

Support Vector Machines for Prediction of Futures Prices in Indian Stock Market

Shom Prasad Das

Department of Computer Science & Engineering
National Institute of Science & Technology
Palur Hills, Odisha- 761008, India

Sudarsan Padhy

Department of Computer Science & Engineering
International Institute of Information Technology
Bhubaneswar, Odisha- 751003, India

ABSTRACT

Machine learning methods are being used by several researchers for successfully predicting prices of financial instruments from the financial time series data of different markets. As the nature of markets in different regions are different, in this paper two machine learning techniques: Back Propagation Technique (BP) and Support Vector Machine Technique (SVM) have been used to predict futures prices traded in Indian stock market. The performances of these techniques are compared and it is observed that SVM provides better performance results as compared to BP technique. The implementation is carried out using MATLAB and SVM Tools (LS-SVM Tool Box).

General Terms

Machine Learning, Forecasting

Keywords

Back Propagation Neural Network (BPN), Support Vector Machines (SVM), Futures Contract, Financial Time Series.

1. INTRODUCTION

Accurate prediction of prices of financial instruments is essential to take better investment decisions with minimum risk. In view of the complexity of the financial time series data, resulting from a huge number of factors which could be economic or political [1-2]. Machine learning and soft computing methods have been used by several authors in the last two decades for financial time series forecasting.

The nonlinearity of financial time series motivated many authors to use back-propagation (BP) neural network due to its simple architecture design yet powerful problem solving ability. However this method has following drawbacks: presence large number of controlling parameters, over-fitting problem, slow convergence and stuck to local minima (as the back propagation algorithm is obtained by minimizing a nonlinear error function) [3].

In order to address these difficulties Vapnik and his coworkers have introduced support vector machine based on structural risk minimization principle [4]. Support vector machine successfully overcame the defects as a result predictive accuracy is improved in many applications. This motivated us to use SVM method for prediction of futures prices in Indian stock market. Rest of the paper is organized as follows. In section 2, we give a brief description of SVM for regression. In section 3 some related works are presented followed by proposed method for future prediction in Section 4. In Section 5 and 6 results and conclusions are discussed.

2. SVM FOR REGRESSION

Given a set of training data $\{(x_1, y_1), \dots, (x_\ell, y_\ell)\}$, where each $x_i \in X \subset R^n$, (X denotes the input space of the sample) and corresponding target value $y_i \in R$ for $i=1, \dots, \ell$ (where ℓ corresponds to the size of the training data), the objective of the regression problem is to determine a function that can approximate the value of y for an x not in the training set.

The estimating function f is taken in the form:

$$f(x) = (w \cdot \Phi(x)) + b \quad (1)$$

where $w \in R^m$, $b \in R$ is the bias and Φ denotes a non-linear function from R^n to high dimensional space R^m ($m > n$). The objective is to find the value of w and b such that values of $f(x)$ can be determined by minimizing the risk:

$$R_{reg}(f) = C \sum_{i=1}^n L_\epsilon(y_i, f(x_i)) + \frac{1}{2} \|w\|^2 \quad (2)$$

where L_ϵ is the extension of ϵ -insensitive loss function originally proposed by Vapnik [4] and defined as:

$$L_\epsilon(y, z) = \begin{cases} |y - z| - \epsilon, & |y - z| \geq \epsilon \\ 0, & \text{otherwise} \end{cases} \quad (3)$$

Introducing the slack variables ζ_i and ζ_i^* the above problem may be reformulated as

(P) Minimize $C[\sum_{i=1}^l (\zeta_i + \zeta_i^*)] + \frac{1}{2} \|w\|^2$ subject to the following constraints [3].

$$y_i - w \cdot \Phi(x_i) - b \leq \epsilon + \zeta_i$$

$$w \cdot \Phi(x_i) + b - y_i \leq \epsilon + \zeta_i^*$$

$$\zeta_i \geq 0$$

$$\zeta_i^* \geq 0$$

for $i = 1, 2, \dots, l$ and where C above is an user specified constant.

