Asian Journal of Information Technology 15 (2): 329-337, 2016
ISSN: 1682-3915

Improving Stock Index Predictability of Machine Learning Algorithms with Global Cues

[1]M.V. Subha, [1]Arul Sulochana and [2]Thirupparkadal Nambi
[1]Department of Management Studies, Anna University Regional Centre,
Maruthamalai Main Road, Coimbatore, India
[2]DJ Academy for Managerial Excellence Coimbatore, Pollachi Highway, Othakkalmandapam,
641032 Coimbatore, Coimbatore, India

Abstract: The performance of predictive models depends more on predictor sets than the efficiency of the predictive algorithm. This study investigates the role of global cues as input predictors in improving the predictability of the models. It also attempts to study the efficacy of various machine learning algorithms such as support vector machines, case-based reasoning, decision trees and artificial neural network in forecasting the stock indices. It focuses on studying predictability of the widely-followed Indian stock market indices BSE SENSEX and CNX Nifty using the above-mentioned four machine learning algorithms with two separate set of predictors, a set of commonly used technical indicators and another set of daily global cues such as gold price, crude oil price, exchange rate of strong currencies, LIBOR and close price of major global stock market indices. With its lowest forecasting error values, SVM outperforms other predictive models in terms of all key performance metrics. Among the predictor sets, global cues show a higher level of predictive accuracy.

Key words: Stock index prediction, support vector machines, case based reasoning, decision trees, machine learning, neural network

INTRODUCTION

Predicting stock market movement is an essential activity of paramount importance for all stock market analysts. The ability to forecast the price movement and the trend guides investors in their decisions. It is of immense value to individual investors, institutional investors, financial institutions and others whose fortunes are largely dependent on the movement of the stock market. It is also useful for policy makers to formulate suitable policies for overseeing efficient functioning of the system. However, high volatility in stock prices makes it difficult to predict the stock market movements. Hence, stock market analysts focus on developing models to successfully forecast index values/stock prices, aiming at high profits using well-defined trading strategies. The key to successful forecasting of stock market trends is achieving best results with minimum required input data (Atsalakis and Valavanis, 2009). Apart from the algorithm employed, the efficacy of the predictive model depends on the input parameters chosen for the study.

The field of financial forecasting is characterized by data intensity, noise, non-stationary, unstructured nature, high degree of uncertainty and hidden relationships (Hall, 1994). The efficient market hypothesis claims that the markets are efficient and hence share prices move in a random manner. However, academic research studies have proved that stock market price movements are not always random. Rather, they behave in a highly non-linear dynamic manner (Blank, 1991). Modeling of stock index prices should consider the current trend of the market (bullish/bearish, boom/recession, etc.) apart from other factors. There are myriad number of factors such as domestic and global economic conditions, political situation, investors' psyche, catastrophe and other unexpected events that influence the movement of the stock markets. The markets exhibit both linear and non-linear behavior and hence, models that can capture non-linear behavior are chosen for obtaining accurate predictability. Of late, various machine learning approaches are extensively employed to study the stock market behavior. Some of the popular machines learning approaches are decision trees, Artificial Neural Networks

Corresponding Author: M.V. Subha, Department of Management Studies, Anna University Regional Centre, Maruthamalai Main Road, Coimbatore, India

(ANN), association rule mining, Case-Based Reasoning (CBR), Support Vector Machines (SVM), clustering etc. ANN has been extensively used to forecast stock index movements and number of successful applications has shown that ANN can be a very useful tool for time-series modeling and forecasting (Zhang *et al*., 1998; Aiken and Bsat, 1999). In particular, Back Propagation Neural Network (BPNN) is frequently used in the stock market prediction since its accuracy of prediction is known to be better than others (Min and Lee, 2005). A neural network is more effective in describing the dynamics of non-stationary time series due to its unique non-parametric, non-assumable, noise-tolerant and adaptive properties. However, a critical issue concerning neural network is the over fitting problem. Neural network captures not only useful information contained in the given data but also unwanted noise (Tay and Cao, 2001a). Recent studies tend to hybridize certain ANN techniques. Support vector machine is a novel neural network algorithm, developed by Vapnik 1999. SVM implements the structural risk minimization principle and the solution of SVM may be global optimum and hence solves the problem of overfitting (Kim, 2003). Tay and Cao (2001b) examined the feasibility of applying SVM in financial forecasting and investigated the functional characteristics in forecasting.

