Journal of Computational Science 17 (2016) 1–13

Contents lists available at ScienceDirect

Journal of Computational Science

journal homepage: www.elsevier.com/locate/jocs

# Proximal support vector machine based hybrid prediction models for trend forecasting in financial markets

Deepak Kumar, Suraj S. Meghwani, Manoj Thakur*

*Indian Institute of Technology-Mandi, Mandi 175001, Himachal Pradesh, India*

ARTICLE INFO

*Article history:*
Received 10 October 2015
Received in revised form 4 July 2016
Accepted 19 July 2016
Available online 30 July 2016

*Keywords:*
Proximal support vector machines
Random forest
Feature selection
Stock index trend prediction
RReliefF technical indicators

ABSTRACT

In the recent years, various financial forecasting systems have been developed using machine learning techniques. Deciding the relevant input variables for these systems is a crucial factor and their performances depend a lot on the choice of input variables. In this work, a set of fifty-five technical indicators has been considered based on their application in technical analysis as input feature to predict the future (one-day-ahead) direction of stock indices. This study proposes four hybrid prediction models that are combinations of four different feature selection techniques (Linear Correlation (LC), Rank Correlation (RC), Regression Relief (RR) and Random Forest (RF)), with proximal support vector machine (PSVM) classifier. The performance of these models has been evaluated for twelve different stock indices, on the basis of several performance metrics used in literature. A new performance measuring criteria, called joint prediction error (JPE) is also proposed for comparing the results. The empirical results obtained over a set of stock market indices from different international markets show that all hybrid models perform better than the individual PSVM prediction model. The comparison between the proposed models demonstrates superiority of RF-PSVM over all other prediction models. Empirical findings also suggest the superiority of a certain set of indicators over other indicators in achieving better results.



## 1. Introduction

Financial markets are driven by various factors, viz. government policies, economic conditions, political issues, trader's expectations, etc. Many factors which are either unidentified or incomprehensible makes the prediction of stock prices a challenging task. Several empirical studies [1–3] have suggested that index prices follow nonlinear dynamic behavior instead of complete random behavior. This further indicates that trends in the stock markets can only be predicted up to a certain limits [1].

In the last decade, several machine learning algorithms have been developed and used for stock market trend predictions. Among these algorithms, artificial neural networks (ANNs) and support vector machines (SVMs) [4] are the extensively used algorithms. ANNs are non-parametric, empirical risk minimization based modelling approaches with the capability of approximating any non-linear function up to arbitrary precision, regardless of any prior assumptions about data and input space [5,6]. ANNs have been employed for financial trend prediction in [7–9]. Though, ANNs have been applied to a wide-range of real life problems as learning paradigms, but they are found to have several limitations like non-convex objective function and difficulty in deciding criterion for the number of hidden layers. In addition ANNs also suffer from the over-fitting problem, which arise due to large number of parameters in the model and is a major drawback of empirical risk minimization principle. In financial time series forecasting, Kim and Co [10,11] reported inconsistency in the performance of ANN, arising due to over-fitting problem.

Over-fitting is a major drawback of the empirical risk minimization principle. Due do this drawback of empirical risk minimization based approach, the research in the recent past has been diverted towards another approach called structural risk minimization (SRM) principle which was proposed by Vapnik [12]. Vapnik [13,14] formulated the SRM based classification as an optimization problem which tries to minimize the upper limit of the expected error and found it to be superior than empirical risk minimization. Support vector machine, developed by Cortes and Vapnik [4] (SVM) is a SRM based technique. Mathematically, SVMs are formulated as a quadratic optimization problem which ensures global optimal solution and are found to be better than ANNs [15] as far as the over-fitting problem is concerned. SVMs have been successfully applied for trend predication in financial markets. Several empirical studies [11,10,16] have shown that SVMs provide a promising alternative to ANNs.

* Corresponding author.
*E-mail address:* manojpma@gmail.com (M. Thakur).

http://dx.doi.org/10.1016/j.jocs.2016.07.006
1877-7503/

Deciding input variables or features for trend prediction in the stock market is a challenging task. In literature [17–20], factors like historical price patterns, macroeconomic variables and technical indicators have been applied as inputs to the trend prediction models. Though, none of the feature set is expected to exactly replicate market dynamics which depends upon many other undefined and incomprehensible factors. Nevertheless, deriving features from available quantitative information in market is found to be a promising strategy in trend prediction.

Recently, use of feature selection techniques in wrapper and filter methods is used to improve the computational complexity and interpretability of the models [21,22]. Various feature selection techniques have been employed in combination with SVM for financial forecasting in [23–25,19]. All of the above studies have concluded that two-stage architecture in financial forecasting achieves higher performance as compared to the individual SVMs or other learning paradigms. However, improving the effectiveness and efficiency of financial forecasting still remains the major challenge for the researchers in this area.

This study proposes and compares several two-stage architectures that can predict the next-day direction of the stock markets. These models combine various feature selection methods with Proximal Support Vector Machine (PSVM) [26] as a classifier for twelve stock indices. Four feature selection techniques, viz. Linear correlation (LC), rank correlation (RC), regression relief (RR) and random forest (RF) are used in conjunction with PSVM. Proposed methods are named as RF-PSVM, RR-PSVM, LC-PSVM and RC-PSVM.

PSVM formulation for classification problem can be interpreted as a special case of regularized least squares, which leads to a strongly convex objective function. As compared to other variants of SVMs, e.g., LS-SVM [27] and SVM-Light [28], PSVM is much faster. PSVM classifies data points by allotting the close of two parallel planes that are pushed aside as far as possible. Performance of proposed hybrid models is tested over twelve stock indices on the basis of several performance metrics like precision, recall, testing accuracy, $F_1$ score along with a new proposed metric called joint prediction error (JPE). JPE encapsulates the combined effect of $F_1$ score of stock rise and fall. The empirical study shows that PSVM has better performance in predicting stock rise and stock fall. Based on JPE and other performance metrics the performance of all hybrid models are found to be better than ALL-PSVM (PSVM with all features) and ALL-BPNN (BPNN with all features). Moreover, RF-PSVM outperforms all other hybrid models in nine out of twelve stock indices. This study is limited for predicting index trend, although this work can also be used for individual stocks. Here, our proposed software tool is applied to generate buy and sell signals based on the combination of optimal number of technical indicators.

The rest of this paper is organized as follows. Section 2 gives a brief introduction of feature selection techniques and PSVM used in the current study. Proposed hybrid models are discussed in Section 3. The details about the data used in the current work is reported in Section 4. Performance measures & implementation of prediction models is presented in Section 5. Section 6 reports empirical findings in this study. Conclusion from the current study is drawn in Section 7.

## 2. Methodology

### 2.1. Feature selection

A selection of important features enhances interpretability of the model. It can improve the performance of learning algorithms and also helps in reducing the computational complexity of the model. In this section, various feature selection techniques used in this study for trend prediction of stock indices are briefly discussed.

#### 2.1.1. Correlation criteria

One of the simplest means to draw out relevant features is by using correlation coefficient [29]. The importance score or measure of importance for feature $f$ can be assessed by value of correlation coefficient with an output variable, say $y$. Thus, importance score ($S_f^{LC}$) for feature $f$ is calculated by Pearson's coefficient:

$$S_f^{LC} = \left( \frac{cov(f, y)}{\sqrt{var(f) var(y)}} \right)^2 \tag{1}$$

where $cov(.,.)$ and $var(.)$ denote the covariance and variance respectively. A slightly different correlation criterion for feature selection is Spearman's rank correlation, which does not depend upon the distribution of participating variables as opposed to normality assumption in Pearson's correlation coefficient. Spearman correlation is defined as Pearson correlation of ranks of the variables. Importance scores $S_f^{RC}$ for feature $f$ using Spearman coefficient criterion can be calculated as

$$S_f^{RC} = \left( 1 - \frac{6 \times \sum d_i^2}{n \times (n^2 - 1)} \right) \tag{2}$$

Here $d_i$ is a difference in ranks of feature $f$ and output variable $y$.

#### 2.1.2. Regression Reflief feature selection

From the nearest neighbor algorithm, Relief algorithm was proposed by Kira et al. [30]. Relief algorithm and its extensions are mostly used for the feature pruning task in many regression and classification algorithms [31,32]. In this algorithm, importance score of all features are first initialized with zero. Then, for a randomly selected instance ($X_i$) from training instances, Relief algorithm searches for its $k$ proximal instances of same and opposite class which are denoted by $N_i^+$ and $N_i^-$ respectively. Let $N_i^+[f], N_i^-[f]$, and $X_i[f]$ represent the numerical value of $f$th feature in respective instances, then importance score of feature $f$th, denoted by ($S_f^{RR}$), is updated by the following rule

$$S_f^{RR} = S_f^{RR} + \sum_k \frac{d(X_i[f], N_i^-[f]) - d(X_i[f], N_i^+[f])}{k} \tag{3}$$

where

$$d(X_i[f], N_i^-[f]) = \frac{|X_i[f] - N_i^-[f]|}{\max(f) - \min(f)}, \quad d(X_i[f], N_i^+[f]) = \frac{|X_i[f] - N_i^+[f]|}{\max(f) - \min(f)}$$

$max(f)$ and $min(f)$ are the maximum and minimum value of feature $f$ in training instances. This procedure is repeated for $m$ randomly selected instances from training instances. One of the extension of relief algorithm is Regression Relief Feature (RReliefF) selection (proposed in [33]). Since, target variable in regression problem is continuous, concept of proximal classes instance is not possible. Here, proximity is modelled as the relative distance between predicted values of two instances.

#### 2.1.3. Random forest

Random forest [34] is one of the most popular classification and regression algorithm. It has many advantageous characteristics like better generalization capability, robustness, feature pruning ability and simplicity to do non-linear classification. The principle behind the random forest algorithm is the construction of many unpruned decision trees with each tree using bootstrapped training data. In random forest rather than determining the best split among all the features, only a subset of randomly selected features is considered.