Solution of the above problem (P) using primal dual method

leads to the following dual problem: [3]

Determine the Lagrange multipliers $\{\alpha_i\}_{i=1}^l$ and $\{\alpha_i^*\}_{i=1}^l$ that maximize the objective function.

$$Q(\alpha_i, \alpha_i^*) = \sum_{i=1}^l y_i (\alpha_i - \alpha_i^*) - \frac{1}{2} \sum_{i=1}^l \sum_{j=1}^l (\alpha_i - \alpha_i^*) (\alpha_j - \alpha_j^*) K(x_i, x_j)$$

subjected to the following conditions:

$$(1) \quad \sum_{i=1}^l (\alpha_i - \alpha_i^*) = 0$$

$$(2) \quad \begin{aligned} 0 &\leq \alpha_i \leq C \\ 0 &\leq \alpha_i^* \leq C \end{aligned}$$

for $i = 1, 2, \dots, l$

where C is a user specified constant and $K: X \times X \rightarrow \mathbb{R}$ is the Mercer Kernel defined by:

$$K(x, z) = \Phi(x) \cdot \Phi(z) \quad (4)$$

The solution of the Primal yields

$$w = \sum_{i=1}^l (\alpha_i - \alpha_i^*) \Phi(x_i) \quad (5)$$

An b is calculated using Karush-Kuhn-Tucker (KKT) conditions

$$\begin{aligned} \alpha_i (\varepsilon + \zeta_i - y_i + w \cdot \Phi(x_i) + b) &= 0, \\ \alpha_i^* (\varepsilon + \zeta_i^* + y_i - w \cdot \Phi(x_i) - b) &= 0, \end{aligned} \quad (6)$$

$$(C - \alpha_i) \zeta_i = 0$$

and

$$(C - \alpha_i^*) \zeta_i^* = 0$$

$$\text{for } i = 1, 2, \dots, l \quad (7)$$

Since $\alpha_i, \alpha_i^* = 0$ and $\zeta_i^* = 0$ for $\alpha_i^* \in (0, C)$, b can be computed as follows:

$$\begin{aligned} b &= y_i - w \cdot \Phi(x_i) - \varepsilon \quad \text{for } 0 < \alpha_i < C \\ b &= y_i - w \cdot \Phi(x_i) + \varepsilon \quad \text{for } 0 < \alpha_i^* < C \end{aligned} \quad (8)$$

For those α_i and α_i^* for which the x_i 's corresponding to $0 < \alpha_i < C$ and $0 < \alpha_i^* < C$ are called support vectors.

Using expression for w and b in equations (6) and (7), $f(x)$ is computed as:

$$f(x) = \sum_{i=1}^n (\alpha_i - \alpha_i^*) (\Phi(x_i) \cdot \Phi(x)) + b$$

$$= \sum_{i=1}^l (\alpha_i - \alpha_i^*) K(x_i, x) + b \quad (9)$$

It is to be noted that we do not require function Φ to compute $f(x)$ which is one of the advantages of using the kernel.

Advantages of SVM

The support vector machine has the following specific advantages over other machine learning methods for supervised learning:

- training a support vector machine involves optimization of a convex function with linear constraint. This problem has a unique global minimum which in turn overcome strutting to local minima observed in neural network.
- the constructed model has an explicit dependence only on the support vectors, which reduces the computational cost.
- it scales relatively well to high dimensional data and the trade-off between complexity and empirical error can be controlled explicitly by the appropriate choices of C and ε .

3. RELATED WORK

In the past, most prediction models were based on conventional statistical methods, like time-series and multivariate analysis. In recent years, however, machine learning methodologies, including the artificial neural networks, genetic algorithms (GA) and fuzzy technologies, have been used for predictions.

In this paper, we apply two machine learning techniques, back propagation neural networks and support vector-machines, to construct the prediction models for forecasting the five major futures index of Indian markets. Refenes et al. [5], Tsibouris et al. [6] and Steiner et al. [7] have used ANN Models for predicting the prices of stocks in the UK, US, and German markets respectively. Wittkemper et al. [8] and Shazly et al. [9] have used neural network along with genetic algorithmic (hybrid system) for predicting the prices of stocks and currency for German, UK, Japan, and Swiss markets. Tay and Cao [10] have developed pricing model for futures in US market using SVMs, Gestel et al. [11] have used LS-SVM for T-Bill rate and stock index pricing in US and German market. Chen et al. [12] applied support-vector machines and back propagation neural networks for six Asian stock markets. Tasi [13] bankruptcy prediction using SVM and comparisons to neural network has been done. Tay et al. [14] have used Chicago mercantile markets data sets to see the feasibility of applying SVM for financial time series forecasting.