Lahmiri (2011) applied the PNN and SVM techniques to predict the stock market trend using technical and macro economic indicators and combining them together in order to compare the performance of the classifiers. Yang *et al*. (2002) applied the SVM regression model to financial forecasting by incorporating market adaptations. Wang *et al*. (2011), proposed a hybrid approach combining ESM, ARIMA and BPNN to forecast the stock index movement. Dai *et al*. (2012) proposed a time series prediction model by combining Nonlinear Independent Komponent Analysis (NLICA) and neural network to forecast Asian stock markets. Ticknor (2013) has tried a Bayesian regularized artificial neural network as a novel method to forecast financial market behavior using individual stocks.

The most commonly used inputs are the opening and closing price of the index as well as the daily highest and lowest values supporting the statement that soft computing methods use quite simple input data to provide predictions (Atsalakis and Valavanis, 2009). Some studies have also used exchange rate of strong currencies along with the daily prices of established markets like the Dow Jones, S&P (Abraham *et al.* 2001; Huang *et al.* 2005; Siekmann *et al.* 2001; Wikowska, 1995) and technical indicators as predictors of stock index movements (Armano *et al*., 2005). Review of recent studies show that majority of the research works aim at developing various hybrid machine learning tools to improve the predictability. However, not many studies are carried out to understand the role of input parameters in improving the accuracy of predictive models. Hence, this study employs popular machine learning algorithms to forecast the stock index price by using various technical and global cues as separate input factors and tries to check the efficacy of predictor sets. It also examines the efficacy of SVM by benchmarking its performance with CBR, ANN and decision tree models.

Basic concepts of SVM, ANN, CBR and decision tree

Machine learning: Machine learning deals with the development and study of systems that canlearnfrom past data. Once the system learns, it can be put to actual use for prediction or classification of new data instances. Machine learning process consists of representation and generalization. Representation includes depiction of past data instances and the learning that happens to the system that are expressed in terms of functions and parameters. These functions and parameters are repeatedly evaluated and attain their critical values as a result of iterative training process using past data set. Generalization is the ability of trained system to perform accurately the task of prediction or classification on new data sets. Thus, the core objective of machine learning is to generalize from its experience. Many machine learning approaches such as decision trees, artificial neural networks, association rule mining, case-based reasoning, support vector machines and clustering are extensively employed to study the stock market behavior. This study attempts to compare the performance of support vector machines with other machine learning tools like case-based reasoning, ANN and decision tree.

Support vector machines: Support vector machine is a promising algorithm, developed by Vapnik (1995) which can be used to forecast both linear and non linear data. It uses non linear mapping to transform training data into higher dimension plane. Within the new dimension, it searches for a linear optimal separating plane demarcating one class from the other. The transformation of data from lower dimension to higher dimensional plane is done by kernel functions. Different types of kernel functions including linear, polynomial and Radial Bias Functions (RBF), Pearson function-based Universal Kernel (PUK) can be adopted for SVM. SVM tries to create an optimal line of separation or hyperplane between the objects of two different classes. It also tries to optimize the width of the hyperplane with two boundaries which is known as maximal margin hyperplane. The instances which are touching the margins, are support vectors. A separating hyper plane can be written as:

$$W.X+b=0 \tag{1}$$

Where:

W = A weight vector W = $\{w_1, w_2, w_3, \ldots w_n)$
n = The number of attributes
b = A scalar called bias

$$w_0+w_1x_1+w_2x_2+\ldots=0 \tag{2}$$

The weights can be adjusted so that the hyper planes defining the sides of the margin can be written as:

$$H_1 w_0+w_1x_1+w_2x_2 \geq 1 \text{ for } y_i=+1 \text{ and} \tag{3}$$

$$H_2 w_0+w_1x_1+w_2x_2 \leq 1 \text{ for } y_i=-1 \tag{4}$$

Lagrangian formulation and quadratic programming can be used to find support vectors and maximum margin hyperplane and that can be used as the decision boundary:

$$d(X^T)=\sum_{i=1}^{l} y_i \alpha_i X_i X^T+b_0 \tag{5}$$

Where:

y_i = The class label of the support vector X_i
X^T = A test data
a_i = The Lagrangian multiplier
b_0 = The bias
l = The number of support vectors

Thus, the sign of X^T help in predicting the class of the test data. SVM for regression employs Sequential Minimal Optimization (SMO) technique to optimize the width of hyperplane and the model learns input-output relationship between input training datasets X_i and their corresponding outputs that are continuous in nature. Thus, the identified relationship is used to construct regression equation which is used for forecasting continuous values.

