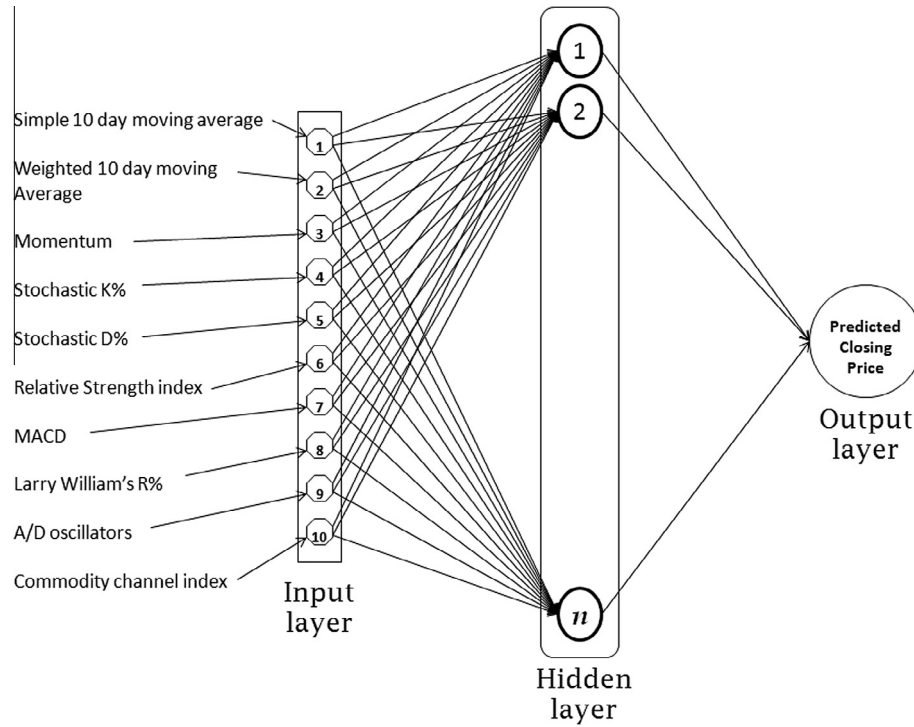


Fig. 2. ANN model.

Table 2
ANN parameters and their values tested.

Parameters	Level(s)
Number of hidden layer neurons (n)	10, 20, ..., 100
Epochs (ep)	1000, 2000, ..., 10,000

Table 3
SVR parameters and their values tested.

Parameters	Level(s)
Degree of kernel function (d)	1, 2, 3, 4
Gamma in kernel function (γ)	0, 0.5, 1, 1.5, 2, 2.5, 3, 4, 5, 10, 20, 50, 100
Regularization parameter (c)	1

respectively. Ten technical indicators (Table 1) are given as inputs to SVR which enables it to predict the future closing price.

$$\text{PolynomialFunction} : K(x_i, x_j) = (x_i \cdot x_j + 1)^d \quad (8)$$

$$\text{RadialBasisFunction} : K(x_i, x_j) = \exp(-\gamma \|x_i - x_j\|^2) \quad (9)$$

where, d is the degree of polynomial function and γ is the constant of radial basis function.

Choice of kernel function, degree of kernel function (d) in case of polynomial kernel, gamma in kernel function (γ) in case of radial basis kernel and regularization constant (c) are considered as the design parameters of SVR. Comprehensive number of experiments are carried out by varying the parameter values as shown Table 3.

2.3. Random Forest

One of the most popular technique for classification is decision tree learning. It is very efficient and at the same time its classification accuracy is comparable with other classification methods. The

tree which is popularly called as the decision tree is used to represent the classification model learnt through these techniques. Decision tree can also be learnt as the regression tree and it is also very efficient and exhibits low error values. Examples of decision tree learning algorithms are ID3 (Quinlan, 1986), C4.5 (Quinlan, 1993) and CART (Breiman, Friedman, Stone, & Olshen, 1984). Details can be found in Han, Kamber, and Pei (2006).

Random Forest belongs to the category of ensemble learning algorithms. Decision tree is used by it as the base learner of the ensemble. In this paper, the base learner is actually a regression tree. A single regression tree is not accurate enough in determining predicted value of a dependent variable is the basic idea of ensemble learning. Reason being, based on sample data, regressor is not able to distinguish between noise and pattern. So, sampling with replacement is performed such that given n trees to be learnt are based on these data set samples. Also, in the experiments performed in this study, each tree is learnt using 3 randomly selected features. After creation of n trees, when testing data is used, the average of predicted value by each of the tree in the ensemble is considered as the final predicted value of the dependent variable. This also eludes problem of over-fitting. Our implementation of Random Forest algorithm is summarized in the Algorithm 1.

Algorithm 1. Our implementation of Random Forest

Input: training set D , number of trees in the ensemble k

Output: a composite model M^*

```

1: for  $i = 1$  to  $n$  do
2:   Create bootstrap sample  $D_i$  by sampling  $D$  with replacement.
3:   Select 3 features randomly.
4:   Use  $D_i$  and randomly selected three features to derive a regression tree  $M_i$ .
5: end for
6: return  $M^*$ .
```