4. PROPOSED METHODOLOGY

4.1 Data Sets

The model of forecasting is used with the following real index futures data collected from the National Stock Exchange (NSE) of India Limited. They are S&P CNX NIFTY, BANK NIFTY, S&P CNX 500, CNX INFRA, and CNX 100. We have taken 990 samples for each of the futures contracts mentioned above. The time period for each contract is from 1st January, 2007 to 31st December, 2010.

Table 1 - Description of Data Sets

INDEX FUTURES	HIGH	LOW	MEAN	MEDIAN	SD
S&P CNX NIFTY	6312.45	2524.2	4627.85	4713.8	913.018
BANK NIFTY	13268.7	3339.7	7662.5	7350.45	2236.99
S&P CNX 500	5502.6	1966.8	3794.94	3837.0	816.909
CNX INFRA	6260.66	2006.7	3611.63	3499.385	806.942
CNX 100	6280.45	2388.6	4509.59	4545.95	947.771

The data collected consists of daily previous closing price, open price, high price, low price, traded volume, and traded value. The daily closing prices are used as the data sets. Table 1 shows high price, low price, mean, median and standard deviation of the five futures prices collected for our experiment.

4.2 Preprocessing

Choosing a suitable forecasting horizon is the first step in financial forecasting. From the trading aspect, the forecasting horizon should be sufficiently long so that the over-trading resulting in excessive transaction costs could be avoided. From the prediction aspect, the forecasting horizon should be short enough as the persistence of financial time series is of limited duration. As suggested by Thomason [15] a forecasting horizon of five days is a suitable choice for the daily data. As the precise values of the daily prices is often not as meaningful to trading as its relative magnitude, and also the high-frequency components in financial data are often more difficult to successfully model, the original closing price is transformed into a five-day relative difference in percentage of price (RDP). The input variables are determined from four lagged RDP values based on five-day periods (RDP-5, RDP-10, RDP-15, and RDP-20) and one transformed closing price which is obtained by subtracting a 15-day exponential moving average (EMA15) from the closing price. The subtraction is performed to eliminate the trend in price as the maximum value and the minimum value is in the ratio of about 2: 1 in all the five data sets. The optimal length of the moving day is not critical, but it should be longer than the forecasting horizon of five days. EMA15 is used to maintain as much of the information contained in the original closing price as possible since the application of the RDP transform to the original closing price may remove some useful information. The output variable RDP+5 is obtained by first smoothing the closing price with a three-day exponential moving average, because the application of a smoothing transform to the dependent variable generally enhances the prediction performance of neural networks. The calculations for all the indicators are given in table 2 below [16].

Table 2 - Performance Indicator

INDICATOR	CALCULATION
<i>Input Variables</i>	
EMA15	$p(i) - \text{EMA15}(i)$
RDP-5	$(p(i) - p(i-5)) / p(i-5) - 100$
RDP-10	$(p(i) - p(i-10)) / p(i-10) - 100$
RDP-15	$(p(i) - p(i-15)) / p(i-15) - 100$
RDP-20	$(p(i) - p(i-20)) / p(i-20) - 100$
<i>Output Variables</i>	
RDP+5	$\frac{(p(i+5) - p(i)) / p(i)}{p(i) - \text{EMA}_3(i)} - 100$

EMA_n(i) is the n-day exponential moving average of the ith day and p(i) is the closing price of the ith day.

Since outliers may make it difficult or time-consuming to arrive at an effective solution for the SVMs, RDP values beyond the limits of ± 2 standard deviations are selected as outliers. They are replaced with the closest marginal values. Another pre-processing technique used in this study is data scaling. All the data points are scaled into the range of $[-0.9; 0.9]$ as the data points include both positive values and negative values. All of the five data sets are partitioned into three parts according to the time sequence.

4.3 Performance Criteria

The prediction performance is evaluated using the following statistical metrics, namely, the normalized mean squared error (NMSE), mean absolute error (MAE) and directional symmetry (DS) [14]. The definitions of these criteria can be found in Table 3.