Case-based reasoning: As the name implies, this algorithm looks for 'k' nearest cases in sample dataset and uses outputs of those nearest cases for forecasting the output for a new instance of data. All neighbor's output can be given equal weights or the closest neighbors can be given more weights which are inversely proportional to its distance from the new data record. One of the most widely used metric for identifying nearest neighbor is Euclidean distance. The Euclidean distance between two instances $(x_1, x_2, x_3, \ldots, x_p)$ and $(u_1, u_2, u_3, \ldots, u_p)$ is given by the following Equation:

$$\sqrt{(x_1-u_1)^2+(x_2-u_2)^2+\ldots+(x_p-u_p)^2}$$

where, $x_1, x_2, x_3, \ldots, x_p$ are predictors of the instance 1 and $u_1, u_2, u_3, \ldots u_p$ are predictors of the instance 2.

Artificial neural network: ANN is an efficient information processing system that mimics the way a human brain functions. It consists of layered set of highly interconnected processing elements called nodes or neurons. Each node emulates a neuron in the brain and is connected with the other by a connection link. An artificial neural network contains several neurons with an input layer and an output layer. The number of nodes in the input layer is determined by the number of input attributes. The number of nodes in the output is determined by the number of classes in the output. Depending on the complexity, there may be one or more intermediary layers. These intermediary layers are called hidden layers. Feed forward net works are the most popular networks widely used. In these networks, the inputs are applied at the nodes in the input layer and they propagate through the network in forward direction to the output layer. As they propagate, they are modified by the weights associated with links. The weights can be a positive, negative, or even a fraction. Each node in a neural net, combine the incoming values and transmits the same to the next layer either altered or unaltered. The alteration is done by an activation function only if the sum of the inputs is greater than a threshold value. The most common activation function is the sigmoid function which is given by Eq. 6:

$$f(y)=1/\left(1+e^{-ky}\right) \tag{6}$$

A back propagation feed forward neural network can propagate the errors back into the network, so that weights associated with links are suitably adjusted till the neural network is adequately trained. The underlying principle is simple. Inputs from the training datasets are propagated through the network through the input layer, getting affected by the arbitrary initial weights assigned to links, result in output. The output of the model is compared with the actual outputs in the training data. The difference is the error. This error is then fed back into the network and the connection weights are suitably adjusted, so that errors are minimized. This is the learning process. A neural net is trained by repeated passes of the all data in the training dataset. Each pass is called an epoch. With more epochs during training phase, the neural network learns more about the data but too much of training may also result in over fitting.

Decision trees: It is based on constructing hierarchical trees from the historical dataset available. The fully grown trees and pruned trees are then used for forecasting new instances. Decision trees are represented by a set of questions which splits the entire dataset into smaller subgroups. It searches for one predictor variable and its particular value that splits the entire dataset into two parts with maximum homogeneity in terms of decision variable. The choice of split variable is based on impurity function. Two of the most popular impurity function measures are entropy measure and Gini Index:

$$\text{Entropy} = -\sum_{i=1}^{m} p_i \log_2(p_i) \tag{7}$$

$$\text{Gini index} = 1 - \sum_{i=1}^{m} p_i^2 \tag{8}$$

where, p_i is the probability that an arbitrary new instance belongs to a class Ci. The predictor with least impurity measure is the choice of splitting the dataset. This splitting process is then continued with each of the resulting data fragments until "leaves" or decision nodes are reached. The resulting tree can be used for making decision rules for forecasting.

MATERIALS AND METHODS

Experimental setup

Data set: The daily closing price of BSE-SENSEX and S&P CNX Nifty for the period from 1st Jan 2008 to 31st May 2015, comprising of >1800 trading days, are considered for this study. This data set is split in the ratio of 80:20 to be used as training dataset and test dataset

Table 1: Predictor sets

Predictor set#1 (Global cues)	Predictors set #2 (Technical indicators)
Open price	Open price
High price	High price
Low price	Low price
Close price	Close price
NASDAQ	K%
NYSE	D%
Nikkei	MI
FTSE	ROC10
US$	William %R
Pound	MACD
Euro	P-MA20
Yen	A/DOscillator
Gold	OSCP
Brentoil	RSI
LIBOR	Output

The next day's closing price of the index

respectively. The training data set covers the time period from Jan 2008 to Dec 2013 with>1500 trading days and the remaining data of the period from Dec 2013 to May 2015 are used as test data set in order to validate the outcome of the predictive models.