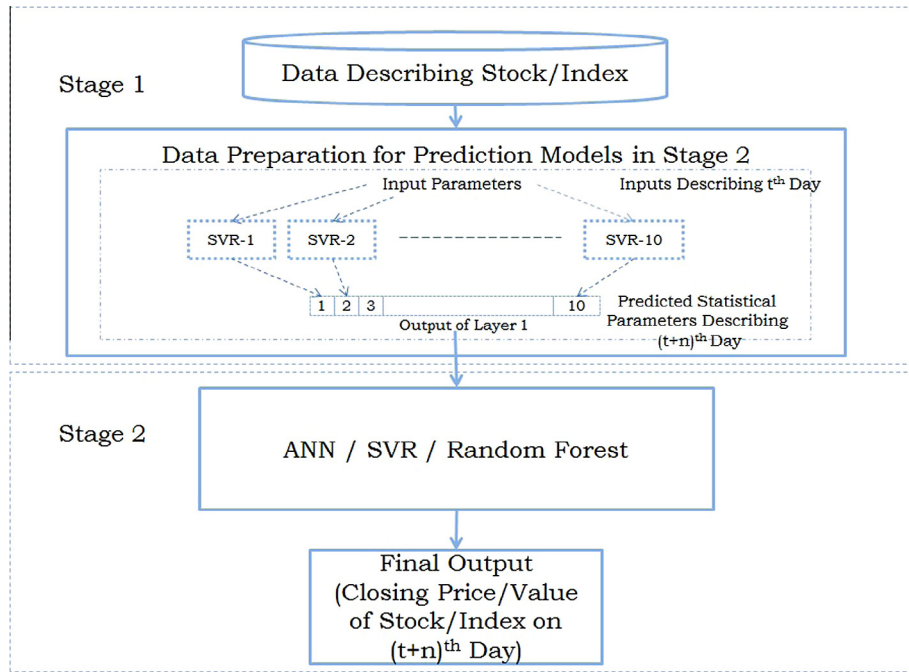



Fig. 3. General architecture of two stage fusion approach for predicting n day ahead of time.

Number of trees (n) in the ensemble is considered as the design parameter of Random Forest. Experiments are carried out with 50, 100 and 150 number of trees.

3. Two stage fusion approach

The basic idea of two stage fusion approach is illustrated in Fig. 3. The first stage employs SVRs to prepare inputs for the prediction models in the second stage.

Details about inputs and outputs to these SVRs, for the prediction task of n^{th} day ahead of time, are depicted in Fig. 4.

It is to be noticed that inputs to the SVRs in the first stage describe t^{th} day while outputs of this stage describe $(t+n)^{\text{th}}$ day in terms of ten technical indicators. These outputs from the first

stage serve as the inputs to the prediction models in the second stage. This leads to the situation where prediction models in the second stage have to identify mapping transformation from technical parameters describing $(t+n)^{\text{th}}$ day to $(t+n)^{\text{th}}$ day's closing price. This is different from single stage approach, where, prediction models have to identify mapping transformation from technical parameters describing t^{th} day to $(t+n)^{\text{th}}$ day's closing price. It can be seen that the final output in both the approaches is the closing price/value of the $(t+n)^{\text{th}}$ day. As shown in Figs. 3, 4, ANN, SVR and Random Forest are employed as the prediction models in the second stage. Comprehensive number of experiments for each of the prediction models in second stage are carried out by varying the parameter values in the same manner as in the single stage approach. However, to decide best combination of parameter

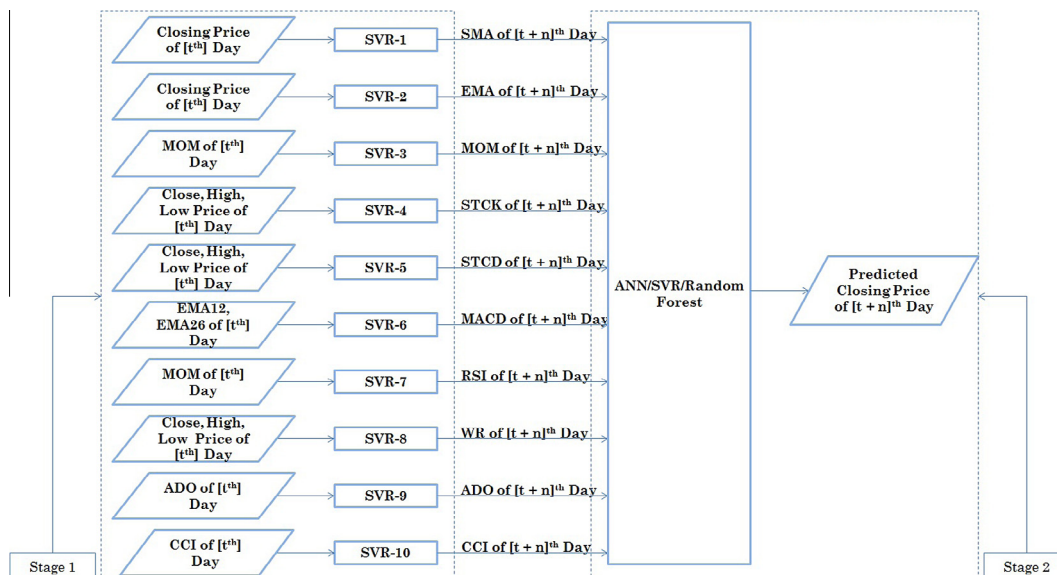


Fig. 4. Details of two stage fusion approach for predicting n day ahead of time.

values for each of the SVRs in the first stage, a parameter tuning data set is formed as the 20% data of the entire data set. The parameter tuning data set is further divided into training and testing set. Training data set consists of 80% of the parameter tuning data set while remaining of the parameter tuning data form the testing data set. By means of experiments on these training and testing set, best combination of parameter values for each of the SVRs in the first stage is identified. We call these experiments as parameter tuning experiments for the SVRs in the first stage. The possible values which are considered for each of the parameters of these SVRs are same as shown in Table 3.

