

# A novel hybrid model using teaching–learning-based optimization and a support vector machine for commodity futures index forecasting

Shom Prasad Das<sup>1</sup> · Sudarsan Padhy<sup>2</sup>

Received: 21 August 2014 / Accepted: 3 April 2015 / Published online: 21 April 2015  
© Springer-Verlag Berlin Heidelberg 2015

**Abstract** The analysis and prediction of financial time-series data are difficult, and are the most complicated tasks concerned with improving investment decisions. In this study, we forecasted a financial derivatives instrument (the commodity futures contract index) using techniques based on recently developed machine learning techniques. These methods have been shown to perform remarkably well in other applications. In particular, we developed a hybrid method that combines a support vector machine (SVM) with teaching–learning-based optimization (TLBO). The proposed SVM–TLBO model avoids user-specified control parameters, which are required when using other optimization methods. We assessed the viability and efficiency of this hybrid model by forecasting the daily closing prices of the COMDEX commodity futures index, traded in the Multi Commodity Exchange of India Limited. Our experimental results show that the proposed model is effective and performs better than the particle swarm optimization (PSO) + SVM hybrid and standard SVM models. For example, the proposed model improved the MAE by 65.87 % (1-day-ahead forecast), 55.83 % (3-days-ahead forecast), and 67.03 % (5-days-ahead forecast), when compared with standard SVM regression.

**Keywords** Support vector machine (SVM) · Teaching–learning-based optimization (TLBO) · Commodity futures contract · Financial time series

## 1 Introduction

In the business environment, we wish to accurately and efficiently forecast various kinds of financial variables to develop successful strategies and avoid large losses [43]. Many researchers have considered financial time series data forecasts since the 1980s, with the objective of beating the financial market. There are a huge number of factors (economic, political, environmental, and psychological) that make financial forecasting an interesting and challenging field. Further, financial time series are inherently noisy, nonstationary, and deterministically chaotic [2, 38].

Conventional statistical methods such as time series and multivariate analyses have been used in most prediction techniques. However, researchers have started to apply artificial intelligence (AI) methods to financial markets, because of recent successful developments such as artificial neural networks (ANNs), support vector machines (SVMs), particle swarm optimization (PSO), genetic algorithms (GAs), and fuzzy technologies. Refenes et al. [34], Tsibouris and Zeidenberg [40], and Steiner and Wittkemper [37] used ANN models to predict stock prices in the UK, US, and German markets, respectively. Wittkemper and Steiner [42] and Shazly et al. [36] used ANNs with GAs (hybrid models) to predict stock prices and currency exchange rates in Germany, the United Kingdom, Japan, and Switzerland.

Vapnik [41] introduced SVM methods to overcome the problems of ANNs (such as getting trapped in local minima, overfitting to training data, and long training times). Since then, several authors have proposed financial

---

✉ Shom Prasad Das  
shomdas@yahoo.com  
Sudarsan Padhy  
spadhy07@gmail.com

<sup>1</sup> Department of Computer Science and Engineering, National Institute of Science and Technology, Palur Hills, Berhampur, Odisha 761008, India

<sup>2</sup> Institute of Mathematics and Applications, Bhubaneswar, Odisha 751003, India

instruments pricing using SVMs. For example, Tay and Cao [38] and Cao and Tay [2] developed pricing models for five specific financial futures in the US market using SVMs, and Gestel et al. [7] used LS-SVM (Least squares support vector machine) for T-bill (Treasury bill) rate and stock index pricing in the US and German markets. Their simulated results show that SVM outperforms ANNs. Moreover, SVM has also been shown to perform better than ANNs and other statistical methods in other domains [23, 24].

Nicholas and Ravi [25] published an exhaustive survey on SVMs for time series prediction. They surveyed papers in the areas of financial market prediction, electric utility load forecasting, environment states, weather prediction, and reliability forecasting. In their survey, they noted that the free parameters chosen for the SVM are significant. The experimental result by Kim [14] showed that SVM predictions are sensitive to these free parameters, and that it is important to select optimal values. Improperly selected free parameters can cause over- or under-fitting problems [14]. The nonlinear nature of financial time series data means that we use nonlinear kernel functions such as the Gaussian and polynomial functions, which require appropriate, user chosen parameter(s). Current approaches for choosing these free parameters are typically based on domain knowledge, cut and try, and ergodic search methods [4]. Several studies proposed selecting the optimal free parameters for SVM/ANNs using PSO, GAs, artificial bee colonies (ABCs), ant colony optimization (ACO), differential evolution (DE), simulated annealing (SA), and so on [3, 11, 12, 21, 22, 44]). However, the optimization model itself introduces additional user-specified controlling parameter(s), making the user's task even more complex. For example, GAs need optimal controlling parameter values for crossover and mutation probabilities; PSO needs specified optimal controlling parameters such as inertia weight and social and cognitive parameters; SA needs cooling temperature and cooling constants; DE requires a differentiation factor and a crossover probability; and ABC requires the optimal controlling parameters for the number of bees (employed, scout, and onlookers), limits, and so on. Variations to the controlling parameters alter the effectiveness of the optimization algorithm.