Let, $m$ denotes number of training examples with $n$ features. Firstly, $m$ examples are sampled with replacement for each $k$ decision tree. During the process of decision tree construction, best split is decided among $mtry << n$ features selected at random from $n$ features. The final regression and classification output is given by aggregating the results from all $k$ decision trees. Randomness in deciding best split and construction of different trees with different bootstrapping samples are important factors for better generalization ability of random forest. Only two-third of randomly split bootstrapped data is used while training and the remaining data, called Out-of-a Bag (OOB) samples, is used to get an unbiased prediction accuracy of the constructed tree and is also used to quantify feature importance scores. All decision trees $T_i \quad \forall i = 1, 2, \ldots, k$ are tested with their respective OOB samples and their error rate (misclassification for classification problem and Mean Squared error for regression problem) $E_{T_i}$ is recorded. To compute the importance score of feature $f$, perturbed OOB samples are produced for each tree. This is done by randomly permuting the feature among the samples. Again, error rate $E'_{T_i}$ for the perturbed OOB samples is calculated for each tree. Then, the feature importance score $S_f^{RF}$ is calculated by the following formula

$$S_f^{RF} = \frac{1}{k}\sum_{T_i}(E_{T_i} - E'_{T_i}) \tag{4}$$

A detailed discussion of variable importance using random forest could be found in [35,36,34].

### 2.2. Classification methods

This section gives the brief description of classification approaches used in this study.

#### 2.2.1. Support vector machine

Support vector machines (SVM) [4] is very powerful tool for binary classification and has strong theoretical foundation. Consider a binary classification problem for classifying $m$ data points in the $n$ dimensional real space $R^n$, represented by matrix $A$ (data point $x_i$ is the $ith$ row of $A$). The class of each point $x_i$ can be denoted by $y_i \in \{1, -1\}$. A linear SVM searches for optimal decision hyperplane defined as:

$$x.w + b = 0, \quad w \in R^n, \quad b \in R \tag{5}$$

where $w$ is normal to the optimal hyperplane (5), termed as *weight vector* and $b$ is known as bias. The optimal hyperplane can be obtained by minimizing $||w||$. An equivalent mathematical formulation of the same can be given by the following quadratic program:

$$\begin{aligned} &\textit{minimize} \quad \frac{1}{2}||w||^2 + \nu\sum_{i=1}^{m}\xi_i \\ &\textit{subject to}: \\ &\qquad y_i(x_i.w + b) \ge 1 - \xi_i, \quad \xi_i \ge 0, \quad \forall i \end{aligned} \tag{6}$$

where $\nu$ is the regularization parameter and $\xi_i$ are non-negative slack variables. Problem (6) is a Quadratic programming (QP) optimization problem with linear constraints and can be solved by using standard QP solver.

#### 2.2.2. Proximal support vector machine

Fung and Mangasarian [26] proposed proximal support vector machine (PSVM) in more general context of regularization network. PSVM classifies the data points by the proximity to one of the two planes which are having the maximum possible margin. The standard SVMs formulation is modified in two ways. First, the margin between the bounding planes is maximized with respect to both $w$ and $b$ instead of just $w$. This leads to model of classification problem to be formulated in $(n+1)$ dimensional space $(R^{n+1})$ instead of $R^n$. Second, inequality constraints are replaced by equality constraints as follows. The mathematical model is given as follows:

$$\begin{aligned} &\textit{minimize} \quad \frac{1}{2}(w'w + b^2) + \frac{\nu}{2}\sum_{i=1}^{m}{\xi_i}^2 \\ &\textit{subject to}: \\ &\qquad y_i(x_i.w + b) = 1 - \xi_i, \end{aligned} \tag{7}$$

Here, planes $x_i.w + b \ge \pm 1$ are called proximal planes and the data points of each class are clustered around them. The 2-norm squared distance between these planes is the reciprocal of the first term in the object function $(w'w + b^2)$. Problem (7) is strongly convex quadratic optimization problem with linear equality constraints and can be solved analytically, whereas an exact closed form solution is not possible in standard SVM. Lagrangian formulation of the problem (7) is given as follows:

$$\begin{aligned} L(w, b, \xi, \lambda) = &\frac{1}{2}(w'w + b^2) + \frac{\nu}{2}\sum_{i=1}^{m}{\xi_i}^2 \\ &- \sum_{i=1}^{m}\lambda_i(y_i(x_i.w + b) - 1 + \xi_i) \end{aligned} \tag{8}$$

Here, $\lambda_i \in R$, $i = 1, 2, \ldots, m$ are Lagrange multipliers for each equality constraints (7). Setting the gradients of the Lagrangian (8) with respect to $(w, b, \xi, \lambda)$ equal to zero gives the following expression:

$$\begin{aligned} &w = \sum_{i=1}^{m}x_iy_i\lambda_i, \quad b = \sum_{i=1}^{m}y_i\lambda_i, \quad \xi_i = \frac{\lambda_i}{\nu}. \\ &y_i(x_i.w + b) - 1 + \xi_i = 0 \end{aligned} \tag{9}$$

Substituting these expressions of $w$, $b$ and $\xi_i$ in the equations of system (9) and solving the expression for $\lambda$ involves inversion of $m \times m$ matrix. Inversion of this massive matrix can be done by using Sherman–Morrison–Woodbury formulation [37], which involve the inversion of a smaller dimension matrix of order $(n+1) \times (n+1)$. The value of $\lambda$ is substituted back in the first two expressions of (9) and the solution for $w$ and $b$ is obtained as follows:

$$[\mathrm{w\,b}] = \left(\frac{I}{\nu} + M'M\right)^{-1} M'Ye \tag{10}$$

where $M = [A \quad e]$, $A$ is a matrix of order $m \times n$ of the training data, and $Y$ is a diagonal matrix with $y_i$ along its diagonal. After obtaining $w$ and $b$, the test data point is classified using the optimal decision surface given by:

$$f(x) = sign(w^Tx + b).$$

If the value of $f(x)$ is positive, the test data point is assigned to class 1, otherwise it is assigned to class −1.

## 3. Proposed prediction models

Four hybrid models for trend prediction in financial markets are proposed. In these models, four feature selection techniques – RF, RR, LC and RC are combined with PSVM and respective models are denoted by RF-PSVM, RR-PSVM, LC-PSVM and RC-PSVM respectively.

In developing a prediction model, the first important step is determining the input variables. Irrelevant variables or correlated variables could deteriorate the performance of classifier. On the other hand, large number of input variables lead to increase in the computational cost and risk of over-fitting as well as noise.

Eq. (10) shows the dependency of PSVM solution on $M^TM$, where matrix $M$ contains information about the features. The regularization term in the solution helps in reducing condition number but at the same time it provides an approximate solution. Therefore, for better approximation, it becomes utmost important that condition number of $M^TM$ should be handled individually. Higher condition number indicates linear dependency of features or existence of high correlation among the features. Hence, incorporating large feature space does not always lead to better prediction and stable results. It is important to analyze the feature space before entering in to actual task of classification. The main goal of incorporating the feature selection techniques is to transform the feature space from higher dimension to low dimension by removing redundant and highly correlated variables from feature space.

Technical analysts use different set of technical indicators for generating buy and sell signals. Menkhoff [38] surveyed 692 fund managers from different countries and reported their reliance on technical analysis. The quality of prediction depends upon how the trading rules have been defined using the potential technical indicators. These trading rules and indicator collection needs to be redefine frequently as their performance change with change in market dynamics. Therefore, selection of technical indicators for any stock index is a challenging task. Further, exhaustive search over all possible feature subsets and checking the discriminative ability of classifier is a combinatorial problem and may be computationally expensive even for a moderate size of indicators. Feature selection can address this problem by determining the optimal subset of indicators. The computation cost of SVM is $O(m^3)$ and that of PSVM is $O(n^3)$, where $m$ is the number of training samples and $n$ is the number of features variables. In this study, the number of features is quite less as compared to the number of training samples (*i.e.* $n \ll m$), which implies that PSVM is much faster than SVM. In addition, if $n$ is decreased, the computation cost of PSVM will decrease significantly.

The motivation behind using different feature selection techniques lies in their underlying methodology which is different from each other. Linear correlation is simple but effective criteria for deciding the feature discovery when relation between feature and response variable is linear and sometimes monotonic. Rank correlation as compared to linear correlation does not depend on the distribution of participating variables and is free from normality assumption as in Pearson correlation. But two main problems with correlation criteria are:

1. Assuming that no transformation of variables are done, it is unable to tackle non-linear relation between the response variable and features.
2. Since the ranking of features is done independently (without considering cross-correlation among each other). There is a possibility of selection of redundant variables which is not desirable while constructing a parsimonious model.

RRelief is inspired from the nearest neighbor technique for segregating relevant and irrelevant features. The major limitation of RRelief is that it does not detect redundant feature [39]. Random Forest is a tree based ensemble technique for calculating relative importance of features. Using the strong law of large numbers Leo Breiman [34] [Theorem 1.2] showed that random forest does not over fit with the number of trees. Genuer et al. [40], further investigated the sensitivity of variable importance with number of trees in the forest (*ntree*). Their empirical study suggests that the effect of increasing tree is less on variable importance, instead, large *ntree* values leads to better stability of variable importance. Hence, theoretically one can assert that RF among other considered techniques should provide the better representation of feature space.