Two groups of variables are chosen as the predictors to the model. The first group comprises of global cues that include daily close price of major global indices such as NYSE, NASDAQ, FTSE and NIKKEI, the daily exchange rate of strong currencies such as US Dollar, Pound, Euro and Japanese Yen and other global cues such as daily price of brent oil, gold price and London.

Interbank Offer Rate (LIBOR), apart from the intraday movements of the index under study. The second group comprises of intra-day movements of the index such as opening price, high price, low price, closing price and widely used technical indicators such as K and D %, Momentum Indicator (MI), ROC, William R%, MACD, P-MA, A/D Oscillator, OSCP and RSI (Table 1).

Description of each technical indicators used for this study and its respective calculations are shown in Table 2. WEKA's implementations of Machine Learning algorithms are employed in this study. All the inputs are normalized and the predictive model is developed and fine-tuned with the training data set. The developed model is then used with the test data sets of both BSE SENSEX and S&P C NX Nifty for thetask of fore casting. To compare the predictive performance of SVM, the same data sets are used with CBR, ANN and decision tree algorithms (Fig. 1).

Performance metrics: The performances of predictive models are evaluated using the following statistical metrics: correlation coefficient, Mean Absolute Error

Table 2: List of technical indicators

Description and technical indicator	Calculation
Stochastic K (%)	$\frac{C_t - LL_{t-n}}{HH_{t-n} - LL_{t-n}} \times 100$ where LL_t and HH_t are the lowest low and highest high of the index price in the last n days
Stochastic %D.This is the moving average of K (%)	$\frac{\sum_{i=0}^{n-1} \%k_{t-1}}{n}$
Momentum Indicator (MI)	C_t-C_{t-n}
Rate of Change of price (ROC)	$C_t/C_{t-n}\times 100$
It is a variant of momentum indicator (Williams' R, %)	H_n-C_t/H_n-$L_n\times 100$
Moving Average Convergence and Divergence (MACD)	MA_{10}-MA_{20}
Price Minus moving average (P-MA20)	C_i-MA_{20}
Accumulation/Distribution oscillator (A/D oscillator)	H_t-C_{t-1}/H_t-L_t
Price Oscillator (OSCP)	MA_{10}-MA_{20}/MA_{10}
Relative Strength Index (RSI)	$\frac{100}{1+\left(\sum_{i=1}^{n}\frac{UP_{t-i}}{n}\right)\left(\sum_{i=1}^{n}\frac{DN_{t-i}}{n}\right)} - 100$

C_t: the close price at time t; L_t: the low price at time t; Ht: the high price at time t and MA_t: the moving average for t days (n = 10)