Rao et al. [30] proposed the teaching–learning-based optimization (TLBO) algorithm, which is an optimization technique for a mechanical design problem that does not require user defined parameters. They tested their novel technique using different benchmark functions. Their results show that TLBO can outperform many optimization algorithms such as particle evolutionary swarm optimization, ABC, and cultural DE. Similarly, Rao et al. [31] compared TLBO with well-known optimization techniques such as GA, ABC, PSO, HS, DE, and hybrid-PSO, by applying the methods to different benchmark problems

[such as Griewank ( $D = 10$ ), Hyper Sphere ( $D = 6$ ), Rosenbrock ( $D = 1$ ,  $D = 3$ ), Rastrigin, and Ackley]. They considered the effectiveness of TLBO in terms of different performance criteria (such as the average number of function evaluations, success rate, convergence rate, and mean solution). These results also showed that the TLBO method performed better than other nature-inspired optimization techniques, for the considered benchmark functions. The TLBO optimization technique developed by Rao et al. [30] performed well in many studies [26–33, 35, 45].

In this study, we propose an SVM–TLBO hybrid regression prediction model for forecasting the multicommodity futures index (COMDEX) traded in the Multi Commodity Exchange of India Limited (MCX). We use the TLBO algorithm to select the free parameter(s) of the SVM, and the free parameter(s) of the kernel function. We compared the standard SVM method with the SVM–TLBO hybrid technique. The commodity futures index under consideration is a significant indicator for the performance of the Indian commodities market. MCX COMDEX is composed of futures contracts on 15 physical commodities with three sub-indices, representing the key commodity sectors within the index: metals, energy, and agricultural. Investors can use MCX COMDEX futures to efficiently hedge commodity and inflation exposure and lay off residual risk [1]. We developed the SVM–TLBO hybrid regression model because the most important consideration when using the standard SVM model is to properly select the free parameters [ $C$  (regularization) and  $\varepsilon$  (insensitive loss function radius)] and the kernel parameter(s) for training the data.

TLBO does not require any user-defined, controlling parameters(s), which means that can it effectively determine the free parameter(s) of the SVM without any user input. Our experimental results show that the proposed hybrid SVM–TLBO regression model produces better forecasts than the PSO + SVM hybrid and standard SVM models. The remainder of this paper is structured as follows. In Sect. 2, we provide a summary of SVM regression and the SVM-TLBO hybrid regression model for selecting the optimal free parameters. Section 3 contains the proposed method for predicting the commodity futures index, followed by our results, comparisons, and analysis in Sect. 4. Section 5 concludes the study and outlines some future work.

## 2 SVM for regression and the SVM–TLBO hybrid regression model

### 2.1 SVM for regression

Vapnik and his coworkers have developed an SVM technique for regression. The method was presented as follows.

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

The estimating function  $f$  is defined as

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

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

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

Here,  $L_{\in}$  is the extension of the  $\in$ -insensitive loss function originally proposed by Vapnik [41], which is defined as

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

By introducing the slack variables  $\zeta_i$  and  $\zeta'_i$ , the problem in Eq. (2) can be reformulated to the following.

$$(P) \text{ Minimize } C \left[ \sum_{i=1}^l (\zeta_i + \zeta'_i) \right] + \frac{1}{2} \|w\|^2 \text{ subject to}$$

$$\begin{aligned} y_i - w^T \Phi(x_i) - b &\leq \in + \zeta_i, \\ w^T \Phi(x_i) + b - y_i &\leq \in + \zeta'_i, \\ \zeta_i &\geq 0, \\ \zeta'_i &\geq 0, \end{aligned} \quad (4)$$

where  $i = 1, \dots, l$ , and  $C$  is a user-specified constant known as regularization parameter.

We can solve (P) using the primal dual method to get the following dual problem.

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

$$\begin{aligned} Q(\alpha_i, \alpha_i^*) &= \sum_{i=1}^l y_i (\alpha_i - \alpha_i^*) - \in \sum_{i=1}^l (\alpha_i - \alpha_i^*) \\ &\quad - \frac{1}{2} \sum_{i=1}^l \sum_{j=1}^l (\alpha_i - \alpha_i^*) (\alpha_j - \alpha_j^*) K(x_i, x_j), \end{aligned} \quad (5)$$

subject to

$$\sum_{i=1}^l (\alpha_i - \alpha_i^*) = 0, \quad (6)$$

and

$$0 \leq \alpha_i \leq C, \quad 0 \leq \alpha_i^* \leq C. \quad (7)$$

Here,  $i = 1, \dots, l$ , and  $K: X \times X \rightarrow R$  is the Mercer kernel defined by

$$K(x, z) = \Phi(x)^T \Phi(z). \quad (8)$$

The solution of the primal dual method yields

$$w = \sum_{i=1}^l (\alpha_j - \alpha_i^*) \Phi(x_i), \quad (9)$$

where  $b$  is calculated using the Karush–Kuhn–Tucker conditions. That is,

$$\begin{aligned} \alpha_i (\in + \zeta_i - y_i + w^T \Phi(x_i) + b) &= 0, \\ \alpha_i^* (\in + \zeta'_i + y_i - w^T \Phi(x_i) - b) &= 0, \end{aligned} \quad (10)$$

$$(C - \alpha_i) \zeta_i = 0 \text{ and } (C - \alpha_i^*) \zeta'_i = 0, \text{ where } i = 1, \dots, l. \quad (11)$$