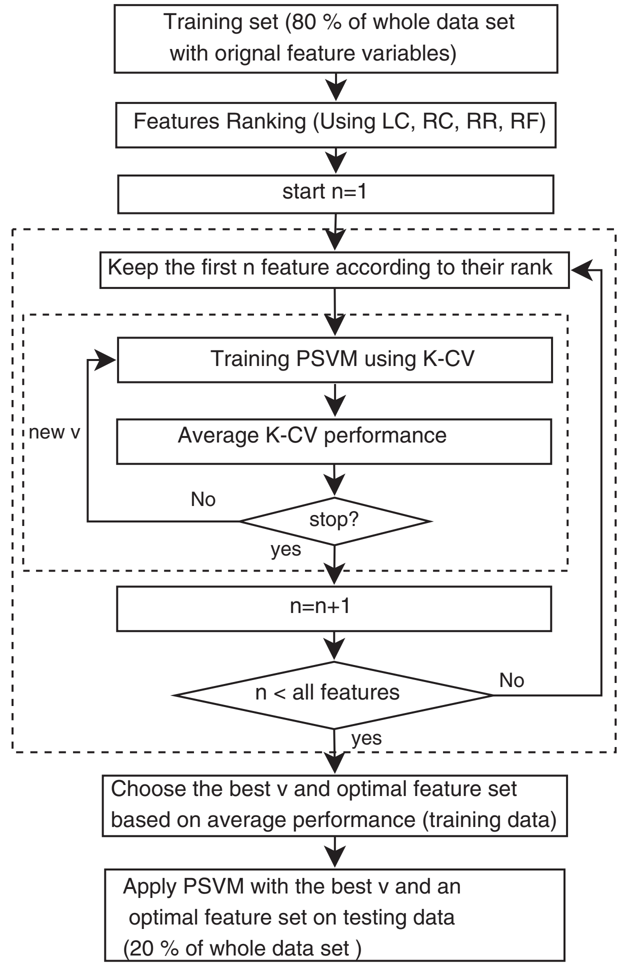


**Fig. 1.** The procedure of the proposed hybrid prediction models.

Fig. 1 illustrates the flowchart of the proposed hybrid prediction model. The data corresponding to each index is divided into two parts. For each index eighty percent of the whole data, called *training set*, is used for training the classifier and feature pruning task. The remaining twenty percent data, called *testing data*, is used for testing the performance of proposed hybrid models. Training data of each index is given as input to a feature selection technique to rank all the features according to their importance. Then, these features are added iteratively (one by one) to PSVM model according to their ranks i.e., a feature with more importance or higher rank is added before the one with comparatively less importance or lower rank. In next step, PSVM is trained on training set. During training phase, regularization parameter $\nu$ of PSVM model plays a crucial role on the performance of the model. Smaller values of $\nu$ imparts better generalization ability but poor classification accuracy. On the other hand, larger values of $\nu$ improves the classification accuracy but affects its generalization ability. Hence, optimal value of $\nu$ is based on trade-off between generalization ability and prediction accuracy. In order to determine the optimal value of $\nu$, *K*-fold Cross Validation (*K*-CV) method [41], which have been extensively used in conjunction with SVM for financial time series data [24,23,25,19] is used. In *K*-CV validation, training set is divided into *K* sequential blocks of same size. The training of PSVM is performed on (*K*-1) blocks of data and the remaining one block is used as validation data. This process is repeated until all *K* blocks are covered as validation data and their average validation accuracy is computed. *K*-CV is performed for all specified values of $\nu$ and its optimal value is chosen for which the average validation accuracy is maximum. Once the optimal regularization parameter is decided, PSVM is trained on whole training set and its training results are recorded. This training procedure is performed each time when a new feature is added to the PSVM model. Finally, the feature subset which reports the maximum training accuracy is chosen as the optimal feature set for the PSVM model.

Same procedure is followed for all four feature selection techniques and all stock indices considered in this study. Finally, testing data is used to evaluate the performance and robustness of the hybrid prediction model.

## 4. Experimental data description

To perform comparative study of models, various experiments have been performed on twelve major stock indices from different parts of the world. Historical OHLC (Open, High, Low and Close) data of all twelve indices (listed in Table 1) considered in this study is collected from Yahoo Finance for the period January 2008 to December 2013. The data sets are partitioned into two parts – training set (from January 2008 to December 2012) and testing set(from January 2013 to December 2013). 55 technical indicators and oscillators have been computed as input features. Some of the technical indicators are selected from previous studies done in [11,24] while rest are collected on the basis of their popularity and application in technical analysis. Description of technical indicators along with their mathematical formulations are summarized in Table 2.

**Table 1**
Market indices.

| Symbol | Index name | Country |
|---|---|---|
| BSESN | S&P BSE SENSEX | India |
| GDAXI | DAX | Germany |
| HSI | Hang Seng | Hong Kong |
| JKSE | Jakarta Composite | Jakarta |
| KLSE | KLSE Composite | Korea |
| N100 | EURONEXT 100 | Europe |
| NSEI | CNX NIFTY | India |
| N225 | Nikkei 225 | Japan |
| NYA | NYA Composite | USA |
| RUA | Russell 3000 | USA |
| STI | Straits Times | Singapore |
| TWII | Taiwan Weighted | Taiwan |

Due to their own inherent property, different indicators are effective under the different market scenarios. Basic financial time series data (OHLC) are considered as input variables and are also used to compute other indicators. Moving averages (MA, EMA) are popular tools for smoothing out the price information, and reflect the potential change in the price. Whereas, RDP transforms the price into more symmetrical and closer to normal distribution.

Indicators described above are preferable for trending market (bullish or bearish). They are not much more informative in non-trending market scenarios. Usefully oscillators are more effective for extracting price information in such condition, e.g., Stochastic

**Table 2**
Indicators description and their inputs.

| Indicator name | Description | Formulation | Inputs |
|---|---|---|---|
| $OP$ | Open price | $OP_t$ is the open price at time $t$ | – |
| $HI$ | High price | $HI_t$ is the high price at time $t$ | – |
| $LO$ | Low price | $L0_t$ is the low price at time $t$ | – |
| $CL$ | Closing price | $CL_t$ is the closing price at time $t$ | – |
| $MA(x)$ | $x$-days moving average | $MA_t = \frac{1}{x}\sum_{i=1}^{x} CL_{t-i}$ | $x = \{5, 10, 15, 20\}$ |
| $EMA(x)$ | $x$-days exponential moving average | $EMA_t = \alpha(CL_t - EMA_{t-1}) + EMA_{t-1}$ where, $\alpha = \frac{2}{x+1}$ | $x = \{5, 10, 15, 20\}$ |
| $RDP(x)$ | Relative difference in percentage | $RDP_t = \frac{CL_t - CL_{t-x}}{CL_{t-x}} \times 100$ | $x = \{5, 10, 15, 20\}$ |
| $\%K(x)$ | Stochastic $\%K$ | $\%K = \frac{CL_t - LL_{t-x}}{HH_{t-x} - LL_{t-x}}$ | $x = 5$ |
| $\%D(x)$ | $\%D$ is the moving average of $\%K$ | $\%D = \frac{\sum_{i=0}^{i=x-1} \%K_{t-i}}{x}$ | $x = 5$ |
| $\%R(x)$ | Larry William's %R | $\%R_t = \frac{HH_{t-x} - CL_t}{HI_{t-x} - LL_{t-x}}$ | $x = \{5, 10, 15, 20\}$ |
| $BIAS(x)$ | $x$-days Bias | $BIAS_t = \frac{CL_t - MA(x)}{x}$ | $x = \{5, 10, 15, 20\}$ |
| $MACD(x, y)$ | $x$ days moving average convergence and divergence | $MACD_t = (EMA(x) - EMA(y))$ | $(x, y) = \{(5, 10), (5, 15), (5, 20), (10, 15), (10, 20), (15, 20)\}$ |
| $MTM(x)$ | Momentum measures change in stock price over last $x$ days | $MTM_t = CL_t - CL_{t-x}$ | $x = \{5, 10, 15, 20\}$ |
| $ROC(x)$ | Price rate of change | $ROC_t = \frac{CL_t}{CLt-x} \times 100$ | $x = \{5, 10, 15, 20\}$ |
| $OSCP(x, y)$ | Price oscillator | $OSCP_t = \frac{MA(x) - MA(y)}{MA(x)}$ | $(x, y) = \{(5, 10), (5, 15), (5, 20), (10, 15), (10, 20), (15, 20)\}$ |
| $MP(x)$ | Median price | $MP(x) = \frac{1}{2}\left(LL_{(t-x)} + HH_{(t-x)}\right.$ | $x = 10$ |
| $LL(x)$ | Lowest price | $LL(x) = min(LO_{(t-x)})$ | $x = 10$ |
| $HH(x)$ | Highest price | $HH(x) = max(HI_{(t-x)})$ | $x = 10$ |
| $CCI(x)$ | Commodity channel index | $CCI_t = \frac{CL_t - MA(x)}{0.015\sigma}$ | $x = 10$ |
| $SignalLine(x, y)$ | A signal line is also known as a trigger line | $signalLine(x, y) = \frac{1}{10*SMA(y)}(SMA(x) - SMA(y)) + SMA(y)$ | $(x, y) = (10, 20)$ |
| $ATR_t(x)$ | Average true range | $ATR_t(x) = \frac{1}{x}(ATR_{(t-1)} \times (x - 1) + TR_t)$ where $TR_t = \max[(H_t - L_t), abs(H_t - C_{(t-1)}), abs\left(L_t - C_{(t-1)}\right]$ | $x = 10$ |
| $UO(x, y, z)$ | Ultimate oscillator | $UO = 100 \times \frac{1}{4+2+1}(4 * avg(x) + 2 * avg(y) + avg(z))$ where $avg(t) = \frac{bp_1 + bp_2 + \cdots + bp_t}{tr_1 + tr_2 + \cdots + tr_t}$, $tr_t = \max\left(H_t, CL_{(t-1)} - \min(LO_t - CL_{(t-1)}\right)$, $bp_t = CL_t - min(LO_t - CL_{(t-1)})$ | $(x, y, z) = (10, 20, 30)$ |
| $Ulcer(x)$ | Ulcer index | $Ulcer(x) = \sqrt{(R_1^1 + R_2^2 + \cdots + R_n^2)}$ where $R_t(x) = \frac{100}{HH_{(t-x)}} \times \left(CL_t - HH_{(t-x)}\right.$ | $x = 10$ |
| $TSI(x)$ | True strength index | $\text{TSI(x)} = 100 \times \frac{(EMA(EMA(MTM(1),x),x)}{(EMA(EMA(\lvert MTM(1)\rvert,x),x)}$ | $x = 10$ |

$LL_{t-x}$ and $HH_{t-x}$ are the lowest low and highest high in last $x$ trading days respectively. For different values of input parameter $x$, an indicator generates different features.

%K, %D (moving average of %K), %R, BAIS, RSI, CCI, MTM, ROC, TSI and OSCP. Further, they are also used for identifying the oversold or overbought market and divergences. MACD is the difference of two EMAs and indicates trend-following signals that are effective, in momentum-driven scenario. Compared to MACD, UO, which captures the momentum across three different time intervals, provides more reliable momentum of market trend. However, UO is not recommended to be used solely and is usually used in combination with MACD indicator.

MP, LL and HH provide the average price, support and resistance levels respectively. The basic idea behind these indicators is that in a downward-trending market, prices tend to close near their low, and during an upward-trending market, index tend to close near their high. ATR is an indicator which is not meant for directional information, but measures the volatility of a market. Higher ATR values indicate strong movement in either direction. ATR can also be used as a strategy in deciding stop-losses when it is combined with other technical indicators.