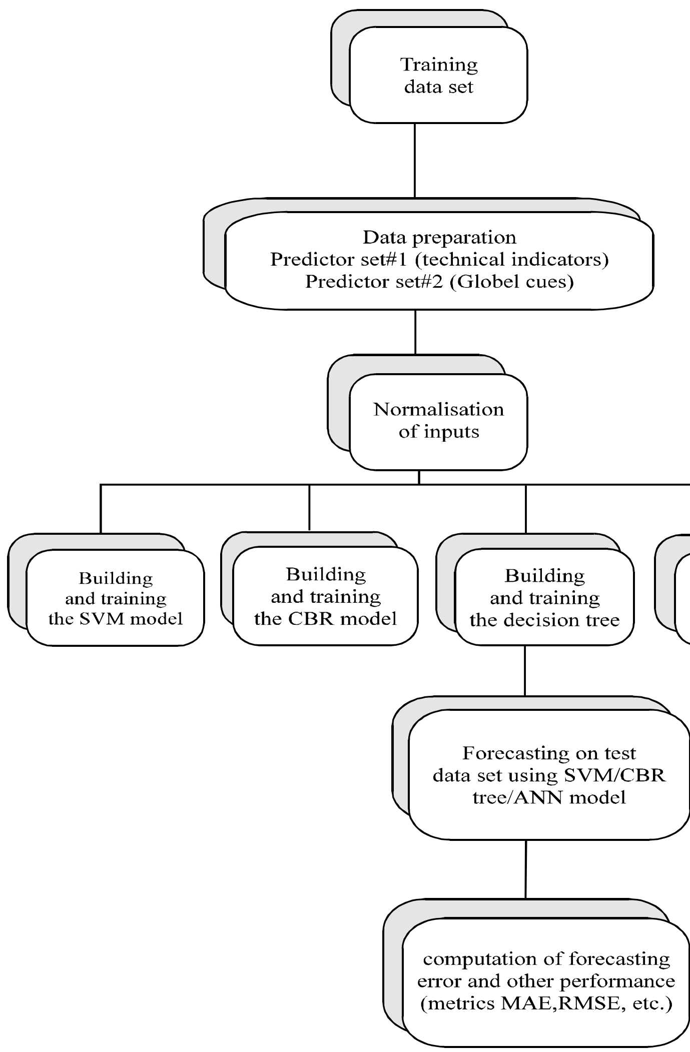

Fig. 1: Schematic diagram of the experimental set up

Table 3: Performance metrics

Performance metrics	Calculation
Correlation coefficient	$\frac{\sum(y_i-\bar{y}_i)^2\sum(y_i^{'}-\bar{y}_i)^2}{\sqrt{\sum\left[(y_i-\bar{y}_i(y_i^{'}-\bar{y}_i))\right]}}$
Mean Absolute Error (MAE)	$\frac{\sum_{i=0}^{n}\lvert y_i-y_i}{n}$
Root Mean Squared Error (RMSE)	$\sqrt{\frac{\sum_{i-0}^{n}(y_i-y_i')^2}{n}}$
Relative Absolute Error (RAE)	$\frac{\sum_{i=0}^{n}\lvert y_i-y'\rvert_i}{\sum_{i=0}^{n}\lvert y_i-\bar{y}_i\rvert}$
Root Relative Squared Error (RRSE)	$\sqrt{\frac{\sum_{i=0}^{n}(y_i-y_i')^2}{\sum_{i=0}^{n}(y_i-\bar{y}_i)^2}}$

(MAE), Root Mean Squared Error (RMSE), Relative Absolute Error (RAE) and Root Relative Squared Error (RRSE). The definitions for these metrics are given in Table 3.

RESULTS AND DISCUSSION

The predictive models are tested independently with two sets of predictors; one with the most commonly used technical indicators and the other with Global cues to forecast the next day's closing price of the index.

Experimental results of the predictive models for S&P CNX Nifty: The results of the four predictive models for S&P CNX Nifty are shown in Table 4. The lowest RMSE of 72.21 is observed in the case of SVM with global cues, SVM with technical indicator inputs gives an RMSE of 78.48. The lowest value of MAE 51.37 is observed again in the case of SVM with global cues, SVM with technical indicators gives an error of 56.39.

The other two error terms of RAE and RRSE reflect the same trend. In effect, SVM with global cues inputs edges over the SVM with technical indicators in forecasting the next day's index value as illustrated in Fig. 2 and 3. CBR Model also seems to function well with Global cues. The lowest RMSE of 77.99 and lowest MAE of 56.81 are obtained with global cues as input factors. The RMSE and MAE of technical indicators inputs are found to be 91.75 and 75.09, respectively. RAE and RRSE also verify the same trend. CBM with global cues inputs edges over the CBM with technical indicators in forecasting the next day's index value as illustrated in Fig. 4 and 5.

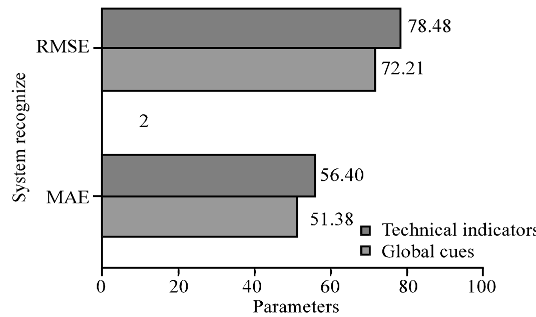

Fig. 2: Prediction error-system (Nifty)

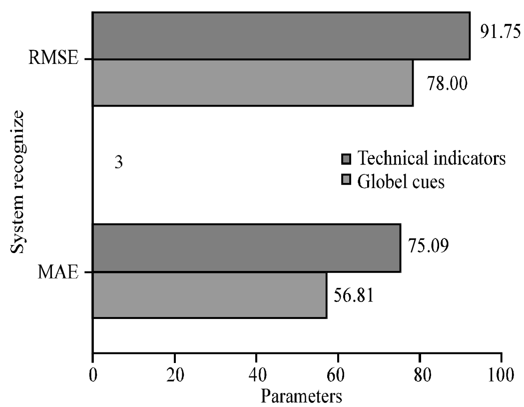

Fig. 3: Prediction error-CBR (Nifty)