NMSE and MAE are the measures of the deviation between the actual and predicted values. The smaller the values of NMSE and MAE, the closer are the predicted time series values to the actual values. DS provides an indication of the correctness of the predicted direction of RDP+5 given in the form of percentages (a large value suggests a better predictor).

Table 3 - Performance Metrics & Their Calculation

Metrics	Calculation
NMSE	$\frac{1}{\sigma^2 n} \sum_{i=1}^n (y_i - \hat{y}_i)^2$ $\sigma^2 = \frac{1}{n-1} \sum_{i=1}^n (y_i - \bar{y})^2$ $\bar{y} = \frac{1}{n} \sum_{i=1}^n y_i$
MAE	$\frac{1}{n} \sum_{i=1}^n y_i - \hat{y}_i $
DS	$\frac{100}{n} \sum_{i=1}^n d_i$ $d_i = \begin{cases} 1 & (y_i - y_{i-1})(\hat{y}_i - \hat{y}_{i-1}) \geq 0 \\ 0 & \text{otherwise} \end{cases}$

N is the total number of data patterns. y and \hat{y} represent the actual and predicted output value

4.4 Computation Techniques

We have applied Vapnik's SVM for regression by using LS-SVM tool box. The typical kernel functions used in SVRs are the Polynomial Kernel $k(x, y) = (x * y + 1)^d$ and the Gaussian Kernel $k(x, y) = \exp(-(|x - y|^2 / \delta^2))$, where d is the degree of the polynomial kernel and δ^2 is the bandwidth of the Gaussian Kernel. We have taken Gaussian Kernel function because it performs well under general smoothness assumptions. Polynomial Kernel gives inferior result compared to Gaussian Kernel and takes a longer time in training SVMs [14]. In the SVR, those values of δ^2 & epsilon are taken, that produces the best result on the validation set of our data.

To compare the prediction performance of SVM with Neural Network (NN), BP models are implemented with Levenberg Marquardt learning has been used.

5. RESULTS AND DISCUSSION

After the training with BP model and SVR, the forecasted price and the actual price for the test data are exhibited in the following figures.