In this study, supervised PSVM approach have been used for which, training labels are assigned on the basis of closing price. $C(i)$ denotes the closing price of $i$th day and $C(i+1)$ denotes the closing price of next trading day. The training label for $i$th day, i.e., $y_i$ is computed based on the following rule.

$$y_i = \begin{cases} 1, & \text{If}\, C(i+1) > C(i) \\ -1, & \text{If otherwise} \end{cases} \tag{11}$$

**Table 3**
Optimal setting for the regularization parameter $\nu$ for the prediction models.

| Symbol/$\nu$ | PSVM | LC-PSVM | RC-PSVM | RR-PSVM | RF-PSVM |
|---|---|---|---|---|---|
| BSESN | 0.06399 | 0.0120 | 0.0120 | 0.0027 | 0.0383 |
| NSEI | 0.05099 | 0.0559 | 1.0399 | 0.4879 | 1.0339 |
| GDAXI | 0.00001 | 0.0357 | 1.0489 | 1.0489 | 0.0020 |
| N100 | 0.00001 | 0.9419 | 0.9419 | 1.0039 | 0.6899 |
| HSI | 0.00001 | 0.0171 | 0.0136 | 0.3529 | 0.2439 |
| KLSE | 0.00001 | 0.5909 | 0.5449 | 0.5699 | 0.1089 |
| N225 | 0.00001 | 0.0271 | 0.0271 | 0.3939 | 0.0137 |
| NYA | 0.65599 | 0.0007 | 0.0007 | 0.0008 | 0.0008 |
| RUA | 1.35999 | 0.1929 | 0.0001 | 0.0024 | 0.0001 |
| STI | 0.00398 | 0.9639 | 0.0001 | 0.7219 | 0.7929 |
| TWII | 0.00001 | 0.6489 | 0.0769 | 0.6209 | 0.0939 |
| JKSE | 0.50499 | 0.0829 | 0.0819 | 0.2689 | 0.1569 |

## 5. Performance measures and implementation of prediction model

### 5.1. Performance measures

Since, practitioners can take both long and short positions in case of upside and downside movement respectively to make profit, therefore, in financial forecasting recall and precision for stock rise and fall are equally important, Therefore, to evaluate the performance and robustness of the proposed models, we have used several measures which are computed from confusion matrix.

**Table 4**
Prediction accuracy and number of indicators used on the testing data of different stock indices obtained by five different models.

| | BSESN | NSEI | GDAXI | N100 | HSI | KLSE | N225 | NYA | RUA | STI | TWII | JKSE |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Train accuracy | % | % | % | % | % | % | % | % | % | % | % | % |
| Test accuracy | % | % | % | % | % | % | % | % | % | % | % | % |
| No. of features | *n* | *n* | *n* | *n* | *n* | *n* | *n* | *n* | *n* | *n* | *n* | *n* |
| ALL-BPNN | 49.37 | 59.59 | 60.36 | 48.66 | 59.79 | 63.20 | 50.68 | 56.58 | 46.82 | 49.84 | 65.52 | 62.05 |
| | 49.32 | 48.01 | 50.32 | 49.49 | 47.19 | 50.83 | 51.52 | 54.78 | 44.22 | 52.61 | 49.66 | 52.21 |
| | – | – | – | – | – | – | – | – | – | – | – | – |
| LC-BPN | 59.59 | 54.44 | 57.60 | 52.52 | 51.78 | 56.33 | 57.12 | 52.44 | 54.26 | 54.41 | 61.40 | 61.37 |
| | 56.75 | 56.62 | 49.03 | 57.52 | 57.73 | 55.85 | 56.61 | 59.07 | 52.14 | 55.22 | 54.02 | 53.58 |
| | 42 | 34 | 49 | 13 | 12 | 38 | 43 | 6 | 31 | 30 | 43 | 42 |
| RC-PBPN | 55.45 | 57.93 | 54.51 | 51.17 | 54.50 | 55.99 | 51.53 | 50.95 | 55.42 | 55.15 | 55.26 | 58.55 |
| | 56.08 | 59.60 | 52.59 | 57.19 | 57.75 | 55.85 | 56.94 | 46.86 | 56.76 | 56.20 | 57.71 | 55.29 |
| | 23 | 34 | 51 | 6 | 18 | 30 | 26 | 12 | 23 | 32 | 50 | 37 |
| RR-PBPN | 54.61 | 53.94 | 56.05 | 53.69 | 51.86 | 55.41 | 50.93 | 48.80 | 46.98 | 54.98 | 56.60 | 55.81 |
| | 57.09 | 55.62 | 45.45 | **58.19** | 56.43 | 57.85 | 53.22 | 44.88 | 54.45 | 53.92 | 57.38 | 52.55 |
| | 32 | 37 | 41 | 7 | 11 | 40 | 15 | 7 | 35 | 39 | 29 | 45 |
| RF-PBPN | 58.58 | 55.35 | 56.46 | 53.61 | 58.55 | 56.66 | 57.12 | 57.49 | 53.76 | 51.80 | 54.84 | 54.87 |
| | 55.40 | 56.29 | 53.87 | 55.85 | **58.41** | **60.20** | 57.98 | 54.78 | 54.78 | 55.55 | 56.04 | 60.45 |
| | 28 | 18 | 21 | 14 | 24 | 19 | 40 | 38 | 34 | 36 | 34 | 30 |
| ALL-PSVM | 50.63 | 50.54 | 50.77 | 51.85 | 50.87 | 49.96 | 51.69 | 55.30 | 51.53 | 51.47 | 51.89 | 49.66 |
| | 48.98 | 50.00 | 50.97 | 48.82 | 48.51 | 49.83 | 49.49 | 55.02 | 48.18 | 45.42 | 46.30 | 55.82 |
| | – | – | – | – | – | – | – | – | – | – | – | – |
| LC-PSVM | 56.21 | 56.27 | 51.99 | 52.77 | 53.67 | 54.82 | 53.14 | 50.45 | 50.45 | 55.07 | 56.18 | 55.47 |
| | 55.74 | 57.28 | 52.59 | 53.51 | 54.78 | 53.51 | 53.55 | 53.46 | 46.20 | 55.88 | 56.37 | 58.36 |
| | 41 | 42 | 10 | 17 | 18 | 54 | 13 | 5 | 2 | 49 | 44 | 54 |
| RC-PSVM | 56.21 | 56.27 | 54.02 | 52.77 | 53.92 | 54.65 | 53.14 | 50.45 | 54.34 | 52.61 | 54.58 | 55.21 |
| | 55.74 | 55.62 | 53.57 | 53.51 | 54.78 | 53.51 | 53.55 | 53.46 | 51.48 | 53.92 | 55.03 | 58.36 |
| | 40 | 27 | 16 | 17 | 19 | 22 | 13 | 5 | 50 | 46 | 26 | 50 |
| RR-PSVM | 53.93 | 56.27 | 54.02 | 52.52 | 55.99 | 54.48 | 53.22 | 52.61 | 53.93 | 55.23 | 55.34 | 55.38 |
| | 57.09 | 56.95 | 53.57 | 55.51 | 52.14 | 53.17 | 53.89 | 54.12 | 49.83 | 53.59 | 56.37 | 59.04 |
| | 11 | 53 | 16 | 22 | 22 | 54 | 12 | 16 | 21 | 37 | 25 | 44 |
| RF-PSVM | 57.14 | 55.68 | 51.75 | 51.93 | 54.91 | 51.47 | 54.75 | 50.04 | 54.84 | 52.21 | 53.49 | 55.47 |
| | **60.37** | **62.72** | **53.94** | 56.18 | 57.95 | 57.35 | **58.94** | **58.32** | **60.2** | **57.57** | **60.6** | **61.1** |
| | 48 | 47 | 7 | 4 | 16 | 7 | 20 | 7 | 39 | 3 | 8 | 28 |

Where, (%) is the accuracy in percentage and *n* is the number of features used. Best results are shown in bold.

| ↓ Predicted/Actual → | Rise | Fall |
|---|---|---|
| Rise | TR | FR |
| Fall | FF | TF |

where TR: Number of correctly predicted rise in stock index; TF: Number of correctly predicted fall in stock index; FR: Number of incorrectly predicted rise in stock index; FF: Number of incorrectly predicted fall in stock index.

Then, percentage accuracy (denoted by Accuracy (%)) is computed by following formula

$$\text{Accuracy}(\%) = \frac{TR+TF}{TR+FR+FF+TF} \times 100$$

#### 5.1.1. Precision

Precision denotes the fraction of predicted rise (or fall) in stock index those are truly rise (or fall)

$$\text{Precision for index Rise}\,(P^R) = \frac{TR}{(TR+FR)} \times 100$$

$$\text{Precision for index Fall}\,(P^F) = \frac{TF}{(TF+FF)} \times 100$$

**Table 5**
Optimal feature set produced by different hybrid prediction models based on accuracy (%).

| Model | RF-PSVM | | | | | | | | | RF-BPNN | | RR-BPNN |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| feature | BSESN | NSEI | GDAXI | N225 | NYA | RUA | STI | TWII | JKSE | HSI | KLSE | N100* |
| OP | √ | √ | | | | √ | | | √ | | | |
| HI | √ | √ | | √ | | √ | | | | | | √ |
| LO | √ | | | √ | | √ | | | | | | |
| CL | √ | √ | | | √ | √ | √ | | √ | | | √ |
| SMA(5) | √ | √ | | | | √ | | √ | √ | | √ | |
| SMA(10) | | √ | | √ | √ | √ | | | | | | |
| SMA(15) | √ | √ | | | | √ | | | √ | | | |
| SMA(20) | √ | √ | | √ | √ | √ | | | | | | |
| EMA(5) | | √ | | | | √ | | | | | | |
| EMA(10) | √ | √ | | | | √ | | | √ | √ | | |
| EMA(15) | √ | √ | | | √ | √ | | √ | √ | √ | | |
| EMA(20) | √ | √ | | | √ | √ | | | √ | | √ | √ |
| RDP(5) | √ | √ | | √ | | √ | | | √ | | | |
| RDP(10) | √ | | | | | | | | √ | √ | | |
| RDP(15) | √ | √ | | √ | | | | √ | √ | | | |
| RDP(20) | √ | √ | | | | | √ | | | | | |
| %K(5) | √ | √ | | √ | | | | | | | | |
| %D(5) | √ | √ | | | | √ | | | | √ | | |
| %R(5) | √ | √ | | | | | | | | √ | | |
| %R(10) | √ | √ | | | | | | | | | | |
| %R(15) | √ | √ | | √ | | | | | √ | | | |
| %R(20) | √ | √ | | | | √ | | | | √ | | |
| BIAS(5) | | √ | √ | | | √ | | | | | | |
| BIAS(10) | √ | √ | | | | √ | | | | | | |
| BIAS(15) | √ | √ | | | | √ | | | | | | |
| BIAS(20) | √ | √ | | | | √ | | | √ | | | |
| MACD(5,10) | | | | √ | | √ | | | √ | | | |
| MACD(5,15) | √ | √ | | | √ | | | | | | | |
| MACD(5,20) | | | | | | | | | | | | |
| MACD(10,15) | √ | √ | | | | √ | | | √ | √ | | |
| MACD(10,20) | √ | √ | | | | √ | | | √ | | | |
| MACD(15,20) | √ | √ | √ | √ | | √ | | | √ | | | |
| MTM(5) | √ | √ | | | | | | | √ | | | |
| MTM(10) | √ | √ | √ | | | | | | √ | √ | | |
| MTM(15) | √ | | √ | √ | | √ | | √ | | √ | | |
| MTM(20) | √ | √ | | √ | | √ | √ | | | √ | | |
| ROC(5) | √ | | | | | √ | | | √ | | | |
| ROC(10) | √ | √ | | | | √ | | | √ | √ | | |
| ROC(15) | √ | √ | | √ | | √ | | √ | | √ | | |
| ROC(20) | √ | √ | | | | | | | | √ | | |
| OSCP(5,10) | √ | √ | | | | √ | | | √ | | | |
| OSCP(5,15) | √ | √ | | √ | | √ | | | √ | | | |
| OSCP(5,20) | √ | √ | √ | √ | | √ | | | | | | √ |
| OSCP(10,15) | √ | √ | √ | √ | | √ | | | √ | √ | √ | |
| OSCP(10,20) | √ | | | | | √ | | | | | √ | |
| OSCP(15,20) | √ | √ | | √ | | √ | | | | √ | √ | |
| MP(10) | √ | √ | | √ | | | | | | √ | | |
| LL(10) | √ | √ | | | | √ | | √ | √ | | √ | |
| HH(10) | √ | √ | | | √ | √ | | √ | √ | | | |
| CCI(10) | √ | √ | | | | √ | | | √ | | | |
| Signal line(10,20) | √ | √ | | | | | | | √ | | | |
| ATR(10) | √ | √ | √ | √ | | √ | | √ | √ | | | |
| UO(10,20,30) | √ | √ | | | | √ | | | | | √ | |
| Ulcer(10) | | √ | | | | | | | | | | |
| TSI(10) | | | | √ | | | | | | | | |

