

## INDIAN STOCK MARKET TREND PREDICTION USING SUPPORT VECTOR MACHINE

M.Suresh Babu\*

Dr. N.Geethanjali\*\*

Prof B.Satyanarayana\*\*\*

---

### ABSTRACT

*Stock return predictability has been a subject of great controversy. The debate followed issues from market efficiency to the number of factors containing information on future stock returns. The analytical tool of support vector regression on the other hand, has gained great momentum in its ability to predict time series in various applications and also in finance (Smola and Schölkopf, 1998). Support vector machines (SVM) are employed to predict stock market daily trends: ups and downs. The purpose is to examine the effect of macroeconomic information and technical analysis indicators on the accuracy of the classifiers. The construction of a prediction model requires factors that are believed to have some intrinsic explanatory power. These explanatory factors fall largely into two categories: fundamental and technical. Fundamental factors include for example macro economical indicators, which however, are usually only infrequently published. Technical factors are based solely on the properties of the underlying time series and can therefore be calculated at the same frequency as the time series. Since this study applies support vector regression to high frequent data, only technical factors are considered. It is found that macroeconomic information is suitable to predict stock market trends than the use of technical indicators. In addition, the combination of the two sets of predictive inputs does not improve the forecasting accuracy. Furthermore, the prediction accuracy improves when trading strategies are considered.*

*Support vector machine (SVM) is a very specific type of learning algorithms characterized by the capacity control of the decision function, the use of the kernel functions and the sparsity of the solution. In this paper, we investigate the predictability of financial movement direction with SVM by forecasting the weekly movement direction of BSE30 index. To evaluate the forecasting ability of SVM, we compare its performance with those of Linear Discriminant Analysis, Quadratic Discriminant Analysis and Elman Backpropagation Neural Networks. The experiment results show that SVM outperforms the other classification*

*methods. Further, we propose a combining model by integrating SVM with the other classification methods. The combining model performs best among all the forecasting methods.*

**Keywords:***Support Vector Machines, Classification, Stock Market, technical indicators.*

---

\*Principal, Intel Institute of Science, Anantapur, Andhra Pradesh, India.

\*\*Associate Professor, Department of Computer Science, S.K. University, Anantapur. India

\*\*\*Professor & Chairman, Board of Studies, Department of Computer Science, Sri Krishnadevaraya University, Anantapur.

## 1. INTRODUCTION

Forecasting stock market behavior is a very difficult task since its dynamics are complex and non-linear. For instance, stock return series are generally noisy and may be influenced by many factors; such as the economy, business conditions, and political events to name a few. Indeed, empirical finance shows that publicly available data on financial and economic variables may explain stock return fluctuations in the Indian Stock Market. For instance, a number of applications have been proposed to forecast stock market returns with macroeconomic variables with the use of neural networks and Bayesian networks and support vector machines. On the other hand, technical indicators have been also used to predict stock market movements using neural networks, adaptive fuzzy inference system, and fuzzy logic. The literature shows that economic variables and technical indicators have achieved success in predicting the stock market. However, none of the previous studies have compared the performance of the economic information and technical indicators in terms of prediction accuracy.

The financial market is a complex, evolutionary, and non-linear dynamical system. The field of financial forecasting is characterized by data intensity, noise, non stationary, unstructured nature, high degree of uncertainty, and hidden relationships. Many factors interact in finance including political events, general economic conditions, and traders' expectations. Therefore, predicting finance market price movements is quite difficult. Increasingly, according to academic investigations, movements in market prices are not random. Rather, they behave in a highly non-linear, dynamic manner. The standard random walk assumption of futures prices may merely be a veil of randomness that shrouds a noisy non-linear process.

Support vector machine (SVM) is a very specific type of learning algorithms characterized by the capacity control of the decision function, the use of the kernel functions and the sparsity of the solution. Established on the unique theory of the structural risk minimization principle to estimate a function by minimizing an upper bound of the generalization error, SVM is shown to be very resistant to the over fitting problem, eventually achieving a high generalization performance.

Another key property of SVM is that training SVM is equivalent to solving a linearly constrained quadratic programming problem so that the solution of SVM is always unique and globally optimal, unlike neural networks training which requires nonlinear optimization with the danger of getting stuck at local minima.

Some applications of SVM to financial forecasting problems have been reported recently. In most cases, the degree of accuracy and the acceptability of certain forecasts are measured by the estimates' deviations from the observed values. For the practitioners in financial market, forecasting methods based on minimizing forecast error may not be adequate to meet their objectives. In other words, trading driven by a certain forecast with a small forecast error may not be as profitable as trading guided by an accurate prediction of the direction of movement. The goal of this study is to predict stock price movements only from the statistical properties of the underlying financial time series and to explore the predictability of financial market movement direction with SVM. Therefore, financial indicators are extracted from the time series, which are then used by a support vector regression (SVR) to predict market movement.