Results of parameter tuning experiments for each of the SVRs in the first stage show that transformation of input space through RBF kernel performs better than the transformation through polynomial kernel. The best parameter combinations as reported by parameter tuning experiments for each of the SVRs in the first stage are summarized in Table 4. It is to be noticed that the aim of the parameter tuning experiments is to identify best parameter combination for each of the SVRs in first stage, so that, error in statistical parameters which are to be used as inputs to the prediction models in second stage is minimized. During the overall experiments of price/value predictions, SVRs in the first stage are used with the parameters determined during the parameter tuning experiments.

4. Experimental results

4.1. Data for experimentation

This study uses total ten years of historical data from Jan 2003 to Dec 2012 of two stock market indices CNX Nifty and S&P BSE Sensex which are highly voluminous. The ten technical indicators used are calculated from close, high, low and opening prices of these indices. All the data is obtained from <http://www.nseindia.com/> and <http://www.bseindia.com/> websites.

4.2. Evaluation measures

Mean Absolute Percentage Error (MAPE), Mean Absolute Error (MAE), relative Root Mean Squared Error (rRMSE) and Mean Squared Error (MSE) are used to evaluate the performance of these prediction models. Formulas of these evaluation measures are shown in Eqs. (10)–(13).

$$MAPE = \frac{1}{n} \sum_{t=1}^n \frac{|A_t - F_t|}{|A_t|} \times 100 \quad (10)$$

$$MAE = \frac{1}{n} \sum_{t=1}^n \frac{|A_t - F_t|}{|A_t|} \quad (11)$$

$$rRMSE = \sqrt{\frac{1}{n} \sum_{t=1}^n \left(\frac{A_t - F_t}{A_t} \right)^2} \quad (12)$$

$$MSE = \frac{1}{n} \sum_{t=1}^n (A_t - F_t)^2 \quad (13)$$

where A_t is actual value and F_t is forecast value.

4.3. Results

Three prediction models namely ANN, SVR and RF are used in single stage approach. In two stage fusion approach, the prediction models that are used are SVR-ANN, SVR-SVR and SVR-RF. For both the approaches, prediction experiments, for 1–10, 15 and 30 days ahead of time are carried out. Results for CNX Nifty are shown in Tables 5–16. Similar results for S&P BSE Sensex are depicted in Tables 17–28.

Table 4

Best parameter combination reported by parameter tuning experiments for each of the SVRs in first stage of two stage fusion approach.

SVR	Kernel function	γ
SVR-1	RBF	2
SVR-2	RBF	2
SVR-3	RBF	2
SVR-4	RBF	100
SVR-5	RBF	100
SVR-6	RBF	10
SVR-7	RBF	4
SVR-8	RBF	100
SVR-9	RBF	2
SVR-10	RBF	1.5

Table 5

Prediction performance of 1-day ahead of time (For CNX NIFTY).

Prediction models	Parameters	Error measures			
		MAPE	MAE	rRMSE	MSE
	<i>Epochs</i>				
ANN	3000	1.91	102.06	2.48	17745.90
SVR-ANN	7000	1.50	79.05	1.93	10006.12
	<i>Gamma</i>				
SVR	4.00	0.99	52.48	1.26	4427.05
SVR-SVR	5.00	1.47	77.63	1.87	9614.30
	<i>ntrees</i>				
Random Forest	50	1.36	72.45	1.68	8086.79
SVR-Random Forest	150	1.29	69.01	1.64	7710.16

Table 6

Prediction performance of 2-days ahead of time (For CNX NIFTY).

Prediction models	Parameters	Error measures			
		MAPE	MAE	rRMSE	MSE
	<i>Epochs</i>				
ANN	3000	1.92	101.79	2.42	16399.21
SVR-ANN	10,000	1.66	87.82	2.13	12299.75
	<i>Gamma</i>				
SVR	2.50	1.40	74.15	1.78	8748.11
SVR-SVR	5.00	1.61	85.14	2.09	12104.73
	<i>ntrees</i>				
Random Forest	50	1.80	95.69	2.24	14206.36
SVR-Random Forest	150	1.55	82.34	1.96	10832.60

Table 7

Prediction performance of 3-days ahead of time (For CNX NIFTY).

Prediction models	Parameters	Error measures			
		MAPE	MAE	rRMSE	MSE
	<i>Epochs</i>				
ANN	6000	2.16	113.79	2.75	20668.78
SVR-ANN	4000	1.86	98.25	2.38	15736.44
	<i>Gamma</i>				
SVR	0.00	1.76	93.13	2.22	13556.21
SVR-SVR	5.00	1.86	98.58	2.39	15833.27
	<i>ntrees</i>				
Random Forest	150	2.12	112.78	2.67	20043.69
SVR-Random Forest	100	1.93	102.07	2.42	16355.13

It is important to notice that for each of the prediction tasks and prediction models, comprehensive number of experiments are carried out, for different possible combinations of model parameters. The values reported in the tables are best parameter combination where minimum prediction error is exhibited.

Table 8

Prediction performance of 4-days ahead of time (For CNX NIFTY).

Prediction models	Parameters	Error measures			
		MAPE	MAE	rRMSE	MSE
ANN SVR-ANN	<i>Epochs</i> 4000	2.49	131.92	3.10	26636.56
	2000	2.12	112.14	2.72	20455.22
SVR SVR-SVR	<i>Gamma</i> 0.00	2.08	109.66	2.59	18445.44
	5.00	2.06	108.64	2.62	19013.75
Random Forest SVR-Random Forest	<i>ntrees</i> 50	2.40	127.34	2.97	24734.84
	100	2.12	112.55	2.69	20067.88

Table 9

Prediction performance of 5-days ahead of time (For CNX NIFTY).