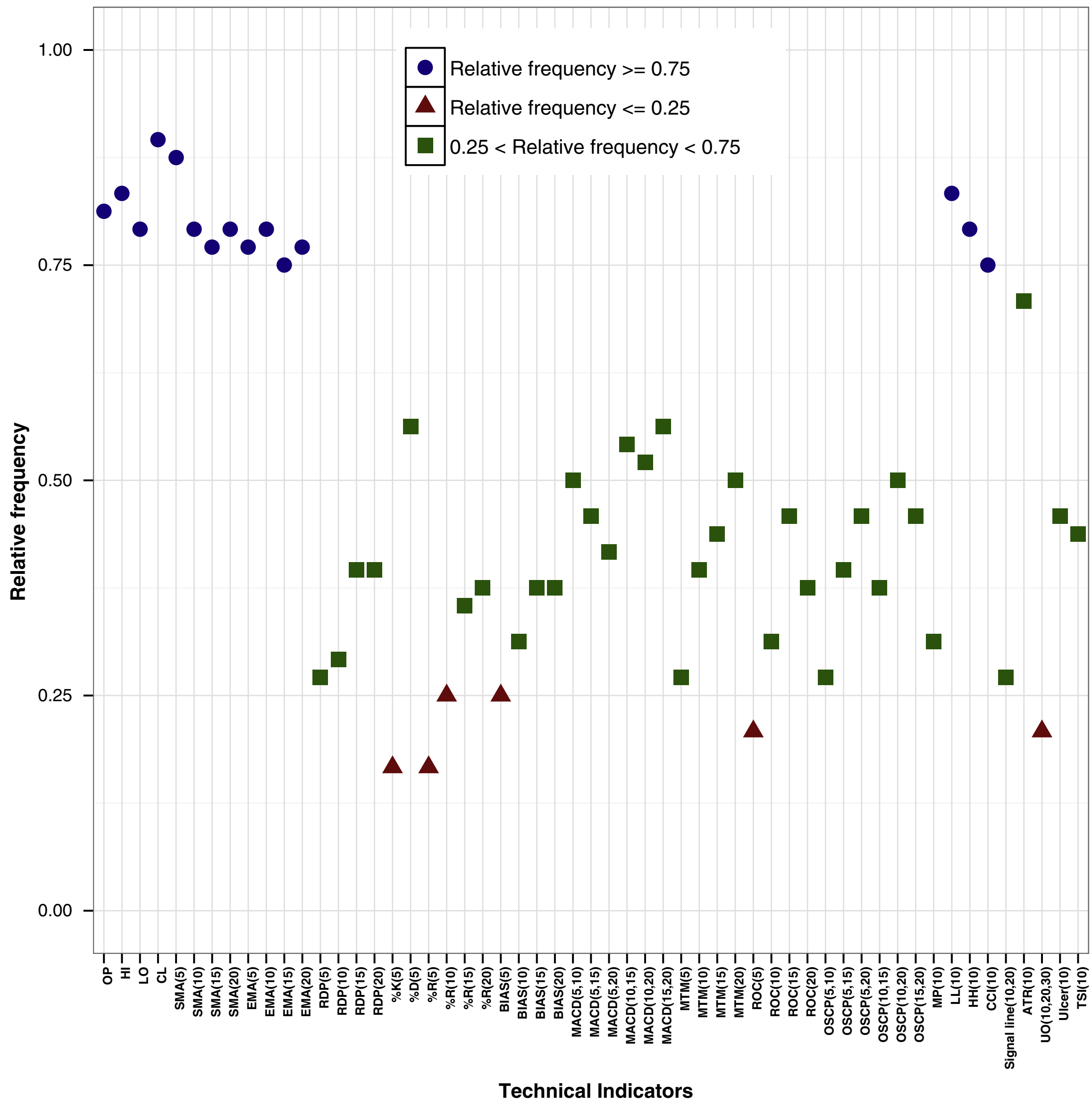


**Fig. 2.** Relative frequencies of selection for all fifty five features.

#### 5.1.2. *Recall*

Recall is the fraction of rise (or fall) in stock index those are predicted by the model

$$\text{Recall for index Rise}\,(R^R) = \frac{TR}{(TR + FF)} \times 100$$

$$\text{Recall for index Fall}\,(R^F) = \frac{TF}{(TF + FR)} \times 100$$

#### 5.1.3. F-*score*

*F*-score is the relation between true stock index (rise/fall) and those given by a predictor, if recall and precision are equally important, F-score is known as $F_1$ score and is defined as:

$$F_1\text{score for index Rise}(F_1^R) = \frac{2 \times P^R \times R^R}{(P^R + R^R)}$$

$$F_1\text{score for index Fall}(F_1^F) = \frac{2 \times P^F \times R^F}{(P^F + R^F)}$$

To evaluate combined effect of $F_1^R$ and $F_1^F$, we define a statistic called as joint prediction error (JPE).

$$\text{Joint prediction error}\,(JPE) = (1 - F_1^R)^2 + (1 - F_1^F)^2$$

It is easy to verify that $JPE \in [0, 2]$. The best prediction model would yield zero *JPE*, having $F_1^R$ and $F_1^F$ equals to 1 and is also called a perfect prediction model. For $JPE = 2$ it represents the worst prediction model.

### 5.2. *Implementation of prediction model*

The empirical computations were performed on the desktop machine having 2.8 GHz processor with 2 GB RAM and algorithms are implemented in Matlab 9.0.

#### 5.2.1. *Random forest implementation*

We have taken tree setting ($ntree = 500$) for computing the variable importance. Parameter *mtry* of random forest is the number of input variables randomly chosen at each node. Using large value

**Table 6**
Precision and recall for index rise and fall on testing data of different stock indices obtained by five different models.