### 2.2.2 Support vector machines

Support Vector Machines (SVM) is a supervised statistical learning technique introduced by Vapnik. It is one of the standard tools for machine learning successfully applied in many different real-world problems. For instance, they have been successfully applied in financial time series trend prediction. The SVM were originally formulated for binary classification. The SVM seek to implement an optimal marginal classifier that minimizes the structural risk in two steps. First, SVM transform the input to a higher dimensional space with a kernel (mapping) function. Second, SVM linearly combine them with a weight vector to obtain the output. As result, SVM provide very interesting advantages. They avoid local minima in the optimization process. In addition, they offer scalability and generalization capabilities. For instance, to solve a binary classification problem in which the output  $y \in \{-1, +1\}$  SVM seek for a hyper-plane  $w \cdot \Phi(x) + b = 0$  to separate the data from classes  $+1$  and  $-1$  with a maximal margin. Here,  $x$  denotes the input feature vector,  $w$  is a weight vector,  $\Phi$  is the mapping function to a higher dimension, and  $b$  is the bias used for classification of samples. The maximization of the margin is equivalent to minimizing the norm of  $w$ . Thus, to find  $w$  and  $b$ , the following optimization problem is solved:

$$\text{Minimize : } \|w\|^2 + C \sum_{i=1}^n \xi_i$$

$$\text{S.t } y_i (w \cdot \Phi(x_i) + b) \geq 1 - \xi_i \quad \xi_i \geq 0 \quad i = 1, \dots, n$$

where  $C$  is a strictly positive parameter that determines the tradeoff between the maximum margin and the minimum classification error,  $n$  is the total number of samples, and generalization and  $\xi_i$  is the error magnitude of the classification.

The conditions ensure that no training example should be within the margins. The number of training errors and examples within the margins is controlled by the minimization of the term:

$$\sum_{i=1}^n \xi_i$$

The solution to the previous minimization problem gives the decision frontier:

$$f(x) = \sum y_i \alpha_i \Phi(x_i) \Phi(x) + b$$

$x_i$

Where each  $\alpha_i$  is a lagrange coefficient. As mentioned before the role of the kernel function is to implicitly map the input vector into a high-dimensional feature space to achieve better separability. In this study the polynomial kernel is used since it is a global kernel. For instance, global kernels allow data points that are far away from each other to have an influence on the kernel values as well.

$$K(x, x_i) = \Phi(x_i) \Phi(x) = ((x_i \cdot x) + 1)^d$$

where the kernel parameter  $d$  is the degree of the polynomial to be used. In this study,  $d$  is set to 2. Finally, the optimal decision separating function can be obtained as follows:

$$y = \text{sign} \left( \sum_{i=1}^n y_i \alpha_i K(x_i, x) + b \right)$$

## 2. THEORY OF SVM IN CLASSIFICATION

The indicators are arbitrarily chosen among a high variety of financial indicators. The chosen indicators include price differences, moving averages, relative strength and so called stochastic indicators as shown in the figure. These indicators are then preprocessed in the sense that the mean vector is subtracted and each indicator time series is divided by its variance in order to receive indicator values with zero mean and unit variance. Before the SVR model is trained, the parameters of the SVR model are optimized using a cross validation procedure on a training set. After that, the optimized model is used to predict financial market movement.

In the process of model selection, models are chosen only on the basis of performance over out of sample data, in order to avoid the critique of judging the model on the basis of in sample performance. The model selection is based on a cross validation procedure commonly used in Data Mining.

Our main results show that stock market prediction based on support vector regression is significantly outperforming a random stock market prediction. However, the prediction in average is only correct in 50.69 percent of times with a standard deviation of 0.26 percent.

We present a basic theory of the support vector machine model. Let  $D$  be the smallest radius of the sphere that contains the data (example vectors). The points on either side of the separating hyperplane have distances to the hyperplane. The smallest distance is called the margin of separation. The hyperplane is called optimal separating hyperplane (OSH), if the margin is maximized. Let  $q$  be the margin of the optimal hyperplane. The points that are distance  $q$  away from the OSH are called the support vectors.

Consider the problem of separating the set of training vector belonging to two separate classes,  $G = \{(x_i, y_i), i = 1, 2, \dots, N\}$  with a hyperplane  $w^T \phi(x) + b = 0$  ( $x_i \in \mathbb{R}^n$  is the  $i^{\text{th}}$  input vector,  $y_i \in \{-1, 1\}$  is known binary target), the original SVM classifier satisfies the following conditions:

$$w^T \phi(x_i) + b \geq 1 \text{ if } y_i = 1, \quad (1)$$

$$w^T \phi(x_i) + b \leq -1 \text{ if } y_i = -1, \quad (2)$$

or equivalently,

$$y_i [w^T \phi(x_i) + b] \geq 1 \quad i = 1, 2, \dots, N, \quad (3)$$

where  $\phi: \mathbb{R}^n \rightarrow \mathbb{R}^m$  is the feature map mapping the input space to a usually high dimensional feature space where the data points become linearly separable.