Since  $\alpha_i \bullet \alpha_i^* = 0$  both  $\alpha_i$  and  $\alpha_i^*$  cannot be simultaneously non-zero, there exists some  $i$  for which either  $\alpha_i \in (0, C)$  or  $\alpha_i^* \in (0, C)$  and hence  $b$  can be computed using

$$b = y_i - \sum_{j=1}^l (\alpha_j - \alpha_j^*) K(x_j, x_i) - \in \text{ for } 0 < \alpha_i < C, \quad (12)$$

$$b = y_i - \sum_{j=1}^l (\alpha_j - \alpha_j^*) K(x_j, x_i) + \in \text{ for } 0 < \alpha_i^* < C.$$

The  $x_i$  corresponding to  $0 < \alpha_i < C$  and  $0 < \alpha_i^* < C$  are called *support vectors*. Using the expressions for  $w$  and  $b$  in Eqs. (9) and (12),  $f(x)$  can be computed using

$$\begin{aligned} f(x) &= \sum_{i=1}^n (\alpha_i - \alpha_i^*) (\Phi(x_i)^T \Phi(x)) + b, \\ &= \sum_{i=1}^l (\alpha_i - \alpha_i^*) K(x_i, x) + b. \end{aligned} \quad (13)$$

Note that we do not require function  $\Phi$  to compute  $f(x)$ , which is an advantage of using the kernel.

#### Advantages of SVM

SVMs have become a well-established tool within machine learning. Conceptually, they have many advantages, which include the following.

- The technique is methodical and derived from statistical learning theory.
- The SVM process requires convex function optimization, so there is a unique optimal solution (global minima).
- The model has an explicit strong dependence on a subset of the data points (support vectors), which improves model design.
- The relatively easy training process is a major strength of SVM.
- There are no local optima, unlike ANNs.

- f. The method scales moderately well to high-dimensional data, and the tradeoff between model complexity and errors can be explicitly controlled using appropriate optimal parameters.

#### Disadvantage of SVM

SVMs have the following disadvantage.

- The training time is roughly between a quadratic and cubic function of the number of samples in the training set.

## 2.2 Teaching–learning-based optimization technique

TLBO is a newly developed novel and effective meta-heuristic population based optimization algorithm [30]. It is similar to PSO, GAs, and ABC. TLBO is modeled on the transfer of knowledge within a classroom atmosphere, where learners (students) first acquire knowledge from a teacher (teacher phase) and then from their peers (student phase). The population in TLBO consists of a group of learners. There are decision variables, similar to other optimization algorithms. The different decision variables in TLBO are equivalent to the different subjects offered to students, and the students' grades are equivalent to the "fitness" in other population-based optimization methods. The flow chart for TLBO algorithm is presented in Fig. 1.

#### Salient features of TLBO

TLBOs have the following features.

- Similar to other population-based methods (e.g., GAs, PSO, and ABC), TLBO uses many results to proceed to the optimal solution.
- We do not need to tune any additional algorithm-specific controlling parameter.
- It uses the best solution of the current iteration to modify the existing solution in the population, which increases the convergence rate.
- The mean value of the population is used to update the solution.
- A good solution is accepted using a greediness approach.
- The population is not divided, unlike methods such as the ABC algorithm.

#### 2.2.1 Steps involved in the TLBO algorithm

The following steps of the TLBO algorithm were described by Rao et al. [30].

##### Step 1: Define the optimization problem and create a solution space

In the initial phase, we identify the decision variable(s) in the problem to be optimized and assign them a

range (minimum and maximum of the variable) where we will search for the optimal solution. If the solution spaces and ranges are not properly defined, then there is a chance that the optimization will take more time.

##### Step 2: Identify the fitness function

In this step, we design or identify the fitness function, which accurately represents how well the optimized solution fits our problem using a single number. The TLBO algorithm uses the fitness function to evaluate its candidate solutions and obtains the optimal solution by minimizing  $f(X)$ , where  $f(X)$  is the fitness function.

##### Step 3: Initializing learners (or students)

Each learner (based on the population size) is initialized using random values for each of the variables (within the appropriate ranges).

The  $i$ th learner is represented by row vector  $X_i$ , defined as

$$X_i = [x_{i,1}, x_{i,2}, x_{i,3}, \dots, x_{i,D}], \quad i = 1, 2, \dots, N, \quad (14)$$

where  $D$  is the number of decision variables, and  $N$  is the number of learners. Each decision variable  $x_{i,j}$  is randomly assigned a value using

$$x_{i,j} = x_j^{\min} + \text{rand}() * (x_j^{\max} - x_j^{\min}) \quad j = 1, 2, \dots, D, \quad (15)$$

where  $x_j^{\min}$  and  $x_j^{\max}$  are the minimum and maximum values of the  $j$ th variable of  $i$ th learner, and  $\text{rand}()$  is the random number function that returns a number between 0 and 1.