Prediction models	Parameters	Error measures			
		MAPE	MAE	rRMSE	MSE
ANN SVR-ANN	<i>Epochs</i> 2000	2.85	151.81	3.57	36016.34
	2000	2.32	121.91	2.96	23562.53
SVR SVR-SVR	<i>Gamma</i> 4.00	2.34	123.77	2.92	23455.13
	5.00	2.26	119.59	2.86	22583.42
Random Forest SVR-Random Forest	<i>ntrees</i> 100	2.62	139.01	3.30	30370.98
	50	2.39	126.86	3.00	24975.76

Table 10

Prediction performance of 6-days ahead of time (For CNX NIFTY).

Prediction models	Parameters	Error measures			
		MAPE	MAE	rRMSE	MSE
ANN SVR-ANN	<i>Epochs</i> 7000	2.80	149.35	3.47	34308.21
	5000	2.48	130.52	3.16	26803.00
SVR SVR-SVR	<i>Gamma</i> 5.00	2.57	135.44	3.22	28502.25
	5.00	2.46	130.04	3.08	26297.36
Random Forest SVR-Random Forest	<i>ntrees</i> 50	2.78	147.98	3.55	35171.38
	50	2.61	138.29	3.23	28782.80

Table 11

Prediction performance of 7-days ahead of time (For CNX NIFTY).

Prediction models	Parameters	Error measures			
		MAPE	MAE	rRMSE	MSE
ANN SVR-ANN	<i>Epochs</i> 9000	3.02	160.20	3.83	41725.69
	9000	2.65	139.58	3.36	30552.47
SVR SVR-SVR	<i>Gamma</i> 10.00	2.74	144.93	3.48	33231.86
	5.00	2.61	137.81	3.33	30462.37
Random Forest SVR-Random Forest	<i>ntrees</i> 50	3.08	164.00	3.96	44032.01
	100	2.84	150.38	3.55	34948.60

Table 12

Prediction performance of 8-days ahead of time (For CNX NIFTY).

Prediction models	Parameters	Error measures			
		MAPE	MAE	rRMSE	MSE
ANN SVR-ANN	<i>Epochs</i> 3000	3.01	160.14	3.80	40747.70
	8000	2.82	149.03	3.60	35377.97
SVR SVR-SVR	<i>Gamma</i> 0.00	2.90	153.08	3.69	37239.22
	4.00	2.77	145.99	3.55	34515.93
Random Forest SVR-Random Forest	<i>ntrees</i> 50	3.33	177.80	4.25	51119.66
	50	2.90	153.03	3.68	37168.41

Table 13

Prediction performance of 9-days ahead of time (For CNX NIFTY).

Prediction models	Parameters	Error measures			
		MAPE	MAE	rRMSE	MSE
ANN SVR-ANN	<i>Epochs</i> 1000	3.36	178.61	4.22	50724.35
	4000	3.00	157.63	3.82	39622.51
SVR SVR-SVR	<i>Gamma</i> 0.00	3.08	162.50	3.92	41815.60
	4.00	2.94	154.96	3.77	38621.40
Random Forest SVR-Random Forest	<i>ntrees</i> 150	3.58	190.76	4.56	58407.62
	50	3.04	160.58	3.87	41177.90

Table 14

Prediction performance of 10-days ahead of time (For CNX NIFTY).

Prediction models	Parameters	Error measures			
		MAPE	MAE	rRMSE	MSE
ANN SVR-ANN	<i>Epochs</i> 6000	3.54	188.73	4.61	60592.10
	7000	3.23	170.40	4.15	46788.00
SVR SVR-SVR	<i>Gamma</i> 0.00	3.24	170.61	4.15	46764.19
	4.00	3.11	163.74	3.99	43197.66
Random Forest SVR-Random Forest	<i>ntrees</i> 150	3.73	198.62	4.81	64653.82
	150	3.26	172.23	4.15	47132.72

Table 15

Prediction performance of 15-days ahead of time (For CNX NIFTY).

Prediction models	Parameters	Error measures			
		MAPE	MAE	rRMSE	MSE
ANN SVR-ANN	<i>Epochs</i> 8000	4.05	215.26	5.06	71431.74
	7000	3.75	195.95	4.87	62369.57
SVR SVR-SVR	<i>Gamma</i> 0.00	4.04	212.36	5.09	69934.62
	4.00	3.83	201.51	4.87	63747.90
Random Forest SVR-Random Forest	<i>ntrees</i> 50	4.12	217.61	5.49	82312.31
	50	3.82	201.15	4.86	63754.16

It is evident from the result that as predictions are made for more and more number of days in advance, error values increase. This may be obvious for any prediction system. Proposed two stage fusion models SVR-ANN and SVR-RF outperform ANN and RF models for almost all prediction tasks for both the data sets.

SVR-SVR outperforms SVR for all the prediction tasks except for the prediction tasks up to 3–4 days in advance.

Tables 29, 30 compares performance of single stage models to two stage fusion models for CNX Nifty. The reported values in these tables are averaged over all 12 prediction tasks (1–10, 15

Table 16
Prediction performance of 30-days ahead of time (For CNX NIFTY).

Prediction models	Parameters	Error measures			
		MAPE	MAE	rRMSE	MSE
ANN SVR-ANN	<i>Epochs</i> 7000	5.02	267.49	6.21	109479.02
	7000	4.56	237.59	5.79	86912.03
SVR SVR-SVR	<i>Gamma</i> 0.00	5.32	278.37	6.82	124246.62
	4.00	4.94	258.41	6.26	103710.57
Random Forest SVR-Random Forest	<i>ntrees</i> 150	5.26	276.87	6.96	130770.09
	50	4.88	255.23	6.19	101094.69

Table 17
Prediction performance of 1-Day Ahead of Time (For S&P BSE SENSEX).