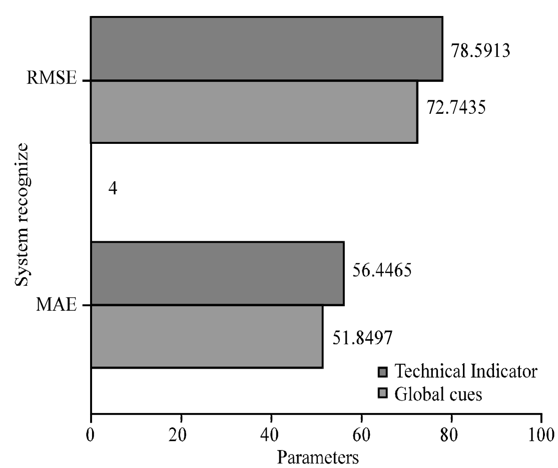

Fig. 4: Prediction error-tree (Tree)

The decision tree model is also found to perform better with global cues. The lowest RMSE of 72.74 and

Table 4: Results of three predictive models for S&P CNX Nifty

		S&P CNX Nift						
		Global cues			Technical Indicators			
Performance metrics	SVM	CBR	Tree	ANN	SVM	CBR	Tree	ANN
Correlation coefficient	0.9972	0.9967	0.9972	0.9972	0.9965	0.9601	0.9965	0.9965
MAE	51.3754	56.8130	51.8497	51.4626	56.3981	75.0890	56.4465	56.4687
RMSE	72.2108	77.9966	72.7435	72.2615	78.4804	91.7450	78.5913	78.5332
RAE(%)	6.2835	6.9485	6.3415	6.2941	7.2197	11.0382	7.2259	7.2287
RRSE(%)	7.4505	8.0474	7.5054	7.4557	8.4223	12.0942	8.4342	8.4280

Table 5: Results of the three predictive models for BSE-SENSEX

		BSE-SENSEX						
		Global cues			Technical Indicators			
Performance metrics	SVM	CBR	Tree	ANN	SVM	CBR	Tree	ANN
Correlation coefficient	0.9976	0.9973	0.9967	0.9966	0.9970	0.9928	0.9965	0.9970
MAE	163.1886	203.4095	191.3134	165.286	175.8184	279.4426	229.8417	177.7092
RMSE	222.3275	261.9101	260.7531	232.3275	238.0258	369.7388	304.5465	238.1244
RAE(%)	5.9261	7.3867	6.9474	6.2610	6.5987	10.9647	9.0185	6.8160
RRSE(%)	6.899	8.1273	8.0914	7.2990	7.7351	12.0154	9.8968	7.7383

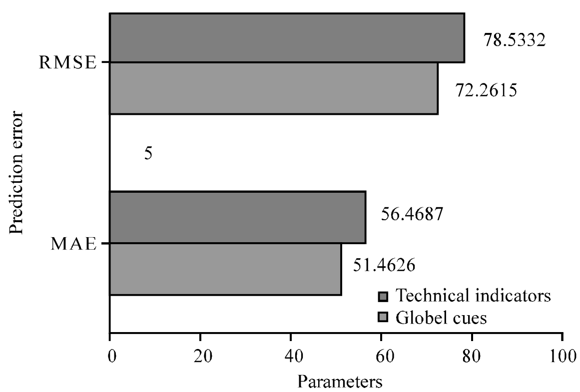

Fig. 5: Prediction error-Ann (Nifty)

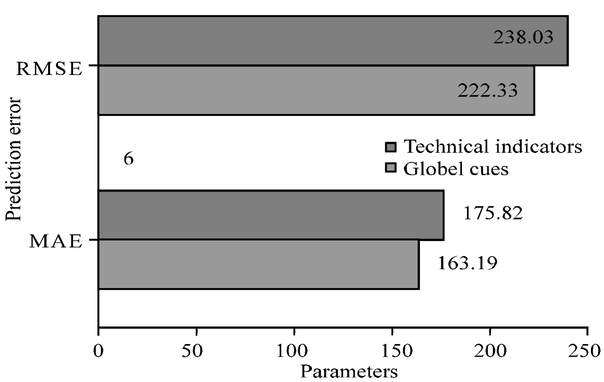

Fig. 6: Prediction error-SVM (sensex)

MAE of 51.85 are obtained with global cues whereas technical indicators results with RMSE of 78.59 and MAE of 56.45. RAE and RRSE also confirm the same trend.