The distance of a point  $x_i$  from the hyperplane is

$$d(x_i, w, b) = \frac{|w^T \phi(x_i) + b|}{\|w\|^2} \quad (4)$$

The margin is  $2/|w|$  according to its definition. Hence, we can find the hyperplane that optimally separates the data by solving the optimization problem:

$$\text{Min } \mathcal{O}(w) = \frac{1}{2} \|w\|^2 \quad (5)$$

under the constraints of Eq. (3).

The solution to the above optimization problem is given by the saddle point of the Lagrange function

$$L_{P1} = \frac{1}{2} \|w\|^2 - \sum_{i=1}^N \alpha_i [y_i (w^T \phi(x_i) + b) - 1] \quad (6)$$

under the constraints of Eq. (3), where  $\alpha_i$  are the nonnegative Lagrange multipliers. So far the discussion is restricted to the case where the training data is separable. To generalize the problem to the non-separable case, slack variable  $\xi_i$  is introduced such that

$$y_i[w^T \varphi(x_i) + b] \geq 1 - \xi_i \quad (\xi_i \geq 0 \quad i = 1, 2, \dots, N) \quad (7)$$

Thus, for an error to occur the corresponding  $\xi_i$  must exceed unity, so  $\sum_{i=1}^N \xi_i$  is an upper bound on the number of training errors. Hence, a natural way to assign an extra cost for errors

$$\text{is to change the objective function from Eq. (5) to } \min \phi(w, \xi) = \frac{1}{2} \|w\|^2 + C \sum_{i=1}^N \xi_i \quad (8)$$

under the constraints of Eq. (7), where C is a positive constant parameter used to control the tradeoff between the training error and the margin. In this paper, we choose C =50 based on our experiment experiences. Similarly, solve the optimal problem by minimizing its Lagrange function

$$L_{P2} = \frac{1}{2} \|w\|^2 + C \sum_{i=1}^N \xi_i - \sum_{i=1}^N \alpha_i [y_i(w^T \varphi(x_i) + b) - 1 + \xi_i] - \sum_{i=1}^N \mu_i \xi_i \quad (9)$$

under the constraints of Eq. (7), where  $\alpha_i, \mu_i$  are the non-negative Lagrange multipliers.

The Karush–Kuhn–Tucker (KKT) conditions [16] for the primal problem are

$$\frac{\partial L_{P2}}{\partial w} = w - \sum_{i=1}^N \alpha_i y_i \varphi(x_i) = 0, \quad (10)$$

$$\frac{\partial L_{P2}}{\partial b} = - \sum_{i=1}^N \alpha_i y_i = 0, \quad (11)$$

$$\frac{\partial L_{P2}}{\partial \xi_i} = C - \alpha_i - \xi_i = 0, \quad (12)$$

$$y_i [w^T \varphi(x_i) + b] \geq 1 - \xi_i, \quad (13)$$

$$\xi_i \geq 0 \quad (14)$$

$$\alpha_i \geq 0, \quad (15)$$

$$\mu \geq 0, \quad (16)$$

$$\alpha_i [y_i (w^T \varphi(x_i) + b) - 1 + \xi_i] = 0, \quad (17)$$

$$\mu_i \xi_i = 0 \quad (18)$$

Hence,

$$w = \sum_{i=1}^N \alpha_i y_i \varphi(x_i). \quad (19)$$

We can use the KKT complementarily conditions, Eqs. (17) and (18), to determine b. Note that Eq. (12) combined with Eq. (18) shows that  $\xi_j = 0$  if  $\alpha_j < C$ . Thus we can simply take any training data for which  $0 < \alpha_j < c$  to use Eq. (17) (with  $\xi_j = 0$ ) to compute b.

$$b = y_j - w^T \varphi(x_j) \quad (20)$$

It is numerically reasonable to take the mean value of all b resulting from such computing.

Hence,

$$b = \frac{1}{N_s} \sum_{0 < \alpha_j < C} [(y_j - w^T \varphi(x_j))], \quad (21)$$

where  $N_s$  is the number of the support vectors.

For a new data  $x$ , the classification function is then given by

$$f(x) = \text{Sign}(w^T \varphi(x) + b) \quad (22)$$

Substituting Eqs. (19) and (21) into Eq. (22), we get the final classification function

$$f(x) = \text{Sign} \left( \sum_{i=1}^N \alpha_i y_i \varphi(x_i)^T \varphi(x) + \frac{1}{N_s} \sum_{0 < \alpha_j < C} \left( y_j - \sum_{i=1}^N \alpha_i y_i \varphi(x_i)^T \varphi(x_j) \right) \right) \quad (23)$$

If there is a kernel function such that  $K(x_i, x_j) = \varphi(x_i)^T \varphi(x_j)$ , it is usually unnecessary to explicitly know what  $\varphi(x)$  is, and we only need to work with a kernel function in the training algorithm.