##### Step 4: Teacher phase

- (a) Compute the mean value of each of the learners' decision variables and denote the population mean as

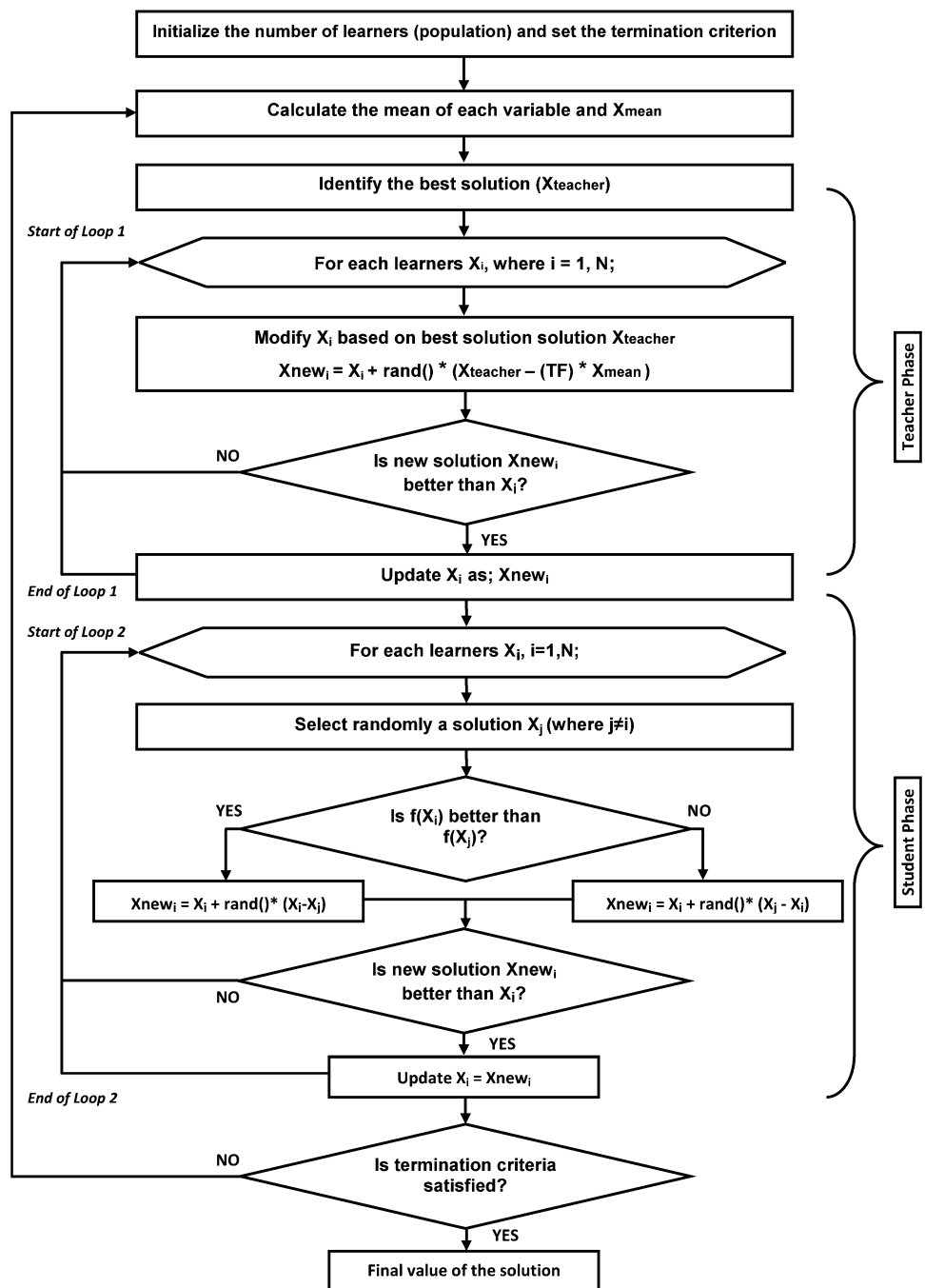
$$X_{\text{mean}} = [\bar{x}_1, \bar{x}_2, \dots, \bar{x}_j, \dots, \bar{x}_D], \quad \text{where } \bar{x}_j = \frac{\sum_{i=1}^N x_{i,j}}{N}.$$

- (b) Compute the fitness values of each learner  $X$  based on the fitness function  $f(X)$ . The learner with the best fitness value (solution) is identified as the teacher ( $X_{\text{teacher}}$ ) for the teacher phase.
- (c) Now the teacher ( $X_{\text{teacher}}$ ) transfers their knowledge and tries to improve the fitness of other learners ( $X_i$ ) using  $X_{\text{new}} = X_i + \text{rand}() * (X_{\text{teacher}} - (TF) * X_{\text{mean}})$ , for  $i = 1, 2, \dots, N$ , (10)

where

$$TF = \text{round}[1 + \text{rand}(0, 1)]. \quad (17)$$

Here,  $TF$  is the teaching factor (either 1 or 2), and  $\text{rand}()$  is the random number function that returns a number between 0 and 1.

**Fig. 1** Flow chart for the TLBO algorithm

Note that  $TF$  is not a parameter of the TLBO algorithm. The value of  $TF$  is not provided as input to the TLBO, but its value is randomly chosen by the algorithm using Eq. (17).

- (d) If the previously mentioned updated solution ( $X_{new}$ ) is better than the existing solution ( $X_i$ ), then we accept the new solution, otherwise we reject it.

*Step 5: Student phase*

In the student phase, the learners (students) enhance their knowledge by communicating with other learners in the classroom. Therefore, an individual learner learns if the other individuals have more knowledge.