Prediction models	Parameters	Error measures			
		MAPE	MAE	rRMSE	MSE
ANN SVR-ANN	<i>Epochs</i> 3000	1.78	313.92	2.31	166090.16
	7000	1.55	272.71	1.96	118395.09
SVR SVR-SVR	<i>Gamma</i> 4.00	0.98	172.47	1.25	47558.47
	0.50	1.48	260.05	1.89	108137.61
Random Forest SVR-Random Forest	<i>ntrees</i> 100	1.25	221.91	1.60	81098.60
	50	1.23	216.02	1.55	73483.60

Table 18
Prediction performance of 2-days ahead of time (For S&P BSE SENSEX).

Prediction models	Parameters	Error measures			
		MAPE	MAE	rRMSE	MSE
ANN SVR-ANN	<i>Epochs</i> 7000	1.92	338.77	2.40	179261.33
	3000	1.69	296.41	2.20	146339.70
SVR SVR-SVR	<i>Gamma</i> 10.00	1.38	241.46	1.75	93134.99
	0.50	1.59	278.92	2.07	131058.24
Random Forest SVR-Random Forest	<i>ntrees</i> 100	1.66	292.50	2.08	134559.91
	150	1.62	285.74	2.01	125881.22

Table 19
Prediction performance of 3-days ahead of time (For S&P BSE SENSEX).

Prediction models	Parameters	Error measures			
		MAPE	MAE	rRMSE	MSE
ANN SVR-ANN	<i>Epochs</i> 1000	2.15	378.59	2.75	233075.75
	9000	2.02	354.01	2.64	212027.62
SVR SVR-SVR	<i>Gamma</i> 0.00	1.75	306.11	2.21	148141.01
	0.50	1.85	324.22	2.37	171209.68
Random Forest SVR-Random Forest	<i>ntrees</i> 100	2.01	353.69	2.54	198459.92
	150	1.89	332.20	2.35	169694.03

and 30 days in advance). Similar results for S&P BSE Sensex are summarized in [Tables 31 and 32](#). [Tables 29, 31](#) compare single stage prediction models to two stage fusion models on the basis of average prediction performance while [Tables 30, 32](#) show percentage improvement in performance achieved by two stage fusion prediction models over single stage prediction models.

Table 20
Prediction performance of 4-days ahead of time (For S&P BSE SENSEX).

Prediction models	Parameters	Error measures			
		MAPE	MAE	rRMSE	MSE
ANN SVR-ANN	<i>Epochs</i> 4000	2.28	399.99	2.87	247592.09
	9000	2.27	392.53	2.99	258336.42
SVR SVR-SVR	<i>Gamma</i> 0.00	2.05	358.97	2.56	199730.19
	0.50	2.04	357.32	2.60	205740.59
Random Forest SVR-Random Forest	<i>ntrees</i> 100	2.42	426.06	3.04	285116.07
	100	2.11	370.79	2.64	215054.35

Table 21
Prediction performance of 5-days ahead of time (For S&P BSE SENSEX).

Prediction models	Parameters	Error measures			
		MAPE	MAE	rRMSE	MSE
ANN SVR-ANN	<i>Epochs</i> 2000	2.70	476.36	3.42	363471.55
	8000	2.30	403.21	2.92	259749.88
SVR SVR-SVR	<i>Gamma</i> 2.50	2.31	405.27	2.90	255505.25
	0.50	2.23	392.30	2.83	244955.34
Random Forest SVR-Random Forest	<i>ntrees</i> 50	2.68	473.25	3.34	344949.70
	100	2.32	409.54	2.89	256949.33

Table 22
Prediction performance of 6-days ahead of time (For S&P BSE SENSEX).

Prediction models	Parameters	Error measures			
		MAPE	MAE	rRMSE	MSE
ANN SVR-ANN	<i>Epochs</i> 7000	2.70	478.29	3.37	357530.68
	6000	2.43	429.90	3.14	312341.99
SVR SVR-SVR	<i>Gamma</i> 4.00	2.53	443.13	3.18	308063.70
	0.50	2.42	425.68	3.04	282869.87
Random Forest SVR-Random Forest	<i>ntrees</i> 50	2.92	516.14	3.68	419168.80
	150	2.50	439.29	3.10	293822.81

Table 23
Prediction performance of 7-days ahead of time (For S&P BSE SENSEX).

Prediction models	Parameters	Error measures			
		MAPE	MAE	rRMSE	MSE
ANN SVR-ANN	<i>Epochs</i> 9000	2.82	497.40	3.63	412421.92
	5000	2.53	445.19	3.23	320178.09
SVR SVR-SVR	<i>Gamma</i> 4.00	2.69	472.27	3.42	356444.96
	1.50	2.55	447.80	3.26	324401.97
Random Forest SVR-Random Forest	<i>ntrees</i> 100	3.20	566.55	4.04	508254.64
	150	2.69	473.19	3.33	341307.44

It can be observed that performance of SVR-ANN and SVR-RF models is improved significantly than ANN and SVR models. SVR-SVR model exhibits a moderate improvement over SVR model. These results demonstrates the effectiveness of our proposal. Comparison of prediction performance of all the models for both the stock market indices reveals that SVR-ANN model performs the best overall.

Table 24

Prediction performance of 8-days ahead of time (For S&P BSE SENSEX).