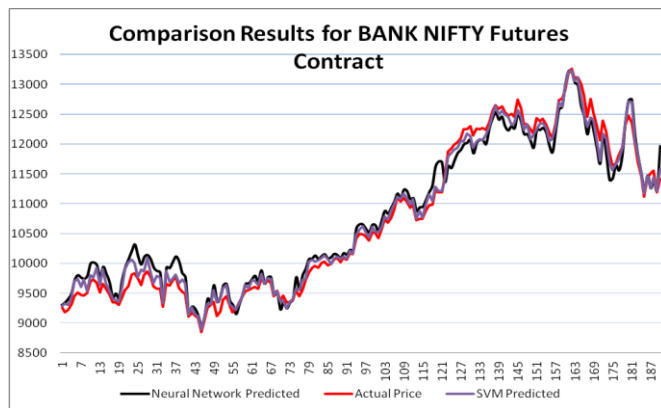


Fig 1 – Comparison of BP and SVM for BANK NIFTY Futures Contract

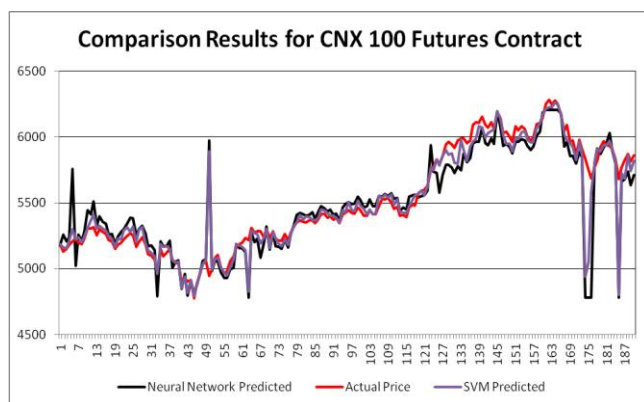


Fig 2 - Comparison of BP and SVM for CNX 100 Futures Contract

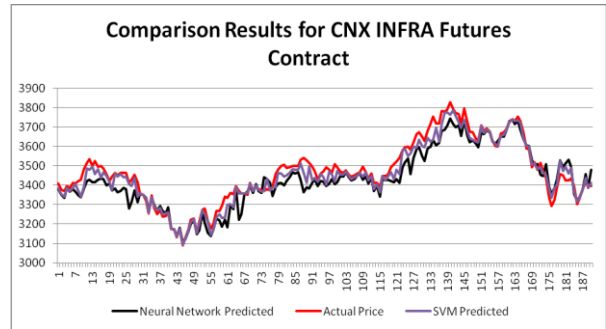


Fig 3 - Comparison of BP and SVM for CNX INFRA Futures Contract

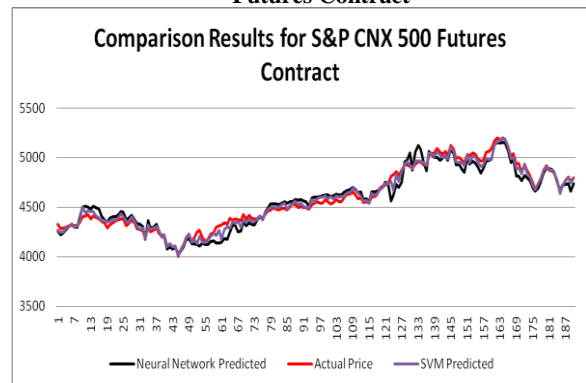


Fig 4 - Comparison of BP and SVM for S&P CNX 500 Futures Contract

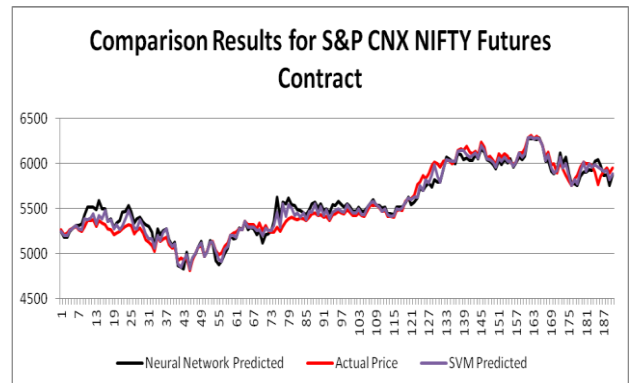


Fig 5 - Comparison of BP and SVM for S&P CNX NIFTY Futures Contract

The performance criteria set for our experiment showed a very good agreement of the predicted price with actual price when SVM method is used. The NSME for all the futures stock index taken into consideration fall in the range of 0.9299 to 1.1521. The MAE fall in the range of 0.2379 to 0.3887 and the last criterion DS starts from 55.17 to 91.2512.

5.1 Comparison of Results

The forecasting results of the SVM and BPN for the test set are collected in the table given below which shows SVM outperforms the BPN in most of the cases.

Table 4- Comparison of the results of SVM & BPN

Futures Details	NMSE		MAE		DS	
	SVM	BPN	SVM	BPN	SVM	BPN
BANK NIFTY	0.929	1.335	0.238	0.250	91.251	83.880
CNX 100	1.012	1.333	0.382	0.381	87.17	85.672
CNX INFRA	1.032	1.471	0.293	0.299	88.192	80.895
S&P CNX 500	1.152	1.378	0.332	0.336	85.671	80.597
S&P CNX NIFTY	1.019	1.100	0.389	0.403	85.17	88.059

SVM provides a smaller NMSE and MAE and larger DS than those of BPN in most of the cases.

6. CONCLUSION

In this research work, we have examined the feasibility of applying two machine learning models, Support Vector Machines (SVM) and Back Propagation Neural Network (BPN), to financial time-series forecasting for the futures trading in Indian derivative markets.

Our experiments demonstrated that:

- SVMs provide a promising alternative tool to the BP neural network for financial time series forecasting.
- SVMs forecast better than that BPN, as shown in our results, SVM provides a smaller NMSE (Normalized Mean Square Error) and MAE (Mean Absolute Error) and larger DS (Directional Symmetry) than of BPN in most of the cases. This is because SVMs adopt the Structural Risk Minimization Principle, eventually leading to better generalization than that conventional technique.