Therefore, the non-linear classification function is

$$f(x) = \text{Sign} \left( \sum_{i=1}^N \alpha_i y_i K(x_i, x) + \frac{1}{N_s} \sum_{0 < \alpha_j < C} \left( y_j - \sum_{i=1}^N \alpha_i y_i K(x_i, x_j) \right) \right) \quad (24)$$

Any function satisfying Mercer's condition [17] can be used as the kernel function. In this investigation, the radial kernel  $K(s, t) = \exp(-1/10 \|s-t\|^2)$  is used as the kernel function of the SVM because the radial kernel tends to give good performance under general smoothness assumptions. Consequently, it is especially useful if no additional knowledge of the data is available.

### 3. Experiment design

Several financial indicators are calculated in order to reduce dimensionality of the time series:

- $RDP_t = \frac{p(t) - p(t-1)}{p(t)}$  : The relative price difference of prices  $p(t)$  at time  $t$  and  $p(t-1)$  at time  $t-1$
- $EMA_t = \sum_{i=-n}^0 \exp(i) p(t-i)$  : The exponential moving average of the prices  $p(t)$
- $RSI_t = \frac{U_{[t-n;t]}}{D_{[t-n;t]}}$  : The relative strength indicator of the number of upward movement  $U_{[t-n;t]}$  and downward movements  $D_{[t-n;t]}$  in the period of  $t-n$  until time  $t$
- $Stochastic_t = \frac{p(t) - L_{[t-n;t]}}{H_{[t-n;t]} - L_{[t-n;t]}}$  : The stochastic indicator of the stock price  $p(t)$ , lowest stock price  $L_{[t-n;t]}$  and highest stock price  $H_{[t-n;t]}$  in the period of  $t-n$  until time  $t$ .

The figure illustrates some of the properties of the indicators derived as above from a random time series. The kernel densities are estimated for each indicator with a bandwidth of 0.001.



Note, that the RDP and EMA indicator are rather gaussian distributed, while the RSI and Stochastic indicators have several modes and especially the Stochastic indicator seems to be a mixture of two different gaussian distributions.

In our empirical analysis, we set out to examine the weekly changes of the BSE30 Index. The BSE30 Index is calculated and disseminated. It measures the composite price performance of 225 highly capitalized stocks trading on the Bombay Stock Exchange (BSE), representing a broad cross-section of Indian industries. Trading in the index has gained unprecedented popularity in major financial markets around the world. Futures and options contracts on the BSE30 Index are currently traded on the Securities and Exchange Board of India (SEBI), the National Stock Exchange (NSE). The increasing diversity of financial instruments related to the BSE30 Index has broadened the dimension of global investment opportunity for both individual and institutional investors. There are two basic reasons for the success of these index trading vehicles. First, they provide an effective means for investors to hedge against potential market risks. Second, they create new profit making opportunities for market speculators and arbitrageurs. Therefore, it has profound implications and significance for researchers and practitioners alike to accurately forecast the movement direction of BSE30 Index.

### **3.0. MODEL INPUTS SELECTION**

Most of the previous researchers have employed multivariate input. Several studies have examined the cross sectional relationship between stock index and macroeconomic variables. The potential macroeconomic input variables which are used by the forecasting models include term structure of interest rates (TS), short-term interest rate (ST), long term interest rate (LT), consumer price index (CPI), industrial production (IP), government consumption (GC), private consumption (PC), gross national product (GNP) and gross domestic product (GDP). However, Indian interest rate has dropped down to almost zero since 1990. Other macroeconomic variables weekly data are not available for our study.

Indian consumption capacity is limited in the domestic market. The economy growth has a close relationship with Indian export. The largest export target for India is the United States of America (USA), which is the leading economy in the world. Therefore, the economic condition of USA influences Indian economy, which is well represented by the BSE30 Index. As the BSE30 Index to Indian economy, the S& P 500 Index is a well-known indicator of the economic condition in USA. Hence, the S& P 500 Index is selected as model input. Another

import factor that affects the Indian export is the exchange rate of US Dollars against Indian Rupee (Rs), which is also selected as model input. The prediction model can be written as the following function:

$$\text{Direction}_t = F(S_{t-1}^{\text{S\&P500}}, S_{t-1}^{\text{IND}}), \quad (25)$$

where  $S_{t-1}^{\text{S\&P500}}$  and  $S_{t-1}^{\text{IND}}$  are first order difference natural logarithmic transformation to the raw S & P 500 index and IND at time  $t-1$ , respectively. Such transformations implement an effective detrending of the original time series.  $\text{Direction}_t$  is a categorical variable to indicate the movement direction



Fig. 1. First-order difference natural logarithmic weekly prices of BSE Index, S & P 500 Index. (observations from October 2010 to September 2011).

of BSE30 Index at time  $t$ . If BSE30 Index at time  $t$  is larger than that at time  $t - 1$ ,  $\text{Direction}_t$  is 1. Otherwise,  $\text{Direction}_t$  is  $-1$ . The above model inputs selection is only based on a macroeconomic analysis. As shown in Fig. 1, the behaviours of the BSE30 Index, the S & P 500 Index are very complex. It is impossible to give an explicit formula to describe the underlying relationship between them.