Prediction models	Parameters	Error measures			
		MAPE	MAE	rRMSE	MSE
ANN SVR-ANN	<i>Epochs</i> 6000	3.12	546.67	3.93	465739.08
	7000	2.64	459.58	3.54	367846.18
SVR SVR-SVR	<i>Gamma</i> 0.00	2.84	497.85	3.61	396295.99
	1.50	2.71	475.37	3.48	368319.67
Random Forest SVR-Random Forest	<i>ntrees</i> 150	3.38	596.67	4.29	569631.81
	100	2.88	507.31	3.58	392375.92

Table 25

Prediction performance of 9-days ahead of time (For S&P BSE SENSEX).

Prediction models	Parameters	Error measures			
		MAPE	MAE	rRMSE	MSE
ANN SVR-ANN	<i>Epochs</i> 9000	3.17	554.86	4.05	496313.24
	6000	2.87	499.37	3.74	411681.57
SVR SVR-SVR	<i>Gamma</i> 0.00	3.02	529.53	3.84	444071.59
	0.50	2.87	502.33	3.68	408835.92
Random Forest SVR-Random Forest	<i>ntrees</i> 50	3.49	616.32	4.46	615743.91
	150	3.10	545.80	3.86	454853.43

Table 26

Prediction performance of 10-days ahead of time (For S&P BSE SENSEX).

Prediction models	Parameters	Error measures			
		MAPE	MAE	rRMSE	MSE
ANN SVR-ANN	<i>Epochs</i> 8000	3.45	603.90	4.40	585260.94
	5000	2.72	474.62	3.60	387086.13
SVR SVR-SVR	<i>Gamma</i> 10.00	3.19	557.98	4.10	505260.72
	0.50	3.00	525.45	3.87	449987.02
Random Forest SVR-Random Forest	<i>ntrees</i> 100	3.62	637.70	4.60	648907.41
	150	3.19	561.07	4.06	497755.22

Table 27

Prediction performance of 15-days ahead of time (For S&P BSE SENSEX).

Prediction models	Parameters	Error measures			
		MAPE	MAE	rRMSE	MSE
ANN SVR-ANN	<i>Epochs</i> 5000	3.90	681.64	4.84	700906.09
	6000	3.58	624.50	4.54	612524.19
SVR SVR-SVR	<i>Gamma</i> 10.00	3.94	688.47	4.96	735083.38
	0.50	3.69	644.66	4.68	651935.80
Random Forest SVR-Random Forest	<i>ntrees</i> 50	4.16	730.56	5.52	916603.04
	150	3.80	664.37	4.81	692641.93

Fig. 5 shows the actual value of the CNX Nifty, value predicted by ANN and SVR-ANN models for the task of predicting 5-day ahead of time. The visual representation also validates the effectiveness of our proposed two stage fusion approach. Visual

Table 28

Prediction performance of 30-days ahead of time (For S&P BSE SENSEX).

Prediction models	Parameters	Error measures			
		MAPE	MAE	rRMSE	MSE
ANN SVR-ANN	<i>Epochs</i> 8000	4.83	839.92	6.28	1152684.67
	1000	4.32	745.01	5.58	891384.72
SVR SVR-SVR	<i>Gamma</i> 0.00	5.26	913.05	6.75	1341763.55
	0.50	4.74	822.34	5.99	1047397.92
Random Forest SVR-Random Forest	<i>ntrees</i> 150	5.29	926.53	6.73	1357838.40
	50	4.67	809.42	5.94	1022031.89

Table 29

Average Prediction performance for CNX Nifty.

Prediction Model	MAPE	MAE	rRMSE	MSE
ANN	3.01	160.10	3.79	43872.97
SVR-ANN	2.66	139.99	3.41	34207.13
SVR	2.71	142.54	3.43	37530.53
SVR-SVR	2.66	140.17	3.39	34975.22
Random Forest (RF)	3.02	160.08	3.87	46992.46
SVR-RF	2.72	143.64	3.44	36166.73

Table 30

Performance Improvement (Single Stage Models vs. Two Stage Fusion Models) for CNX Nifty

Models under comparison	MAPE	MAE	rRMSE	MSE
ANN vs. SVR-ANN	11.57	12.56	10.22	22.03
SVR vs. SVR-SVR	1.66	1.66	1.12	6.81
RF vs. SVR-RF	9.81	10.27	11.2	23.04

Table 31

Average prediction performance for S&P BSE Sensex.

Prediction model	MAPE	MAE	rRMSE	MSE
ANN	2.90	509.19	3.69	446695.60
SVR-ANN	2.58	449.75	3.34	358157.63
SVR	2.66	465.55	3.38	402606.42
SVR-SVR	2.60	454.69	3.31	366287.80
Random Forest (RF)	3.01	529.80	3.83	508104.95
SVR-RF	2.67	467.92	3.35	378614.56

Table 32

Performance improvement (single stage models vs. two stage fusion models) for S&P BSE Sensex.

Models under comparison	MAPE	MAE	rRMSE	MSE
ANN vs. SVR-ANN	11.2	11.67	9.42	19.82
SVR vs. SVR-SVR	2.41	2.33	1.88	9.02
RF vs. SVR-RF	11.31	11.68	12.7	25.48

representation for the other prediction tasks (not shown here) also demonstrates the effectiveness of our proposed approach.

The reason behind the improved performance of two stage fusion approach over the single stage approach may be justified as follows. In two stage fusion approach, prediction models in the second stage, have to identify transformation from technical parameters describing $(t + n)^{\text{th}}$ day to $(t + n)^{\text{th}}$ day's closing price while in single stage approach, prediction models have to identify transformation from technical parameters describing t^{th} day to $(t + n)^{\text{th}}$ day's closing price.