- Randomly select any two solutions  $X_i$  and  $X_j$  such that  $i \neq j$ .
- If  $f(X_i)$ , that is, the fitness value of  $X_i$  is better than  $X_j$ , then we update  $X_i$  to  $X_{new}$  using

$$X_{new} = X_i + rand() * (X_i - X_j) \quad (18)$$

otherwise, we update it to

$$X_{new} = X_i + rand() * (X_j - X_i). \quad (19)$$

Here,  $rand()$  is the random number function that returns a number between 0 and 1.

*Step 6: Iterate until the termination criteria are satisfied*

We then repeat Steps 4 and 5 until our termination conditions are satisfied, i.e., the average value of the fitness function for all learners does not improve, or we reach the maximum number of generations. The  $X_i$  that minimizes  $f(X_i)$  is the final solution of the optimization problem.

### 2.3 SVM–TLBO hybrid regression model

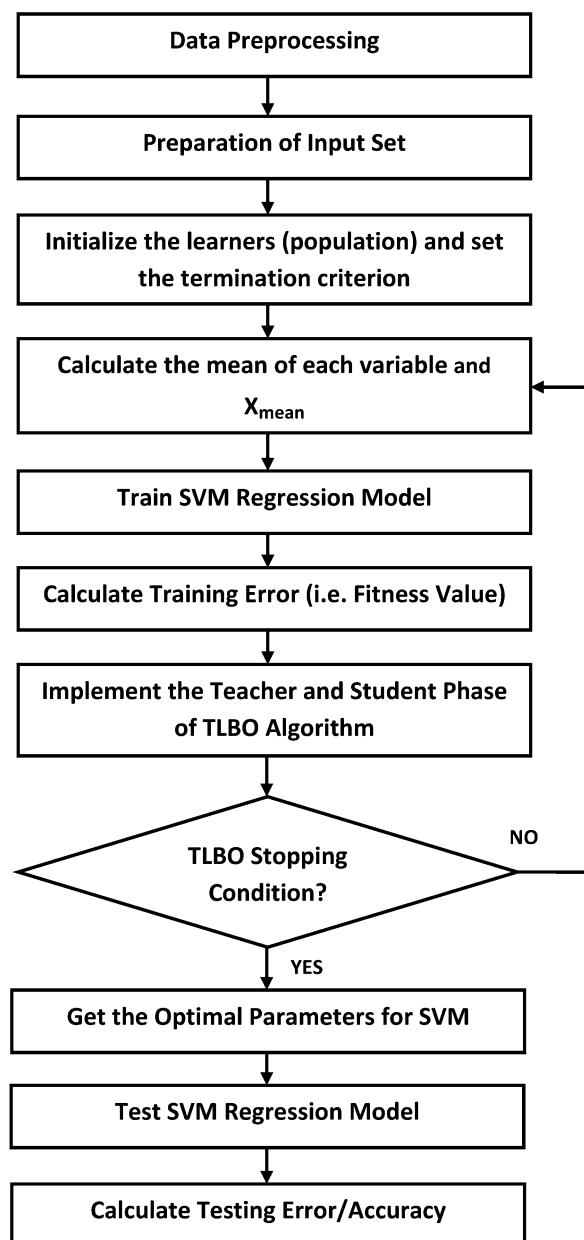
We propose a hybrid SVM–TLBO regression model, which uses SVM for predictions and TLBO for determining the SVM parameters. SVM can use many kernels, for example, linear, polynomial, sigmoid, wavelet, and Gaussian kernels. We have considered the Gaussian kernel (radial basis) function. This produces better financial time series forecasts [2, 38] because the data are complex and nonlinear. A SVM with a Gaussian kernel has three parameters that must be optimized. That is,  $C$  (regularization),  $\sigma$  (kernel width), and  $\varepsilon$  (insensitive loss function radius).

We designed the proposed SVM–TLBO hybrid regression model to work in a two-dimensional solution space, that is, to optimize  $C$  and  $\sigma$ . We keep the  $\varepsilon$  parameter constant at a reasonable value (i.e., 0.0001), because the number of support vectors decreases as  $\varepsilon$  increases, when  $\varepsilon$  is greater than 0.01 [2]. A flow chart of the SVM–TLBO hybrid regression model is presented in Fig. 2.

## 3 Proposed methodology

### 3.1 Dataset

We applied our forecasting model to real multicommodity futures index (MCX COMDEX) data collected from the MCX (<http://www.mcxindia.com>). MCX COMDEX is a collection of futures contracts on 15 physical commodities with a simple weighted average of three subindices (MCX AGRI, MCX METAL, and MCX ENERGY), which represent the key commodity sectors within the index. The index captures various sectors that incorporate futures contracts drawn on metals, energy, and agricultural commodities that are traded in the MCX. 1332 daily trading data points were collected from MCX COMDEX from January 1, 2010, to May 7, 2014. The time series data consist of daily open price, low price, high price, closing