| | | BPNN | | | | | PSVM | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Indices | | ALL | LC | RC | RR | RF | ALL | LC | RC | RR | RF |
| BSESN | $R^R$ | 3.92 | 36.60 | 28.10 | 31.37 | **51.63** | 42.48 | 45.10 | 45.10 | 45.10 | **55.75** |
| | $R^F$ | 97.90 | 78.32 | 86.01 | 84.62 | **59.44** | 55.94 | 67.13 | 67.13 | 69.93 | **65.33** |
| | $P^R$ | 66.67 | 64.37 | 68.25 | 68.57 | **57.66** | 50.78 | 59.48 | 59.48 | 61.61 | **63.06** |
| | $P^F$ | 48.78 | 53.59 | 52.79 | 53.54 | **53.46** | 47.62 | 53.33 | 53.33 | 54.35 | **55.14** |
| NSEI | $R^R$ | 34.48 | 38.62 | 29.66 | 35.17 | 28.28 | 56.55 | 31.72 | 34.48 | 29.66 | **51.72** |
| | $R^F$ | 60.51 | 73.25 | 87.26 | 74.52 | 82.17 | 43.95 | 80.89 | 75.16 | 82.17 | **72.89** |
| | $P^R$ | 44.64 | 57.14 | 68.25 | 56.04 | 59.42 | 48.24 | 60.53 | 56.18 | 60.56 | **60.53** |
| | $P^F$ | 50.00 | 56.37 | 57.32 | 55.45 | 55.36 | 52.27 | 56.19 | 55.40 | 55.84 | **56.19** |
| GDAXI | $R^R$ | 84.88 | 37.21 | 49.42 | 8.14 | 62.54 | 59.30 | 46.51 | 57.56 | 57.56 | **55.23** |
| | $R^F$ | 6.62 | 63.97 | 56.62 | 92.65 | 43.38 | 40.44 | 60.29 | 48.53 | 48.53 | **51.79** |
| | $P^R$ | 53.48 | 56.64 | 59.03 | 58.33 | 58.82 | 55.74 | 59.70 | 58.58 | 58.58 | **57.23** |
| | $P^F$ | 25.71 | 44.62 | 46.95 | 44.37 | 48.76 | 44.00 | 47.13 | 47.48 | 47.48 | **52.77** |
| N100 | $R^R$ | 89.54 | 72.55 | 72.55 | 78.43 | **60.13** | 56.21 | **52.94** | **52.94** | **51.63** | **60.78** |
| | $R^F$ | 7.53 | 41.10 | 41.10 | 36.99 | **51.37** | 41.10 | **54.11** | **54.11** | **59.59** | **51.37** |
| | $P^R$ | 50.37 | 56.35 | 56.35 | 56.60 | **56.44** | 50.00 | **54.73** | **54.73** | **57.25** | **56.71** |
| | $P^F$ | 40.74 | 58.82 | 58.82 | 62.07 | **55.15** | 47.24 | **52.32** | **52.32** | **54.04** | **55.56** |
| HSI | $R^R$ | 52.74 | **61.01** | 34.25 | 47.95 | **63.01** | 52.74 | 33.56 | 34.25 | 41.10 | **52.47** |
| | $R^F$ | 42.04 | **56.69** | 75.62 | 64.33 | **54.14** | 44.59 | 74.52 | 73.89 | 62.42 | **63.06** |
| | $P^R$ | 45.83 | **55.50** | 60.98 | 55.56 | **56.10** | 46.95 | 55.06 | 54.95 | 50.42 | **51.67** |
| | $P^F$ | 48.89 | **56.24** | 56.56 | 57.06 | **61.15** | 50.36 | 54.67 | 54.72 | 53.26 | **54.10** |
| KLSE | $R^R$ | 46.67 | 48.67 | 66.00 | 46.00 | **56.67** | 50.00 | 39.33 | 39.33 | 38.00 | **53.00** |
| | $R^F$ | 55.03 | 63.09 | 45.64 | 69.80 | **63.76** | 49.66 | 67.79 | 67.79 | 68.46 | **61.75** |
| | $P^R$ | 51.09 | 57.03 | 55.00 | 60.53 | **61.15** | 50.00 | 55.14 | 55.14 | 54.81 | **55.81** |
| | $P^F$ | 50.62 | 54.97 | 57.14 | 56.22 | **59.38** | 49.66 | 52.60 | 52.60 | 52.31 | **54.12** |
| N225 | $R^R$ | 41.21 | **60.61** | 65.45 | 46.06 | **52.73** | 64.85 | 52.12 | 52.12 | 51.52 | **50.18** |
| | $R^F$ | 64.62 | **51.54** | 46.15 | 62.31 | **64.92** | 30.00 | 55.38 | 55.38 | 56.92 | **70.08** |
| | $P^R$ | 59.65 | **61.35** | 60.67 | 60.80 | **64.92** | 54.04 | 59.72 | 59.72 | 60.28 | **64.29** |
| | $P^F$ | 46.41 | **50.76** | 51.28 | 47.65 | **50.73** | 40.21 | 47.68 | 47.68 | 48.05 | **52.22** |
| NYA | $R^R$ | 67.44 | 84.81 | 21.51 | 21.51 | 65.70 | 68.94 | 47.09 | 47.09 | 63.37 | **60.47** |
| | $R^F$ | 38.17 | 30.53 | 80.15 | 75.57 | 40.46 | 44.04 | 61.83 | 61.83 | 41.98 | **55.51** |
| | $P^R$ | 58.88 | 58.43 | 58.73 | 53.62 | 59.16 | 55.56 | 61.83 | 61.83 | 58.92 | **58.43** |
| | $P^F$ | 47.17 | 54.79 | 43.75 | 42.31 | 47.32 | 52.17 | 47.09 | 47.09 | 46.61 | **54.60** |
| RUA | $R^R$ | 1.74 | 48.84 | 49.42 | 62.21 | 45.35 | 54.65 | 52.91 | 37.21 | 30.81 | **52.35** |
| | $R^F$ | 100.00 | 56.49 | 66.41 | 44.27 | 67.18 | 39.69 | 37.40 | 70.23 | 74.81 | **70.52** |
| | $P^R$ | 100.00 | 59.57 | 65.89 | 59.44 | 64.46 | 54.34 | 52.60 | 62.14 | 61.63 | **68.42** |
| | $P^F$ | 43.67 | 45.68 | 50.00 | 47.15 | 48.35 | 40.00 | 37.69 | 46.00 | 45.16 | **50.26** |
| STI | $R^R$ | 96.32 | 42.33 | **60.12** | 49.69 | 39.88 | 45.40 | 61.96 | 66.87 | 66.87 | **55.99** |
| | $R^F$ | 2.80 | 69.93 | **51.75** | 58.74 | 73.43 | 45.45 | 48.95 | 39.16 | 38.46 | **59.24** |
| | $P^R$ | 53.04 | 61.61 | **58.68** | 57.86 | 63.11 | 48.68 | 58.05 | 55.61 | 55.33 | **57.89** |
| | $P^F$ | 40.00 | 51.55 | **53.24** | 50.60 | 51.72 | 42.21 | 53.03 | 50.91 | 50.46 | **53.30** |
| TWII | $R^R$ | 48.77 | 51.85 | **59.26** | 74.69 | 61.11 | 37.65 | 74.07 | 69.14 | 71.60 | **64.20** |
| | $R^F$ | 50.74 | 56.62 | **55.88** | 36.76 | 50.00 | 56.62 | 35.29 | 38.24 | 38.24 | **56.32** |
| | $P^R$ | 54.11 | 58.74 | **61.54** | 58.45 | 59.28 | 50.83 | 57.69 | 57.14 | 58.00 | **58.76** |
| | $P^F$ | 45.39 | 49.68 | **53.52** | 54.95 | 51.91 | 43.26 | 53.33 | 50.98 | 53.06 | **56.07** |
| JKSE | $R^R$ | 67.28 | 53.70 | **53.09** | 52.47 | 80.95 | **56.79** | **59.26** | **59.26** | **63.58** | **67.90** |
| | $R^F$ | 33.59 | 53.44 | **58.02** | 52.67 | 35.88 | **53.22** | **57.25** | **57.25** | **53.44** | **52.67** |
| | $P^R$ | 55.61 | 58.78 | **60.99** | 57.82 | 61.82 | **54.44** | **63.16** | **63.16** | **62.80** | **63.95** |
| | $P^F$ | 45.36 | 48.28 | **50.00** | 47.26 | 64.38 | **53.55** | **53.19** | **53.19** | **54.26** | **57.02** |

Where $R^R$ is the recall for index rise, $R^F$ is the recall for index fall, $P^R$ is the precision for index rise, $P^F$ is the precision for index fall. Best results are shown in bold.

of *mtry* just amplifies the variable importance of the features. Hence, we have chosen parameter value $mtry = 18(n/3)$ for random forest.

### 5.2.2. PSVM implementation

Linear PSVM has only one penalty parameter $\nu$ to tune which affects the performance of PSVM. In our experiment five-fold CV method, which has been extensively used in conjunction of SVM for financial time series data [25,19,24,23], is used to choose the value of the parameter $\nu$. The value of the parameter $\nu$ which gives the maximum accuracy over training data is chosen. The optimal values of $\nu$, after conducting 5−CV method over training data, is summarized in Table 3.

### 5.2.3. BPNN implementation

In order to evaluate the performance of the proposed models, the performance of proposed hybrid strategy is compared with BPNN. The BPNN used in this study has $n$ input nodes and $2n$ hidden nodes according to the number of input features [19,25]. The training epoch are set to be 1000 and the optimal value of momentum factor ranged from set $\{0.1, 0.2, 0.30, 0.4, 0.5\}$. The learning rate was selected from $\{0.1, 0.25, 0.5, 0.75, 0.9\}$.

## 6. Results and discussions

To check the performance of the proposed hybrid models, various performance metrics (discussed in Section 5.1) are

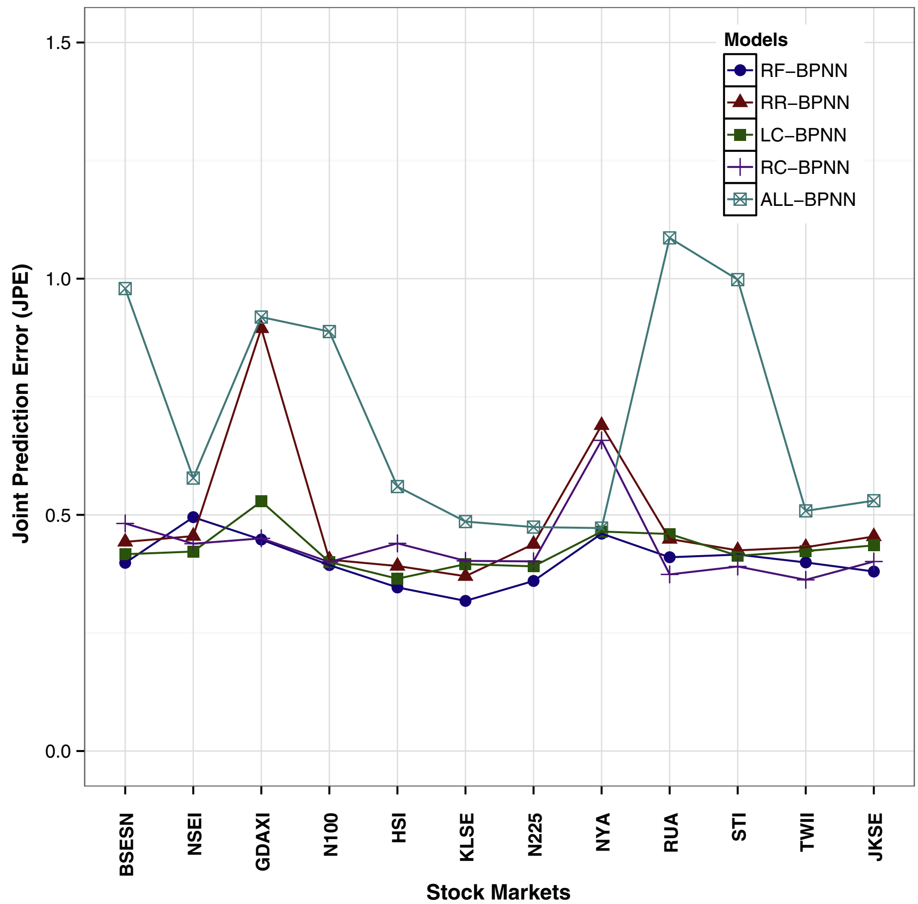


**Fig. 3.** JPE for all indices by BPNN along with all feature selection techniques.

**Table 7**
Wilcoxon rank sum test for pairwise comparison of performance.

| | | | BPNN | | | | | PSVM | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | ALL | LC | RC | RR | RF | ALL | LC | RC | RR |
| BPNN | LC | *z*-statistic | **−3.31** | | | | | | | | |
| | | *p* | **9.0E−4** | | | | | | | | |
| | RC | *z*-statistic | **−3.37** | −0.57 | | | | | | | |
| | | *p* | **7.0E−4** | 0.56 | | | | | | | |
| | RR | *z*-statistic | **−2.56** | 0.66 | 0.95 | | | | | | |
| | | *p* | **0.01** | 0.51 | 0.34 | | | | | | |
| | RF | *z*-statistic | **−4.01** | −0.57 | 0 | −1.35 | | | | | |
| | | *p* | **5.0E−5** | 0.56 | 1 | 0.17 | | | | | |
| PSVM | ALL | *z*-statistic | 0.51 | **3.37** | **3.37** | **2.28** | **3.66** | | | | |
| | | *p* | 0.60 | **7.3E−4** | **7.3E−4** | **0.022** | **2.4E−4** | | | | |
| | LC | *z*-statistic | **−3.12** | 1.24 | 1.56 | 0.14 | 1.94 | **−2.74** | | | |
| | | *p* | **1.8E−3** | 0.21 | 0.12 | 0.88 | 0.52 | **6.1E−3** | | | |
| | RC | *z*-statistic | **−3.52** | 1.70 | 2.17 | 0.43 | **−2.69** | **−3.03** | −1.73 | | |
| | | *p* | **4.3E−4** | 0.09 | 0.03 | 0.66 | **7.2E−3** | **2.4E−3** | 0.86 | | |
| | RR | *z*-statistic | **−3.32** | 0.92 | 1.36 | 0.35 | 1.81 | **−3.03** | −0.23 | 0.46 | |
| | | *p* | **9.0E−4** | 0.36 | 0.17 | 0.72 | 0.07 | **2.4E−3** | 0.82 | 0.64 | |
| | RF | *z*-statistic | **−4.07** | **−2.63** | **−2.63** | **−3.09** | **−1.96** | **−4.01** | **−3.38** | **−3.26** | **−3.26** |
| | | *p* | **4.7E−5** | **8.6E−3** | **8.6E−3** | **2.0E−3** | **0.05** | **6.1E−6** | **7.3E−5** | **1.1E−3** | **1.1E−3** |

Significant difference in performance between different prediction models are marked bold.

calculated for twelve stock indices and for all proposed hybrid models described in Sections 3 and 5.2.3.