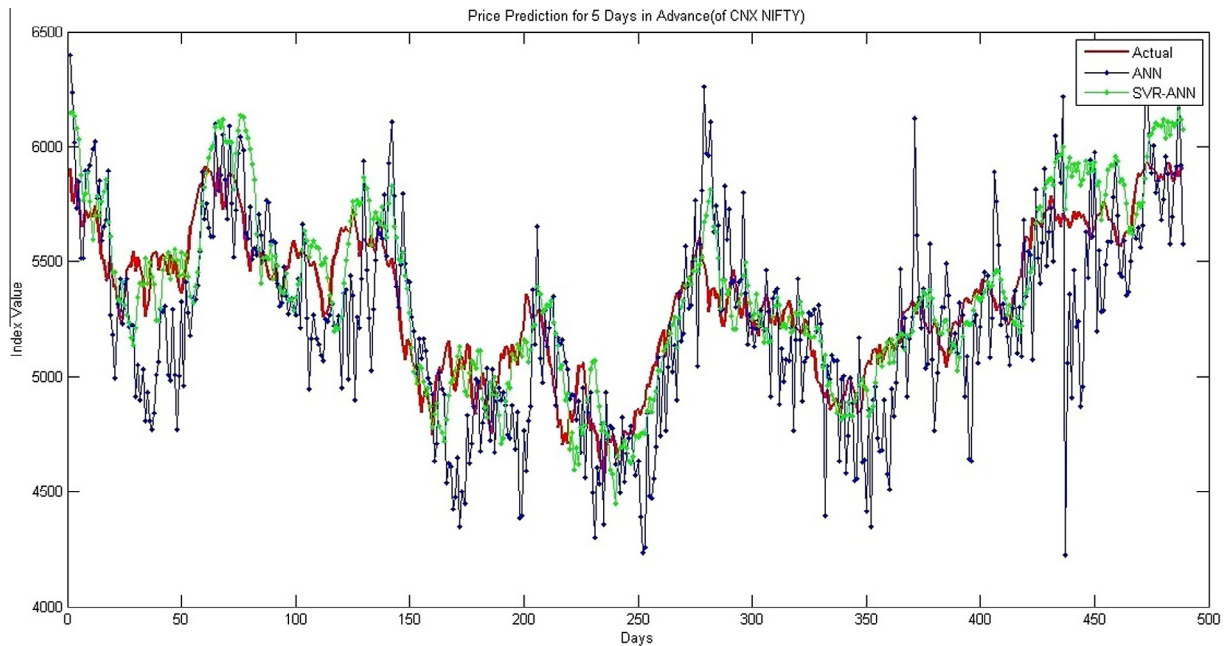


Fig. 5. Prediction performance comparison of ANN and SVR-ANN for predicting 5 day ahead of time for CNX Nifty.

The introduction of an additional stage in case of two stage fusion approach takes the responsibility of preparing data for the prediction models in the second stage. Actually it transforms closing and opening price, low and high of t^{th} day to technical parameters representing $(t+n)^{\text{th}}$ day. This may reduce the prediction error, as now, the prediction models in second stage have to predict based on predicted technical parameters of $(t+n)^{\text{th}}$ day rather than actual technical parameters but of t^{th} day.

5. Conclusions

The task of predicting future values of stock market indices is focused in this paper. Experiments are carried out on ten years of historical data of two indices namely CNX Nifty and S&P BSE Sensex from Indian stock markets. The predictions are made for 1–10, 15 and 30 days in advance.

Review of the existing literature on the topic revealed that existing methods for predicting stock market index's value/price have used a single stage prediction approach. In these existing methods, the technical/statistical parameters' value of $(t)^{\text{th}}$ day is used as inputs to predict the $(t+n)^{\text{th}}$ day's closing price/value (t is a current day). In such scenarios, as the value of n increases, predictions are based on increasingly older values of statistical parameters and thereby not accurate enough. It is clear from this discussion that there is a need to address this problem and two stage prediction scheme which can bridge this gap and minimize the error stage wise may be helpful.

Some of the literatures on the focused topic have tried to hybridize various machine learning techniques but none has tried to bridge the identified gap, rather, in these literatures, generally it is found that one machine learning technique is used to tune the design parameters of the other technique.

A two stage fusion approach involving Support Vector Regression (SVR) in the first stage and Artificial Neural Network (ANN), Random Forest (RF) and SVR in the second stage is proposed in this paper to address the problem identified. Experiments are carried out with single stage and two stage fusion prediction models. The results show that two stage hybrid models perform better than that of the single stage prediction models. The performance improvement is significant in case when ANN and RF are

hybridized with SVR. A moderate improvement in the performance is observed when SVR is hybridized with itself. The benefits of two stage prediction models over single stage prediction models become evident as the predictions are made for more number of days in advance. The best overall prediction performance is achieved by SVR-ANN model.

The proposal of two stage prediction scheme is a significant research contribution of this paper as this scheme provides a kind of new way of feeding adequate information to prediction models. To accomplish this, machine learning methods are used in cascade in two stages. First stage uses SVR to predict future values of statistical parameters which are fed as the inputs to the prediction models in the second stage. Experimental results are promising and demonstrates the usefulness of the proposed approach. The proposed approach is not only successful but also useful and adaptable for other prediction tasks such as weather forecasting, energy consumption forecasting and GDP forecasting. This generalizability of the proposed approach definitely makes the proposal a significant contribution to the research.

In stock market, investors use the term 'stop-loss order' which states order to close a position if/when losses reach a threshold. In this paper, predictions are made for 1–10 days, 15 and 30 days ahead of time. This can help investors by suggesting 'stop-loss order' for 1–10 days or 15 days or 30 days i.e. 'stop loss order for a month'. As the proposed model helps reducing the prediction error, investors can clear their position booking less amount of loss or more profit. As stated earlier proposed approach can also be used in other domains like weather forecasting, energy consumption forecasting or GDP forecasting, which is also an insightful implications of the proposed approach.