**Fig. 2** Flow chart representation of the SVM–TLBO hybrid regression model

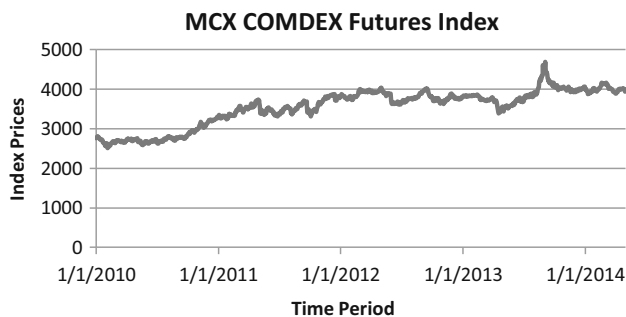
price, and traded date. The raw daily prices were used to calculate our financial technical indicators inputs. The time span covers many important and significant economic events, which we think are appropriate for training the models. Table 1 describes the data set in terms of high, low, mean, median, standard deviation, kurtosis (measure of flatness of the distribution), and skewness (degree of asymmetry of a distribution near its mean). The raw daily closing prices are plotted in Fig. 3. The data description in Table 1 and the plot in Fig. 3 clearly show that the data are well-spread. Therefore, an SVM trained with these data should be a well-generalized model.



**Table 1** Description of MCX COMDEX dataset

Parameter (for entire time span)	Value
High	4689.6
Low	2504.53
Mean	3539.561
Median	3692.17
Standard deviation	473.5277
Kurtosis	−0.4959
Skewness	−0.7378

### 3.2 Preprocessing of data

**Fig. 3** Closing prices of MCX COMDEX

We derived 17 financial technical indicators using the collected data, and used these indicators as input into the SVM regression model to forecast the closing price of the futures index. The technical indicators were computed using the formulas in Table 2. Financial technical indicators are a class of metrics whose values are derived from generic price activities in financial markets, and are extensively used by traders to predict the future price levels of a financial instrument by looking at past patterns. These financial technical indicators capture random price fluctuations in the market and offer a smoother perspective, because they are trend following or lagging indicators. The 17 financial technical indicators used in our study are based on previous work by Kim and Han [15], Kim [14], Kim and Lee [16], Tsang et al. [39], Ince and Trafalis [9], Huang and Tsai [8], Liang et al. [19], Lai et al. [17], and Chih-Ming [6], and from feedback from domain experts. The indicators are (1) 10-day moving average, (2) 20-day bias, (3) moving average convergence/divergence (MACD), (4) stochastic indicator %K, (5) stochastic indicator %D, (6) stochastic slow %D, (7) Larry William's %R, (8) rate of change (ROC), (9) relative strength index (RSI), (10) commodity channel index (CCI), (11) psychological line, (12) buying/selling momentum indicator, (13) buying/selling willingness indicator, (14) momentum, (15) disparity 5, (16) disparity 10, and (17) moving average

oscillators (MAO). After processing the 1332 raw data points, we obtained 1307 transformed data points with dates from February 1, 2010 to May 7, 2014. The 25 data points from January 1, 2010 to January 31, 2010, are not available because of the definitions of some technical indicators. For example, the buying/selling momentum and willingness indicators require 26 days of data.

We linearly normalized the technical indicators so they have a range of [0, 1]. This normalization procedure minimizes the forecasting errors and stops variables with larger numeric ranges from dominating those with smaller numeric ranges. We applied this to both the input technical indicators and the output closing prices. The technical indicators and closing prices were normalized using

$$Y_i = \frac{(P_i - P_{\min})}{(P_{\max} - P_{\min})}, \quad \text{for } i = 1, 2, 3, \dots, N, \quad (20)$$

where  $Y_i$  is the normalized value,  $P_i$  is the original value,  $P_{\min}$  and  $P_{\max}$  are the minimum and maximum values in the original data, and  $N$  is the total number of trading days.

The normalized data were segregated into training and test groups, approximately in the ratio of 5:1. Hence, 1085 data points were used for training with 5-fold cross-validation, and the remaining 222 were used to test the model. We considered three different forecasts of the closing prices: (1) 1 day ahead; (2) 3 days ahead; and (3) 5 days ahead.

In the 1-day-ahead forecasting case, the normalized technical indicators for each trading day from February 1, 2010 to April 30, 2014, and the normalized closing price for the next trading day (from February 2, 2010 to May 1, 2014, 1 day ahead) were partitioned into training and testing sets. The data were split up in a similar way for the 3 and 5-days-ahead forecasts.

### 3.3 Performance criteria

We evaluated the performance of the proposed model using standard statistical metrics: root mean square error (RMSE), normalized mean squared error (NMSE), mean absolute error (MAE), and directional symmetry (DS) [2, 38, 43]. Detailed descriptions and definitions of these performance criteria are given in Table 3. RMSE, MAE, and NMSE measure the deviation between the actual and forecasted futures index prices, so smaller values are preferred. The accuracy of the direction of the prediction is provided by DS (in %). Larger DS values indicate a better forecast.