### 3.1. Data collection

We obtain the historical data from the finance section of Yahoo and the Bombay Stock Exchange and National Stock Exchange respectively. The whole data set covers the period from January 1, 2007 to December 31, 2010, a total of 694 pairs of observations. The data set is divided into two parts. The first part (652 pairs of observations) is used to determine the specifications of the models and parameters. The second part (42 pairs of observations) is reserved for out-of-sample evaluation and comparison of performances among various forecasting models.

### 3.2. Comparisons with other forecasting methods

To evaluate the forecasting ability of SVM, we use the random walk model (RW) as a benchmark for comparison. RW is a one step ahead forecasting method, since it uses the current actual value to predict the future value as follows:

$$\hat{y}_{t+1} = y_t, \quad (26)$$

where  $y_t$  is the actual value in the current period  $t$  and  $\hat{y}_{t+1}$  is the predicted value in the next period.

We also compare the SVM's forecasting performance with that of linear discriminant analysis (LDA), quadratic discriminant analysis (QDA) and Elman backpropagation neural networks (EBNN). LDA can handle the case in which the within class frequencies are unequal and its performance has been examined on randomly generated test data. This method maximizes the ratio of between-class variance to the within-class variance in any particular data set, thereby guaranteeing maximal separability.

QDA is similar to LDA, only dropping the assumption of equal covariance matrices. Therefore, the boundary between two discrimination regions is allowed to be a quadratic surface (for example, ellipsoid, hyperboloid, etc.) in the maximum likelihood argument with normal distributions. In this chapter, we derive a linear discriminant function of the form:

$$L(S_{t-1}^{s\&p500}, S_{t-1}^{IND}) = a_0 + a_1 S_{t-1}^{s\&p500} + a_2 S_{t-1}^{IND} \quad (27)$$

and a quadratic discriminant function of the form:

$$Q((S_{t-1}^{s\&p500}, S_{t-1}^{IND})^T) = a + P((S_{t-1}^{s\&p500}, S_{t-1}^{IND})^T + ((S_{t-1}^{s\&p500}, S_{t-1}^{IND})^T)((S_{t-1}^{s\&p500}, S_{t-1}^{IND})^T)^T, \quad (28)$$

where  $a_0, a_1, a_2, a, P, T$  are coefficients to be estimated.

Elman Backpropagation Neural Network is a partially recurrent neural network. The connections are mainly feed forward but also include a set of carefully chosen feedback connections that let the network remember cues from the recent past. The input layer is divided into two parts: the true input units and the context units that hold a copy of the activations of the hidden units from the previous time step. Therefore, network activation produced by past inputs can cycle back and affect the processing of future inputs.

### 3.3. A combining model

Given a task that requires expert knowledge to perform,  $k$  experts may be better than one if their individual judgments are appropriately combined. Based on this idea, predictive

performance can be improved by combining various methods. Therefore, we propose a combining model by integrating SVM with other classification methods as follows:

$$f_{\text{combine}} = \sum_{i=1}^k w_i f_i, \quad (29)$$

where  $w_i$  is the weight assigned to classification method  $i$ ,  $\sum_{i=1}^k w_i = 1$ . We would like to determine the weight scheme based on the information from the training phase. Under this strategy, the relative contribution of a forecasting method to the final combined score depends on the in sample forecasting performance of the learned classifier in the training phase. Conceptually, a well-performed forecasting method should be given a larger weight than the others during the score combination. In the investigation, we adopt the weight scheme as follows:

$$w_i = \frac{a_i}{\sum_{i=1}^k a_i}, \quad (30)$$

where  $a_i$  is the in sample performance constructed by forecasting method  $i$ .

**Table 1**

Forecasting performance of different classification methods

Classification method	Hit ratio (%)
RW	50
LDA	55
QDA	69
EBNN	69
SVM	73
Combining model	75

**Table 2**

Covariances matrices of input variables when  $\text{Direction}_t = -1$

	$S_{t-1}^{\text{IND}}$	$S_{t-1}^{\text{S\&p500}}$
$S_{t-1}^{\text{IND}}$	0.00015167706	0.00002147347
$S_{t-1}^{\text{S\&p500}}$	0.00002147347	0.00044862762

#### 4. EXPERIMENT RESULTS

Each of the forecasting models described in the last section is estimated and validated by in sample data. The model estimation selection process is then followed by an empirical evaluation which is based on the out-sample data. At this stage, the relative performance of the models is measured by hit ratio. Table 1 shows the experiment results.

RW performs worst, producing only 50% hit ratio. RW assumes not only that all historic information is summarized in the current value, but also that increments—positive or negative—are uncorrelated (random), and balanced, that is, with an expected value equal to zero. In other words, in the long run there are as many positive as negative fluctuations making long term predictions other than the trend impossible.