In this paper design parameters of SVRs in the first stage are determined experimentally, however it may be worth exploring algorithms such as genetic algorithm to tune the design parameters of these SVRs. This may lead to more accurate prediction of statistical parameters by these SVRs. Another direction for future work can be to use more statistical parameters as inputs to find much better correlation. It is well known that the prices of stocks depend on several aspects like government policies, company performance, investors' interest etc. A news related to any of these aspects definitely affects the stocks' prices. These news can be

categorized as 'good', 'very good', 'bad', or 'worse'. An interesting future direction is to incorporate these news related to these aspects by means of their category as discussed above. Such a semi-supervised system will allow to make systems more robust and predictions more accurate.

References

- Aldin, M. M., Dehnavr, H. D., & Entezari, S. (2012). Evaluating the employment of technical indicators in predicting stock price index variations using artificial neural networks (case study: Tehran stock exchange). *International Journal of Business & Management*, 7.
- Asadi, S., Hadavandi, E., Mehmanpazir, F., & Nakhostin, M. M. (2012). Hybridization of evolutionary Levenberg–Marquardt neural networks and data pre-processing for stock market prediction. *Knowledge-Based Systems*, 35, 245–258.
- Bollerslev, T. (1986). Generalized autoregressive conditional heteroskedasticity. *Journal of Econometrics*, 31, 307–327.
- Breiman, L., Friedman, J., Stone, C. J., & Olshen, R. A. (1984). *Classification and regression trees*. CRC press.
- Cheng, C., Xu, W., & Wang, J. (2012). A comparison of ensemble methods in financial market prediction. In *2012 Fifth international joint conference on computational sciences and optimization (CSO)* (pp. 755–759). IEEE.
- Cristianini, N., & Shawe-Taylor, J. (2000). An introduction to support vector machines.
- Hadavandi, E., Ghanbari, A., & Abbasian-Naghneh, S. (2010a). Developing an evolutionary neural network model for stock index forecasting. In *Advanced intelligent computing theories and applications* (pp. 407–415). Springer.
- Hadavandi, E., Shavandi, H., & Ghanbari, A. (2010b). Integration of genetic fuzzy systems and artificial neural networks for stock price forecasting. *Knowledge-Based Systems*, 23, 800–808.
- Han, J., Kamber, M., & Pei, J. (2006). *Data mining: Concepts and techniques*. Morgan Kaufmann.
- Hsieh, D. A. (1991). Chaos and nonlinear dynamics: Application to financial markets. *The Journal of Finance*, 46, 1839–1877.
- Huang, C.-L., & Tsai, C.-Y. (2009). A hybrid SOFM–SVR with a filter-based feature selection for stock market forecasting. *Expert Systems with Applications*, 36, 1529–1539.
- Huang, S.-C., & Wu, T.-K. (2008). Integrating ga-based time-scale feature extractions with SVMS for stock index forecasting. *Expert Systems with Applications*, 35, 2080–2088.
- Kara, Y., Acar Boyacioglu, M., & Baykan, Ö. 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*, 38, 5311–5319.
- Kazem, A., Sharifi, E., Hussain, F. K., Saberi, M., & Hussain, O. K. (2013). Support vector regression with chaos-based firefly algorithm for stock market price forecasting. *Applied Soft Computing*, 13, 947–958.
- Kecman, V. (2001). *Learning and soft computing*.
- Lee, Y.-S., & Tong, L.-I. (2011). Forecasting time series using a methodology based on autoregressive integrated moving average and genetic programming. *Knowledge-Based Systems*, 24, 66–72.
- Miao, K., Chen, F., & Zhao, Z.-g. (2007). Stock price forecast based on bacterial colony RBF neural network [j]. *Journal of Qingdao University (Natural Science Edition)*, 2, 011.
- Ou, P., & Wang, H. (2009). Prediction of stock market index movement by ten data mining techniques. *Modern Applied Science*, 3, P28.
- Pai, P.-F., Lin, K.-P., Lin, C.-S., & Chang, P.-T. (2010). Time series forecasting by a seasonal support vector regression model. *Expert Systems with Applications*, 37, 4261–4265.
- Parrella, F. (2007). *Online support vector regression* (Master's thesis). Italy: Department of Information Science, University of Genoa.
- Quinlan, J. R. (1986). Induction of decision trees. *Machine Learning*, 1, 81–106.
- Quinlan, J. R. (1993). *C4. 5: programs for machine learning* (Vol. 1). Morgan kaufmann.
- Rao, T., & Gabr, M. (1984). *Introduction to bispectral analysis and bilinear time series models. Lecture notes in statistics* (Vol. 24).
- Shen, W., Guo, X., Wu, C., & Wu, D. (2011). Forecasting stock indices using radial basis function neural networks optimized by artificial fish swarm algorithm. *Knowledge-Based Systems*, 24, 378–385.
- Vapnik, V. N. (1999). An overview of statistical learning theory. *IEEE Transactions on Neural Networks*, 10, 988–999.
- Zarandi, M., Hadavandi, E., & Turksen, I. (2012). A hybrid fuzzy intelligent agent-based system for stock price prediction. *International Journal of Intelligent Systems*, 27, 947–969.
- Zhang, Y., & Wu, L. (2009). Stock market prediction of s&p 500 via combination of improved BCO approach and BP neural network. *Expert Systems With Applications*, 36, 8849–8854.