### 3.4 Computation Techniques

We implemented Vapnik's SVM regression technique using LIBSVM, which is a SVM tool box [5]. SVMs for

**Table 2** Technical indicators (features)

Technical indicators (features) formulas

Notations:

 $i$ :  $i$ th day [ $i$  days ( $i = 1, 2, \dots, N$ ) counted from reference date, February 1, 2010, in the experiment] $HP_i$ : highest price of  $i$ th day $LP_i$ : lowest price of  $i$ th day $OP_i$ : open price of  $i$ th day $CP_i$ : closing price of  $i$ th day

Sl no.	Technical indicator name	Technical indicator description and formula
1	10-day moving average	The most current 10-day average closing price of the financial instrument. $MA_{10,i} = \frac{\sum_{j=i-9}^i CP_j}{10}$
2	20-day bias	Closing price and the moving average price deviation for the 20 days. $BIAS_{20,i} = \frac{CP_i - MA_{20,i}}{MA_{20,i}}, \text{ where } MA_{20,i} = \frac{\sum_{j=i-19}^i CP_j}{20}$
3	Moving average convergence/divergence (MACD)	MACD is the change between a 26-day and a 12-day exponential moving average (EMA). ** $MACD_i = EMA_{12,i} - EMA_{26,i}, \text{ where}$ $EMA_{N,i} = (CP_i - EMA_{N,i-1}) \times (2/(N+1)) + EMA_{N,i-1}$ **EMA gives more weight to recent prices and ever-decreasing weight to older data
4	Stochastic indicator %K	Stochastic %K compares where a security's price closed relative to its price range over a given period. (In this experiment, we used a period of 9 days.) $\%K_i = \frac{(CP_i - LLP)}{(HHP - LLP)} \times 100, \text{ where } LLP \text{ is the lowest low price and } HHP \text{ is the highest high price over the last } N \text{ periods}$
5	Stochastic indicator %D	Moving average of %K (three-period simple moving average) $\%D_i = \frac{\sum_{j=0}^2 \%K_{i-j}}{3}$
6	Stochastic slow %D	Moving average of %D (three-period simple moving average) $\%SD_i = \frac{\sum_{j=0}^2 \%D_{i-j}}{3}$
7	Larry William's %R	Larry William's %R is a momentum indicator that measures overbought/oversold levels. (In this experiment, we used a period of 9 days). $\%R_i = \frac{(HP - CP_i)}{(HP - LP)} \times 100$ where $LP$ is the lowest price and $HP$ is the highest price over the last $N$ periods
8	Rate of change (ROC)	Ratio of current closing price to the price a certain number of periods ( $n$ period) ago. (In this experiment, we used a period of 10 days). $ROC_i = \frac{CP_i}{CP_{i-n}} \times 100$ where $CP_{i-n}$ is the Closing Price of ( $i - n$ )th day
9	Relative strength index (RSI)	RSI is a momentum oscillator that compares the magnitude of recent gains to the magnitude of recent losses. (In this experiment, we used a period of 5 days). $RSI_i = \frac{AG_i}{AG_i + AL_i} \times 100,$ where $G_i = \begin{cases} CP_{i-1} - CP_i, & \text{if } CP_i > CP_{i-1} \\ 0 & \text{and} \end{cases}$ $L_i = \begin{cases} CP_{i-1} - CP_i, & \text{if } CP_i < CP_{i-1} \\ 0 & \end{cases}$ $AG_i = \frac{4}{5} \times AG_{i-1} + \frac{1}{5} \times G_i \text{ and } AL_i = \frac{4}{5} \times AL_{i-1} + \frac{1}{5} \times L_i$



**Table 2** continued

Sl no.	Technical indicator name	Technical indicator description and formula
10	Commodity channel index (CCI)	<p>CCI measures the variation of a security's price from its statistical mean. (In this experiment, we used a period of 24 days).</p> $CCI_i = \frac{TP_i - MATP_i}{0.015 \times MD_i}$ <p>where <math>TP_i = \frac{HP_i + LP_i + CP_i}{3}</math>,</p> $MATP_i = \frac{\sum_{j=i-23}^i TP_j}{24}$ $MD_i = \frac{\sum_{j=i-23}^i TP_j - MATP_j}{24}$ <p>where <math>TP_i</math> is the typical price for the <math>i</math>th day, <math>MATP_i</math> is the 24-day simple moving average of the typical price for the <math>i</math>th day, and <math>MD_i</math> is the 24-day mean deviation for the <math>i</math>th day</p>
11	Psychological line	<p>Psychological line is the volatility indicator based on the number of time intervals that the market was rising during the preceding period. (In this experiment, we used a period of 13 days).</p> $PSY_i = \frac{TDU_i}{13} \times 100\%$ , where $TDU_i$ is the total number of days with regard to the rise in stock price in the previous 13 days
12	Buying/selling momentum indicator	<p>Buying/selling momentum indicator (26 days)</p> $BSMI_i = \frac{\sum_{j=i-25}^i (HP_j - OP_j)}{\sum_{j=i-25}^i (OP_j - LP_j)}$
13	Buying/selling willingness indicator	<p>Buying/selling willingness indicator (26 days)</p> $BSWI_i = \frac{\sum_{j=i-25}^i (HP_j - CP_{j-1})}{\sum_{j=i-25}^i (CP_{j-1} - LP_j)}$
14	Momentum	<p>Momentum measures the amount that a security's price has changed over a given period (4 days),</p> $MO_i = CP_i - CP_{i-4}$
15	Disparity 5	<p>Measures the distance between the current price and the moving average over 5 days,</p> $DIS_{5,i} = \frac{CP_i}{MA_{5,i}}$ <p>where <math>MA_{5,i}</math> is the 5-day moving average for the <math>i</math>th day</p>
16	Disparity 10	<p>Measures the distance between the current price and the moving average over 10 days,</p> $DIS_{10,i} = \frac{CP_i}{MA_{10,i}}$ <p>where <math>MA_{10,i}</math> is the 10-day moving average for the <math>i</math>th day</p>
17	Moving average oscillators (MAO)	<p>Price oscillator that displays the difference between two moving averages of different lengths (5 and 10 days),</p> $MAO_i = \frac{MA_{5,i} - MA_{10,i}}{MA_{5,i}}$ <p>where <math>MA_{5,i}</math> and <math>MA_{10,i}</math> are the 5- and 10-day moving averages for the <math>i</math>th day</p>