SVM has the highest forecasting accuracy among the individual forecasting methods. One reason that SVM performs better than the earlier classification methods is that SVM is designed to minimize the structural risk, whereas the previous techniques are usually based on minimization of empirical risk. In other words, SVM seeks to minimize an upper bound of the generalization error rather than minimizing training error. So SVM is usually less vulnerable to the over fitting problem. QDA out performs LDA in term of hit ratio, because LDA assumes that all the classes have equal covariance matrices, which is not consistent with the properties of input variable belonging to different classes as shown in Tables 2 and 3. In fact, the two classes have different covariance matrices. Heteroscedastic models are more appropriate than homoscedastic models.

The integration of SVM and the other forecasting methods improves the forecasting performance. Different classification methods typically have access to different information and therefore produce different forecasting results. Given this, we can combine the individual forecaster's various information sets to produce a single superior information set from which a single superior forecast could be produced.

**Table 3 :** Covariances matrices of input variables when  $\text{Direction}_t = 1$

	$S_{t-1}^{\text{IND}}$	$S_{t-1}^{\text{S\&p500}}$
$S_{t-1}^{\text{IND}}$	0.00018240800	-0.00002932242
$S_{t-1}^{\text{S\&p500}}$	-0.00002932242	0.00044571885

The method of support vector regression includes several parameters to be chosen, which can e.g. optimized using cross validation.

These parameter include the chosen kernel with parameter  $\gamma$ , the  $e$  of the  $e$ -insensitive loss function, the cost of error  $c$  and the number of training samples. The advantage of using a kernel is sometimes to be able to linearly classify inseparable cases like shown on the top of the figure. In this case, the black and white label points on the left side are not linearly separable. After the kernel transformation, however, the black and white labelled points might fall onto the same point in the new space. Here, the classification problem becomes

trivial. Therefore choosing a kernel is high importance, as well as the parameter  $\gamma$  of the kernel function.

Another parameter is the  $e$  of the insensitive loss function, which is illustrated on the bottom of the figure. The support vector regression model is trained placing a penalty for values, which are off target. The penalty depends on the  $e$ -insensitive loss function, with parameter  $e$ . The idea is to penalize values off target only if the difference is higher than the absolute value of  $e$ .

Given the kernel  $K(x_i, x_j) = \varphi(x_i)^T \varphi(x_j)$ , the training set of instance-label pairs  $(x_i, y_i), i = 1, \dots, l$ , where  $x_i \in \mathbb{R}^n$  and  $y_i \in [1, -1]$ , the optimization problem of the support vector machines can be formulated as  $\min \frac{1}{2} w^T w + c \sum_{i=1}^l E_i$  subject to  $y_i (w^T \varphi(x_i) + b) \leq 1 - E_i, E_i \leq 0$ .

The support vector machine then maximizes the margin of the separating hyperplane of the classes, which is equal to minimizing  $|w|/|t|$  and therefore also to minimizing  $|w|^2 / 2$ .

### Cross-validation

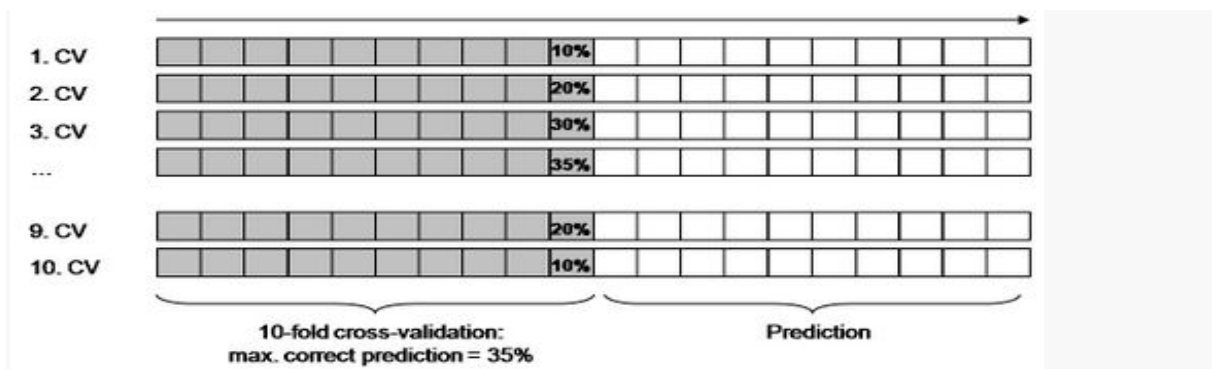


Figure 4: Cross-validation setup. Several parameter values are tested in the prediction accuracy on a training set, of which then the optimal parameter combination is chosen for further prediction on the test set.

Since the SVR parameters can be easily controlled manually, the optimal set of parameters is chosen on a test set and then used on the following training set. The cross-validation is applied as illustrated in the figure. The total data set is divided into two parts, one for cross-validation and one for testing. A third part of the data set in order to optimize the structure of the model, like the used indicators, is omitted in this study.