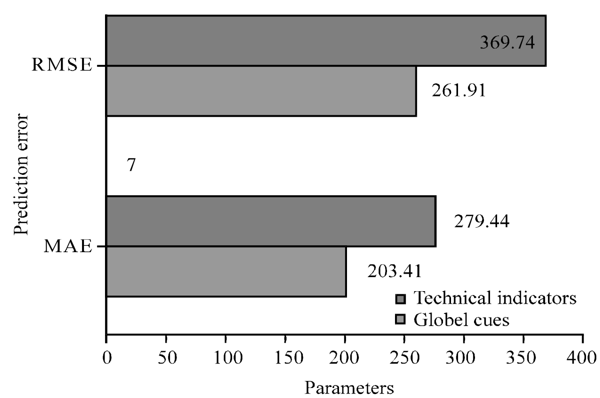

Fig. 7: Prediction error-CBR (sensex) the same trend

ANN Model also performs better with global cues. The lowest RMSE of 72.26 and lowest MAE of 51.46 are obtained with global cues as input factors. The RMSE and MAE of technical indicators inputs are found to be 78.53 and 56.47, respectively. RAE and RRSE also confirm the same trend.

Experimental results of the predictive models for bse SENSEX: The results of the four predictive models for BSE SENSEX are shown in Table 5. Fig. 6 and 7 The lowest RMSE of 222.33 is observed in the case of SVM with global cues, where as SVM with technical indicators gives RMSE of 238.03. For MAE, the lowest value of 163.19 is observed in the case of SVM with global cues, which is followed by technical indicator's 175. 82. RAE and RRSE also reflects the same trend.

It is found that the CBR model performs better with global cues. The lowest RMSE of 261.91 and lowest MAE of 203.41 are obtained with global cues. The RMSE and

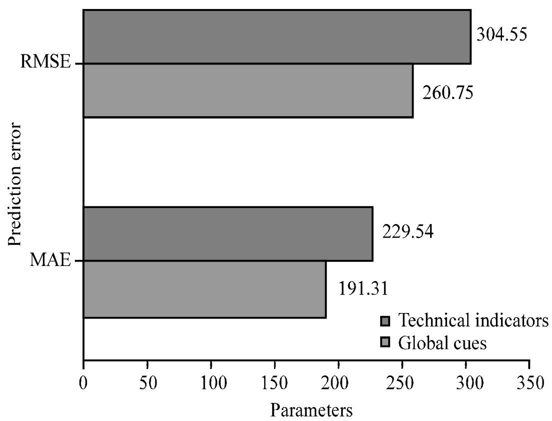

Fig. 8: Prediction error-tree (sensex)

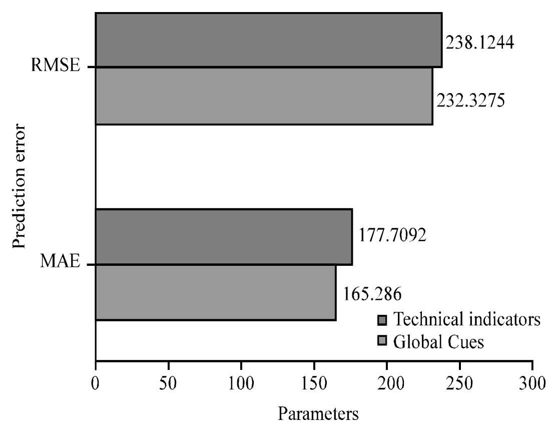

Fig. 9: Prediction error-ANN (ensex)

MAE of technical indicators inputs are found to be 369.74 and 279.44, respectively. The other two error terms, RAE and RRSE display the same trend (Fig. 8 and 9).

The decision tree model researches better with global cues. The RMSE of 260.75 and MAE of 191.31 are obtained with global cues whereas technical indicators result with the RMSE of 304.54 and MAE of 229.84. RAE and RRSE display the same trend. ANN Model also found to perform better with global cues. The lowest RMSE of 232.33 and lowest MAE of 165.29 are obtained with global cues as input factors. The RMSE and MAE of technical indicators inputs are found to be 238.12 and 177.70, respectively. RAE and RRSE display the same trend.

Comparative results of SVM, decision tree and CBR using various input parameters: Machine learning algorithms with global cue inputs are found to be performing better than technical indicator inputs. This is found to be corroborated with all the four predictive models in terms of all performance metrics.