The results obtained thus, are reported in Table 4. All the models are compared in terms of number of features selected and their corresponding testing performance. It is clear from Table 4 that the highest testing accuracy (marked in bold face) for nine out of twelve stock indices considered in this study is achieved by RF-PSVM model. All four feature selection techniques clubbed with PSVM (LC-PSVM, RC-PSVM, RR-PSVM and RF-PSVM) achieve higher accuracy as compared to ALL-PSVM in eleven out of twelve indices (except NYA index). On another note, RF-PSVM is the only hybrid model which achieves higher accuracy than ALL-PSVM for NYA index. Also, RR-PSVM, RC-PSVM and LC-PSVM do not show any significant difference in the accuracy. When BPNN is clubbed with four feature selection algorithms considered it results in four hybrid BPNN methods (LC-BPNN, RC-BPNN, RR-BPNN and RF-BPNN). Al

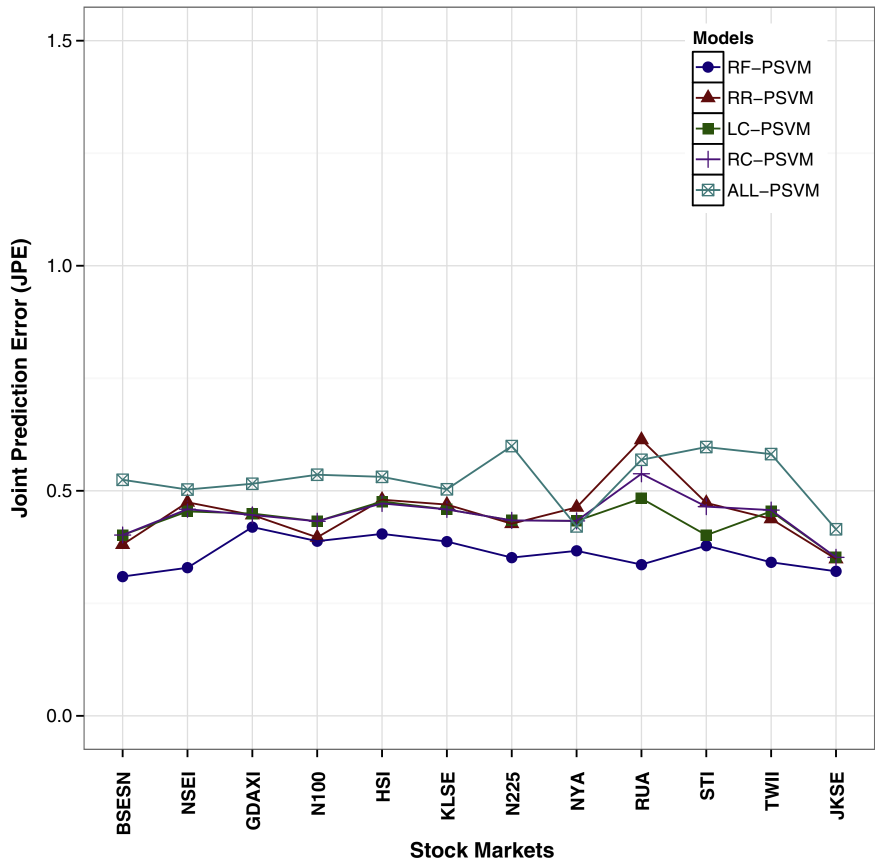


**Fig. 4.** JPE for all indices by PSVM along with all feature selection techniques.

these hybrid BPNN models also achieves higher accuracy as compared to ALL-BPNN in ten out of twelve indices (except for GDAXI and NYA index). RF-BPNN is the only hybrid model which was able to achieves higher accuracy than ALL-BPNN for all twelve stock indices. These results confirm the claims made in [23,24], that learning algorithm with feature selection techniques outperform the individual learning models.

It is observed that in most of the cases, less than twenty features (out of fifty-five) are required by hybrid models to beat the accuracy of ALL-PSVM model. Hence, dimension reduction not only helps in reducing computational time but also helps in improving accuracy of the model. The selected optimal feature set using different feature selection techniques considered in this study for all twelve indices are shown in Table 5. Results show that optimal feature set in all twelve indices is determined by RF in conjunction with PSVM or BPNN except for *N*100. For *N*100, the optimal feature set is determined using RR-BPNN. From Table 5, it also clear that the optimal feature subset for different indices is different. Fig. 2 shows dissimilar prediction capability of various technical indicators used in current study. To produce a consolidated list of relevant indicators selected by different stock indices, we compute relative frequency of selected features by all hybrid models in all indices. Relative frequency of fifty-five features considered in this study is depicted in Fig. 2. High (HI) and Close(CL) are selected more than 85% of times while %K(5), %D(5) are selected with less than 20% of times. Open(OP), Low (LO), Simple moving averages (SMA(5), SMA(10), SMA(15), SMA(20)), exponential moving average (EMA(5), EMA(10), EMA(15), EMA(20)), Lowest Low (LL(10)), Highest High (HH(10)) and Commodity Channel Index (CCI(10)) are selected more than 75% of times, which clearly establish the superiority of these technical indicators over others in the trend prediction of the considered indices.

Performance evaluation of prediction model only on the basis of classifying accuracy can be misleading. To check the bias of each of the model, we have to assess the precision and recall of the classifier. Performance metrics of various feature selection technique with PSVM and BPNN as classifier is reported in Table 6. For example, for GDAXI index, BPNN and their hybrid models are biased towards predicting either only stock rise or stock fall. BPNN classification of stock rise and fall along with various feature selection technique yields biased classifier. Even though its prediction capability of stock rise is higher, it is not able to classify events when the stock index falls.

The results indicate that RF-PSVM is the only model which achieves higher than 50% precision and recall for stock index rise and fall for all indices considered in this study. ALL-BPNN and RR-BPNN models, are unable to achieve higher than 50% precision and recall accuracy for many stock indices. On an average, RF-PSVM outperforms all other models followed by RF-BPNN. The performance of LC-PSVM, RC-PSVM, RR-PSVM, LC-BPNN, RC-BPNN, and RR-BPNN models are found to be more or less similar. ALL-BPNN is found to have the lowest precision and recall rate for stock index rise and stock index fall among all stock indices considered in this study.

To study the joint effect of the $F_1^R$ and $F_1^F$, we define a performance metric called Joint Prediction Error (JPE) (discussed in Section 5.1). With this performance metric, in most of the models for different indices, ALL-BPNN has maximum JPE. Therefore, ALL-BPNN has worst performance when compared with other hybrid models as depicted in Fig. 3.

Performance of LC-PSVM, RC-PSVM and RR-PSVM are found to be more or less similar on the basis of this criteria. It is also noted from Figs. 3 and 4 that among the all hybrid models considered RF-PSVM shows least JPE in ten out of twelve indices under

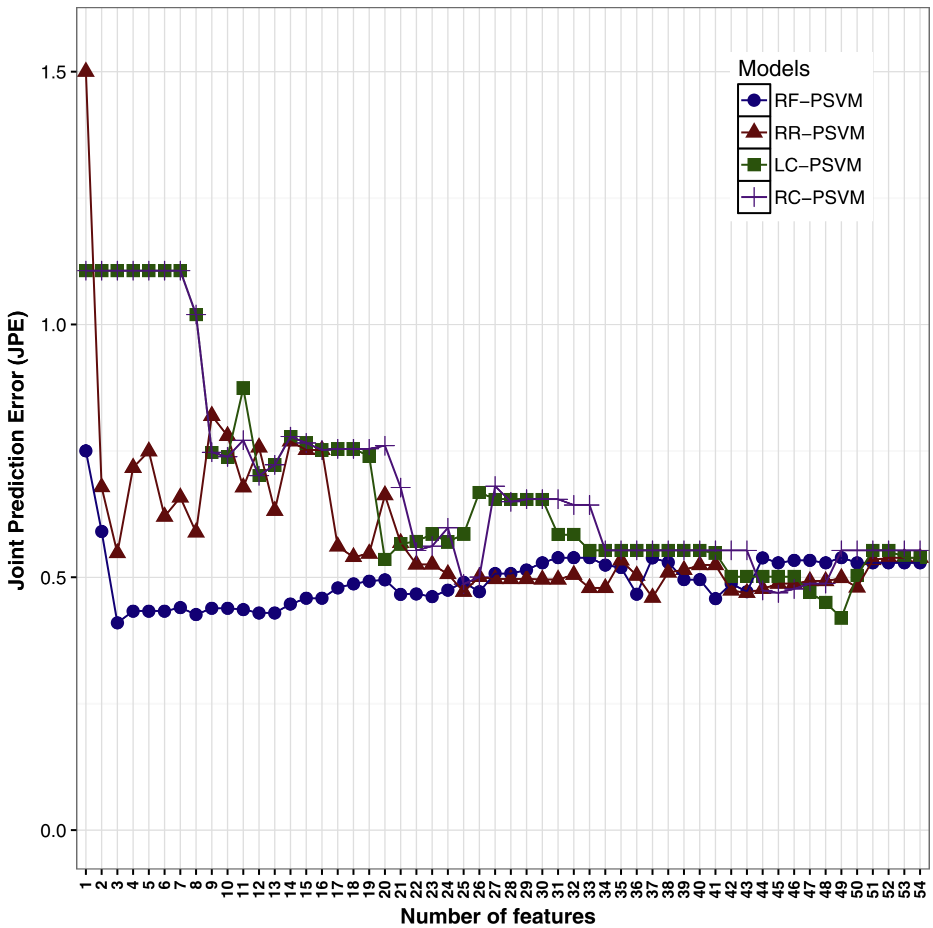


**Fig. 5.** Effect of increasing features on JPE for STI for PSVM based hybrid model. Variation of JPE with number of features for PSVM based hybrid models on STI.

consideration which indicates the superiority of RF-PSVM over all other methods in question.