7. ACKNOWLEDGEMENT

The authors thank the referees for the suggestions which improved the presentation of the paper.

8. REFERENCES

- [1] Hall JW. Adaptive selection of U.S. stocks with neural nets. In: GJ Deboeck (Ed.), *Trading on the edge: neural, genetic, and fuzzy systems for chaotic financial markets*. New York: Wiley, 1994.
- [2] Yaser SAM, Atiya AF. *Introduction to financial forecasting*. Applied Intelligence 1996; 6: 205–13.
- [3] Haykin, S., 1999, *Neural Networks: A Comprehensive Foundation*, 2nd Ed. (Englewood Cliffs, NJ: Prentice-Hall).
- [4] V. Vapnik., 1995, *The Nature of Statistical Learning Theory*. Springer, N.Y. ISBN 0-387-94559-8.
- [5] Refenes, A-P., Zapranis, A.D. and Francis, G. (1995) 'Modeling stock returns in the framework of APT: a comparative study with regression models', *Neural Networks in the Capital Markets*, pp.101–125.
- [6] Tsibouris, G. and Zeidenberg, M. (1995) 'Testing the efficient markets hypothesis with gradient descent algorithms', *Neural Networks in the Capital Markets*, pp.127–136.

[7] Steiner, M. and Wittkemper, H-G. (1995) 'Neural networks as an alternative stock market model', *Neural Networks in the Capital Markets*, pp.135–147.

[8] Wittkemper, H-G. and Steiner, M. (1996) 'Using neural networks to forecast the systematic risk of stocks', *European Journal of Operational Research*, Vol. 90, pp.577–588.

[9] Shazly, M.R.E. and Shazly, H.E.E. (1999) 'Forecasting currency prices using genetically evolved neural network architecture', *International Review of Financial Analysis*, Vol. 8, No. 1, pp.67–82.

[10] Tay FEH, Cao LJ., 2001, Improved financial time series forecasting by combining support vector machines with self-organizing feature map. *Intelligent Data Analysis*.

[11] Gestel, T.V., Suykens, J.A.K., Baestaens, D-E., Lambrechts, A., Lanckriet, G., Vandaele, B., Moor, B.D. and Vandewalle, J. (2001) 'Financial time-series prediction using least squares support vector machines within the evidence framework', *IEEE Transactions on Neural Networks*, Vol. 12, No. 4, pp.809–821.

[12] Wun-Hua Chen and Jen Ying Shih., "Comparison of support vector machines and back propagation neural networks in forecasting the six major Asian stock markets", *International Journals Electronics Finance*, Vol. 1, 2006, pp.49-67.

[13] Chih Fong Tsai., "Financial decision support using neural network and support vector machines", *Expert Systems*, Vol. 25, 2008, pp.380-393.

[14] Tay, F.E.H. and Cao, L., "Application of support vector machines in financial time-series forecasting", *Omega*, Vol. 29, 2001, pp.309–317.

[15] Thomason M., "The practitioner methods and tool", *Journal of Computational Intelligence in Finance* 1999; pp.35–45.

[16] Bao, Yukun., Lu, Yansheng., Zhang, Jinlong., "Forecasting Stock Price by SVMs Regression", *Artificial Intelligence Methodology Systems and Applications* Springer 2004; pp.295-303.

9. AUTHORS PROFILE

Shom Prasad Das currently working as a Faculty at National Institute of Science & Technology, Berhampur, Odisha. He has total of 11 years of experience in Industry and Academics. He is continuing his research in the area of Financial Engineering. He is M.Tech. in Computer Science & Engineering from Biju Patnaik University of Technology, Odisha.

Sudarsan Padhy is currently Professor of Computer Science & Engineering at IIIT-Bhubaneswar. Prior to this he was Professor of Mathematics at Utkal University, India. He obtained his Ph.D. degree in Mathematics from Utkal University in 1979 and Postdoctoral research at University of Freiburg, Germany during 1980-81. He has over fifty published research papers and five books to his credit extending over Fluid dynamics, Finite difference and Finite element method for solving partial differential equations, Operation research, Parallel algorithms, Computational finance and Computational biology.