In order to optimize the number of training samples, the cost of error  $c$ , the kernel parameter and the parameter  $e$  of the  $e$ -insensitive loss-function, a  $k$ -fold cross-validation is used as follows: the dataset is divided into  $k$  folders of equal size; subsequently, a model is built on all possible ( $k$ ) combinations of  $k-1$  folders, and each time the remaining one folder is used for validation. The best model is the one that performs best on average over the  $k$  validation

folders. The benefit of using a cross-validation procedure is that by construction it ensures that model selection is based entirely on out-of-sample rather than in-sample performance. Thus, the search for the best Support Vector Regression model is immune to a critique of drawing conclusions about the merits of a factor model based on its in-sample performance.

In this study, a 10-fold cross-validation procedure was used for each parameter above. In each validation loop, different values for each parameter are chosen, while the other parameters are set constant. Then the SVR model is trained with this set of parameters and the prediction accuracy is calculated. This is done for all parameter combinations and then the combination with the maximal prediction accuracy chosen.

### Basic model

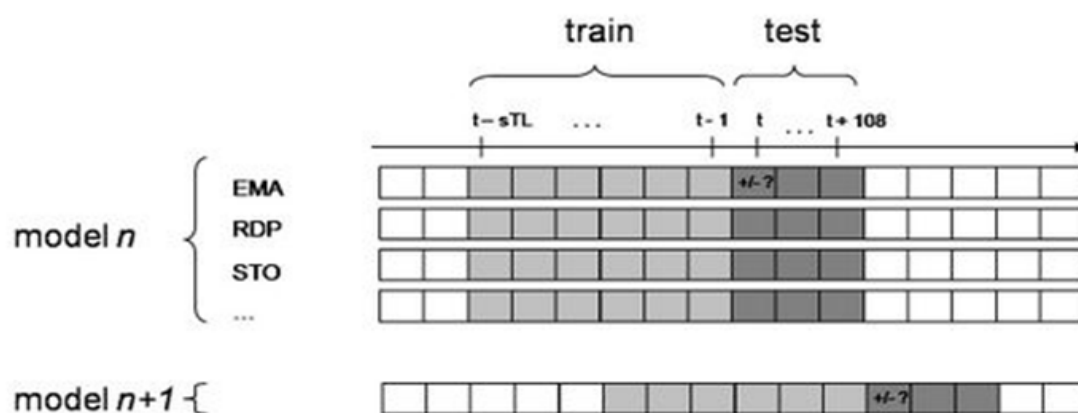


Figure 5: The basic model. The machine is trained on the past values of the indicators. The resulting model is used to predict the movement on the next day (= 108 data points). After that the model is shifted and proceeds again.

The basic simulation consists of two steps: First, at month  $t$ , all historical values for all explanatory factors together with the difference in returns for the period's  $t - n1$  till  $t - 1$  are used to build numerous support vector regressions. Thus the dependent variable is the return of the stock in the period of  $t$  till  $t + n2$ . The variable  $n2$  is arbitrarily chosen to 108, in order to decrease calculation time. The independent variables are the technical indicators as described above.

Second, once the prediction is calculated, the model is shifted 108 data points and the model is build again in order to predict the next 108 stock price movements.

Using only historically available data ensures the implementation of the trading strategy is carried out without the benefit of foresight, in the sense that investment decisions are not based on data that have become available after any of the to-be-predicted periods. Moreover,



investment decisions for the to-be-predicted months are always based on the entire factor set of historical data, ensuring that no variable-selection procedures based on extensive manipulation of the whole available data have been carried out. At any rate, the utilized cross-validation procedure for model selection ensures that the best candidate model is selected on the basis of performance in the training set and not on the basis of performance on external validation samples.

## RESULTS AND DISCUSSION

The data set consists of 5 minute closing prices  $p(t)$  for 28 stocks in the BSE Sensex. The missing stocks are Satyam and Hypo Real Estate due to data unavailability. With a time frame of nearly 7 years between April 2004 and August 2010, the data set comprises 140.000 data points per stock.

From this data set, the log return of each stock  $i$  is calculated as

$$x_i(t) = \log\left(\frac{p(t)}{p(t-1)}\right)$$

with price  $p(t)$  at time  $t$  as well as the market average as

$$x_{market}(t) = \sum_{i=1}^{28} x_i(t)$$

over all stocks  $i$ . From this the log return above market is calculated as

$$x_i'(t) = x_i(t) - x_{market}(t) \text{ for each stock } i.$$

### Cross-validation

Several parameter values are chosen for each of the machine parameters. The cost and training length parameter show linear dependencies, while the kernel parameter gamma shows a quadratic dependency. The  $e$  parameter is rather nonlinear dependent to the prediction accuracy. Several parameter conditions were tested on the first half of the data set. The figure shows the tested values for each parameter. The optimality criterion used here, is the cumulated return. Therefore, the model is trained with the parameter set, the prediction calculated and then the return resulting from the prediction is cumulated over time. The parameter values are tested on half of the data set, that is between May 2009 and July 2011.