Performance of each hybrid model with respect to complexity of the model is also assessed through JPE. Here also RF-PSVM outperforms all other models. For e.g., Fig. 5 shows JPE of all models for STI index. Clearly, performance of the RF-PSVM is more stable and better than others.

Through various performance metrics, we can judge the performance of these models. However, it is unclear whether there is statistically significant difference between them. In order to examine which prediction model significantly outperforms other models listed in this paper, the Wilcoxon rank sum test is performed at 0.05 significance level. Table 7 shows the result of the Wilcoxon rank sum test for all twelve stock indices with respect to testing data among the five models.

From Table 7, it is clear that the prediction from all hybrid models provide superior results as compared to the ALL-PSVM and ALL-BPNN. This indicates that there is a significant positive impact of feature selection methods on the performance of the prediction model. Here also it can be concluded that RF-PSVM outperforms all other models considered in this work.

## 7. Conclusion

In this study we have introduced and compared four hybrid models LC-PSVM, RC-PSVM, RR-PSVM and RF-PSVM. To test the robustness and performance of these models, empirical studies have been performed on twelve stock indices of different financial markets across the globe. Performance of these models is evaluated for financial trend prediction on the basis of various performance metrics, viz. testing accuracy, precision, recall, $F_1$ score and a newly proposed criteria JPE.

Empirical findings suggest superiority of proposed hybrid models, when compared with original PSVM and BPNN algorithms without any feature selection. Hence, dimension reduction not only yields parsimonious model, but also improves the prediction accuracy of the model. It is observed that RF-PSVM outperforms all other models over all performance metrics for ten out of twelve stock indices considered in this study. This suggests better generalization ability of random forest algorithm as compared to RReliefF and correlation criteria based feature selection methods. Apart from RF-PSVM, all other hybrid models, LC-PSVM, RC-PSVM, RR-PSVM, LC-BPNN, RC-BPNN and RR-BPNN on an average show almost similar performance, when compared among themselves. Present experiment also discovers that PSVM is less biased than BPNNs, i.e., its classifying capability to predict both stock rise and stock fall is comparable. This is one of the most desirable characteristic of any classifier used for prediction in financial markets and is not observed in BPNN for most of the stock indices.

The study also establishes superiority of several technical indicators over others, e.g., open, high, low, close, simple moving averages, exponential moving averages, lowest low, highest high and commodity channel index are found to have more than 75% of relative frequency of selection by proposed hybrid models for all stock indices considered in this study.

One of the future direction to extend the findings of current study is to understand theoretical properties of RF-PSVM for its better generalization ability. Further, introducing more input feature (e.g., inclusion of fundamental and micro economic variables) for the prediction algorithms in financial domain is also a very important area of consideration.

## References


[1] Y.S. Abu-Mostafa, A.F. Atiya, Introduction to financial forecasting, Appl. Intell. 6 (3) (1996) 205–213.
[2] G. Deboeck, Trading on the Edge: Neural, Genetic, and Fuzzy Systems for Chaotic Financial Markets, Vol. 39, John Wiley & Sons, 1994.
[3] S.C. Blank, "chaos" in futures markets? A nonlinear dynamical analysis, J. Futures Mark. 11 (6) (1991) 711–728.
[4] C. Cortes, V. Vapnik, Support-vector networks, Mach. Learn. 20 (3) (1995) 273–297.
[5] K. Hornik, M. Stinchcombe, H. White, Multilayer feedforward networks are universal approximators, Neural Netw. 2 (5) (1989) 359–366.
[6] S. Haykin, R. Lippmann, Neural networks, a comprehensive foundation, Int. J. Neural Syst. 5 (4) (1994) 363–364.
[7] G. Grudnitski, L. Osburn, Forecasting s&p and gold futures prices: an application of neural networks, J. Futures Mark. 13 (6) (1993) 631–643.
[8] W. Leigh, R. Purvis, J.M. Ragusa, Forecasting the NYSE composite index with technical analysis, pattern recognizer, neural network, and genetic algorithm: a case study in romantic decision support, Decis. Support Syst. 32 (4) (2002) 361–377.
[9] G.P. Zhang, An investigation of neural networks for linear time-series forecasting, Comput. Oper. Res. 28 (12) (2001) 1183–1202.
[10] L. Cao, F. Tay, Support vector machine with adaptive parameters in financial time series forecasting, IEEE Trans. Neural Netw. 14 (6) (2003) 1506–1518.
[11] K.-j. Kim, Financial time series forecasting using support vector machines, Neurocomputing 55 (1) (2003) 307–319.
[12] V.N. Vapnik, V. Vapnik, Statistical Learning Theory, Vol. 1, Wiley, New York, 1998.
[13] V.N. Vapnik, The Nature of Statistical Learning Theory, Springer-Verlag, New York, Inc., New York, NY, USA, 1995.
[14] V.N. Vapnik, An overview of statistical learning theory, IEEE Trans. Neural Netw. 10 (5) (1999) 988–999.
[15] L. Cao, F.E. Tay, Financial forecasting using support vector machines, Neural Comput. Appl. 10 (2) (2001) 184–192.
[16] W. Huang, Y. Nakamori, S.-Y. Wang, Forecasting stock market movement direction with support vector machine, Comput. Oper. Res. 32 (10) (2005) 2513–2522.
[17] G.S. Atsalakis, K.P. Valavanis, Surveying stock market forecasting techniques-part. II: Soft computing methods, Expert Syst. Appl. 36 (3) (2009) 5932–5941.
[18] L. Cao, Support vector machines experts for time series forecasting, Neurocomputing 51 (2003) 321–339.
[19] L. Yu, H. Chen, S. Wang, K.K. Lai, Evolving least squares support vector machines for stock market trend mining, IEEE Trans. Evolut. Comput. 13 (1) (2009) 87–102.
[20] C.-F. Tsai, Y.-C. Hsiao, Combining multiple feature selection methods for stock prediction: union, intersection, and multi-intersection approaches, Decis. Support Syst. 50 (1) (2010) 258–269.
[21] I. Guyon, A. Elisseeff, An introduction to variable and feature selection, J. Mach. Learn. Res. 3 (2003) 1157–1182.
[22] R. Kohavi, G.H. John, Wrappers for feature subset selection, Artif. Intell. 97 (1) (1997) 273–324.
[23] L. Cao, K. Chua, W. Chong, H. Lee, Q. Gu, A comparison of PCA, KPCA and ICA for dimensionality reduction in support vector machine, Neurocomputing 55 (1) (2003) 321–336.
[24] C.-L. Huang, C.-Y. Tsai, A hybrid SOFM-SVR with a filter-based feature selection for stock market forecasting, Expert Syst. Appl. 36 (2 (Part 1)) (2009) 1529–1539.
[25] M.-C. Lee, Using support vector machine with a hybrid feature selection method to the stock trend prediction, Expert Syst. Appl. 36 (8) (2009) 10896–10904.
[26] G. Fung, O.L. Mangasarian, Proximal support vector machine classifiers, in: Proceedings of the seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2001, pp. 77–86.
[27] J.A. Suykens, J. Vandewalle, Least squares support vector machine classifiers, Neural Process. Lett. 9 (3) (1999) 293–300.
[28] T. Joachims, Making large scale SVM learning practical, Tech. rep., Universität Dortmund, 1999.
[29] I. Guyon, J. Weston, S. Barnhill, V. Vapnik, Gene selection for cancer classification using support vector machines, Mach. Learn. 46 (1–3) (2002) 389–422.
[30] K. Kira, L.A. Rendell, The feature selection problem: traditional methods and a new algorithm, in: AAAI, vol. 2, 1992, pp. 129–134.
[31] M. Robnik-Šikonja, I. Kononenko, Theoretical and empirical analysis of relieff and rrelieff, Mach. Learn. 53 (1-2) (2003) 23–69.
[32] L.-P. Ni, Z.-W. Ni, Y.-Z. Gao, Stock trend prediction based on fractal feature selection and support vector machine, Expert Syst. Appl. 38 (5) (2011) 5569–5576.
[33] M. Robnik-Šikonja, I. Kononenko, An adaptation of relief for attribute estimation in regression, in: Machine Learning: Proceedings of the Fourteenth International Conference (ICML'97), 1997, pp. 296–304.
[34] L. Breiman, Random forests, Mach. Learn. 45 (1) (2001) 5–32.
[35] R. Genuer, J.-M. Poggi, C. Tuleau-Malot, Variable selection using random forests, Pattern Recogn. Lett. 31 (14) (2010) 2225–2236.
[36] C. Strobl, A.-L. Boulesteix, A. Zeileis, T. Hothorn, Bias in random forest variable importance measures: illustrations, sources and a solution, BMC Bioinform. 8 (1) (2007) 25.
[37] B.N. Parlett, The Symmetric Eigenvalue Problem, vol. 7, SIAM, 1980.
[38] L. Menkhoff, Examining the use of technical currency analysis, Int. J. Finance Econ. 2 (4) (1997) 307–318.
[39] M. Dash, H. Liu, Feature selection for classification, Intell. Data Anal. 1 (1) (1997) 131–156.
[40] R. Genuer, J.-M. Poggi, C. Tuleau, Random forests: some methodological insights, 2008, arXiv:0811.3619.
[41] T. Hastie, R. Tibshirani, J. Friedman, T. Hastie, J. Friedman, R. Tibshirani, The Elements of Statistical Learning, Vol. 2, Springer, 2009.


**Manoj Thakur** is a faculty at Indian Institute of Technology Mandi. Before joining IIT Mandi he was in Industry for more than three years and worked there as Research Analyst in Financial domain. He did M.Sc. from University of Roorkee and M.Phil. and then Ph.D. from IIT Roorkee. He has published several research papers in referred journals and conferences. His areas of research are Numerical Optimization, Machine Learning and its application to Finance.