It is also observed that SVM outperforms other three predictive models in terms of all the performance metrics-both in the cases of BSE SENSEX and S&P CNX Nifty. In terms of prediction accuracy, ANN comes second which is closely followed by decision tree. CBR lags behind other models in terms of accuracy.

CONCLUSION

This study is an attempt to evaluate the role of global cues as predictor set. This study also tries to evaluate the efficacy of various machine learning tools like SVM, CBR, ANN and Decision trees in predicting the movement of Indian stock market indices CNX Nifty and BSE Sensex. Predictive performances of various models are compared using different input parameters like technical indicators and global cues. The results of the study confirm that machine learning models are found to perform better with the input parameters of global cues. This also confirms that the SVM outperforms other models in terms of all performance metrics. Hence, there is an empirical evidence for global cues augmenting prediction accuracy.

REFERENCES

Abraham, A., B. Nath and P.K. Mahanti, 2001. Hybrid intelligent systems for stock market analysis. Proc. Int. Conf. Comput. Sci., 2074: 337-345.

Aiken, M. and M. Bsat, 1999. Forecasting market trends with neural networks. Inform. Syst. Manage., 16: 42-48.

Armano, G., M. Marchesi and A. Murru, 2005. A hybrid genetic-neural architecture for stock indexes forecasting. Inform. Sci., 170: 3-33.

Atsalakis, G.S. and K.P. Valavanis, 2009. Surveying stock market forecasting techniques-part II: Soft computing methods. Expert Syst. Appl., 36: 5932-5941.

Blank, S.C., 1991. Chaos in futures markets? A nonlinear dynamical analysis. J. Futures Markets, 11: 711-728.

Dai, W., J.V. Wu and C.J. Lu, 2012. Combining nonlinear independent component analysis and neural network for the prediction of Asian stock market indexes. Exp. Syst. Applic., 39: 4444-4452.

Hall, J.W., 1994. Adaptive Selection of US Stocks with Neural Nets. In: Trading on the Edge: Neural, Genetic and Fuzzy Systems for Chaotic Financial Markets, Deboeck, G.J. (Ed.). Wiley, New York, pp: 45-65.

Huang, W., Y. Nakamori and S.Y. Wang, 2005. Forecasting stock market movement direction with support vector machine. Comput. Oper. Res., 32: 2513-2522.

Kim, K.J., 2003. Financial time series forecasting using support vector machines. Neurocomputing, 55: 307-319.

Lahmiri, S., 2011. A comparison of PNN and SVM for stock market trend prediction using economic and technical information. Int. J. Comput. Appl., 29: 24-30.

Min, J.H. and Y.C. Lee, 2005. Bankruptcy prediction using support vector machine with optimal choice of kernel function parameters. Expert Syst. Appl., 28: 603-614.

Siekmann, S., R. Kruse, J. Gebhardt, F.V. Overbeek and R. Cooke, 2001. Information fusion in the context of stock index prediction. Int. J. Intell. Syst., 16: 1285-1298.

Tay, F.E.H. and L. Cao, 2001a. Application of support vector machines in financial time series forecasting. Omega, 29: 309-317.

Tay, F.E.H. and L.J. Cao, 2001b. Improved financial time series forecasting by combining support vector machines with self-organizing feature map. Intell. Data Anal., 5: 339-354.

Ticknor, J.L., 2013. A Bayesian regularized artificial neural network for stock market forecasting. Expert Syst. Appl., 40: 5501-5506.

Vapnik, V.N., 1995. The Nature of Statistical Learning Theory. 1st Edn., Springer-Verlag, New York, USA.

Vapnik, V.N., 1999. An overview of statistical learning theory. IEEE Trans. Neural Networks, 10: 988-999.

Wang, J.Z., J.J. Wang, Z.G. Zhang and S.P. Guo, 2011. Forecasting stock indices with back propagation neural network. Exp. Syst. Applic., 38: 14346-14355.

Wikowska, D., 1995. Neural networks as a forecasting instrument for the polish stock exchange. Int. Adv. Econ. Res., 1: 232-242.

Yang, H., L. Chan and I. King, 2002. Support Vector Machine Regression for Volatile Stock Market Prediction. In: Intelligent Data Engineering and Automated Learning-IDEAL 2002. Yin, H., N. Allinson, R. Freeman, J. Keane and S. Hubbard (Eds.). Springer Berlin Heidelberg, Berlin, Germany, pp: 391-396.

Zhang, G., B.E. Patuwo and M.Y. Hu, 1998. Forecasting with artificial neural networks: The state of the art. Int. J. Forecast., 14: 35-62.