On the top left of the figure, the results for different parameter values of the cost function is shown. With an increasing cost function value, the cumulated return increases. This seems plausible, since with an increasing cost the model is trained longer. However, the parameter exploration is stopped at a cost value of 1000, since higher values increase computation time dramatically.



The top right of the figure shows different parameter values for  $\epsilon$  of the  $\epsilon$ -insensitive loss-function. Here the results seem to be rather nonlinearly related to the cumulated return, since with increasing parameter  $\epsilon$ , the cumulated return decreases only in general. However, generally, smaller values of  $\epsilon$  seem to be more successful. Since this value controls the penalty of the training algorithm, a small value indicates a fast penalty for values off-target.

The kernel parameter  $\gamma$ , plotted for different values on the bottom left, seems to approach an optimum value around 1. The parameter controls the shape of the kernel. With high parameter values, the kernel becomes rather flat and the model increasingly predicts future movements only linearly, which is obviously insufficient. With small parameter values, the kernel becomes very thin and training data are increasingly over fitted with decreasing generalization performance. This again results in a low prediction performance. Last, with an increasing number of training points the prediction performance increases. Therefore the quality of the trained model increases with the number of training samples.

### **Prediction accuracy**

The optimized parameters were tested with the basic model approach described above on the second half of the data set. The prediction accuracy over all 28 stocks reached a mean of 50.69 percent with standard deviation of .26%. With this performance, the reported approach significantly outperformed a random prediction approach. Even if a gain of .69 percent might be a valuable trading prediction, this approach is market neutral and operated only on the basic statistical properties of market movements.

## **5. CONCLUSIONS**

In this Chapter, we study the use of support vector machines to predict financial movement direction. SVM is a promising type of tool for financial forecasting. As demonstrated in our empirical analysis, SVM is superior to the other individual classification methods in forecasting weekly movement direction of BSE30 Index. This is a clear message for financial forecasters and traders, which can lead to a capital gain. However, each method has its own strengths and weaknesses. Thus, we propose a combining model by integrating SVM with other classification methods. The weakness of one method can be balanced by the strengths of another by achieving a systematic effect. The combining model performs best among all the forecasting methods.

The underlying time series were derived from the Bombay Stock Exchange Index. The support vector machine was then trained in order to predict the movement of 28 stocks of the

index against market. Features for training were directly extracted from the statistical properties of the time series and no fundamental information was used.

The model selection was based on the performance on out-of-sample data, in order to avoid critique of foresight and was performed as cross-validation. The main result of this study is that the movement of stocks is significantly predicted only using technical indicators with support vector regression.

## 6. REFERENCES

- [1] Cristianini N, Taylor JS. An introduction to support vector machines and other kernel-based learning methods. New York: Cambridge University Press; 2000.
- [2] Cao LJ, Tay FEH. Financial forecasting using support vector machines. *Neural Computing Applications* 2001;10: 184–92.
- [3] Tay FEH, Cao LJ. Application of support vector machines in financial time series forecasting. *Omega* 2001;29:309–17.
- [4] Castanias R.P. Macro information and the Variability of Stock Market Prices. *Journal of Finance* 34 (1979), pp.439–
- [5] Schwert G William. The Adjustment of Stock Prices to Information about Inflation. *The Journal of Finance* 36 (1981), pp.15-29.
- [6] Schwert G William. Stock Returns and Real Activity: A Century of Evidence. *Journal of Finance* 14 (1990), pp.1237-1257.
- [7] Fama EF. Stock Returns, Real Activity, Inflation and Money. *American Economic Review* 71 (1981), pp.71:545–
- [8] Nai-Fu Chen, Roll R, Ross R. Economic Forces and The Stock Market. *Journal of Business* 59 (1986), pp.383-403.
- [9] Hardouvelis Gikas A. Macroeconomic Information and Stock Prices. *Journal of Economics and Business* 1987;39:131-140.
- [10] Darrat AF. Stock Returns, Money and Fiscal Deficits. *Journal of Financial and Quantitative Analysis* 25 (1990), pp.387–98.
- [11] Blank SC. Chaos in futures market? a nonlinear dynamical analysis. *Journal of Futures Markets* 1991;11:711–28.
- [12] DeCoster GP, Labys WC, Mitchell DW. Evidence of chaos in commodity futures prices. *Journal of Futures Markets* 1992;12:291–305.
- [13] Frank M, Stengos T. Measuring the strangeness of gold and silver rates of return. *The Review of Economic Studies* 1989;56:553–67.

[14] Frank M, Stengos T. Measuring the strangeness of gold and silver rates of return. *The Review of Economic Studies* 1989;56:553–67.

[15] Vapnik VN. *Statistical learning theory*. New York: Wiley; 1998.

[16] Vapnik VN. An overview of statistical learning theory. *IEEE Transactions of Neural Networks* 1999;10:988–99.