financial time series forecasting commonly use the polynomial kernel ( $k(x, y) = (x \cdot y + 1)^d$ ) or the Gaussian kernel ( $k(x, y) = \exp\left(-1/\sigma^2 \|x - y\|^2\right)$ ).  $d$  is the degree of the polynomial kernel and  $\sigma^2$  is the width (bandwidth) of the Gaussian kernel. We used the Gaussian kernel (radial basis) function, because it performs well under general smoothness assumptions. Additionally, the Gaussian kernel has fewer parameters than the polynomial kernel. The

polynomial kernel produces inferior results when compared with the Gaussian kernel, and requires more training time [2, 21, 38, 43]. We used an Intel Core i7 CPU, 4 GB memory PC for our simulations.

Traditional procedures for optimizing the parameters of the SVM model and the kernel function use grid search [13] or cross-validation [10] methods. However, both of these methods are computationally expensive and data intensive [12]. Grid search is a local search technique that

**Table 3** Performance metrics

Performance metrics	Definition
RMSE	$\sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - p_i)^2}$
NMSE	$\frac{1}{\sigma^2 n} \sum_{i=1}^n (y_i - p_i)^2$ where : $\sigma^2 = \frac{1}{n-1} \sum_{i=1}^n (y_i - \bar{y})^2$ $\bar{y} = \sum_{i=1}^n y_i$
MAE	$\frac{1}{n} \sum_{i=1}^n  y_i - p_i $
DS	$\frac{100}{n} \sum_{i=1}^n d_i$ $d_i = \begin{cases} 1 & \text{if } (y_i - y_{i-1})(p_i - p_{i-1}) \geq 0 \\ 0 & \text{otherwise} \end{cases}$

$n$  is the total number of data,  $y_i$  is the actual output value, and  $p_i$  is the predicted output value of the  $i$ th sample data

**Table 4** (a) SVM and (b) TLBO parameters used in our experiments

(a) SVM Parameters	Parameter Value
$C$ (regularization parameter)	0.01 to 35,000
$\sigma^2$ (bandwidth)	0.0001 to 32
$\varepsilon$	0.0001 (Fixed)
Kernel type	Radial basis (Gaussian)
(b) TLBO parameters	Parameter value
TLBO parameter	
Population size (learners/students)	15
MAX iteration	30
Optimization category	Minimum

often becomes trapped in local optima, and it is sometime hard to determine its search interval [21]. In this study, we used grid search to find the best values for  $C$  and  $\sigma^2$  using cross-validation. We considered different pairs of  $(C, \sigma^2)$  and then selected those that minimized the error, which we then used in our comparisons. In the simulation experiment, we used  $C$  values in the range 0.01 to 35,000, and  $\sigma^2$  values between 0.0001 and 32 (Table 4). After determining the final  $(C$  and  $\sigma^2)$  values for all three forecasting cases (i.e., 1, 3, and 5 days ahead), we trained the model again to generate the final forecasting model. The index prices obtained for the standard SVM regression model are shown in Figs. 4a–c.

We wrote our own code to implement TLBO for the proposed SVM-TLBO hybrid regression model. The TLBO

algorithm was defined in two dimensions, to optimize  $\sigma^2$  (bandwidth) of the Gaussian kernel parameter and  $C$  (regularization parameter) of the SVMs. In our experimental runs of the TLBO algorithm, there were no significant changes to  $\sigma^2$  and  $C$  after 25–30 iterations, when using a population size (learners/students) of 15. Rao and Patel [29], Pawar and Rao [26], Rao et al. [28], and Rao and Waghmare [33] also observed that TLBO only requires a small population and few iterations (generations). With this in mind, we fixed the maximum number of iterations for the TLBO to 30, with a population size 15 (Table 4). We observed that the value of our objective function decreased when the algorithm went from the teacher to student phases within the same iteration, and reduced with the number of iterations. Similar observations were made by Rao et al. [31]. When defining the solution space for TLBO, the range of  $C$  was set to 0.01–35,000, and the range of  $\sigma^2$  was set to 0.0001–32 [20]. The hybrid regression model algorithm ran as per the flow chart provided in Fig. 2, and the simulation results are shown in Fig. 4 and Table 8. In these results, the  $k$ th test day means the  $(1085 + k)$ th day from our reference date (February 1, 2010), because we have taken the first 1085 days of data for training, and used the remaining 222 days for testing. We compared the results of our proposed SVM–TLBO hybrid regression approach with standard SVM regression and the PSO + SVM model of Lin et al. [21]. We used a sequential optimization (SMO)-based algorithm to train the SVM regression, because it is fast and efficient for large data sets.

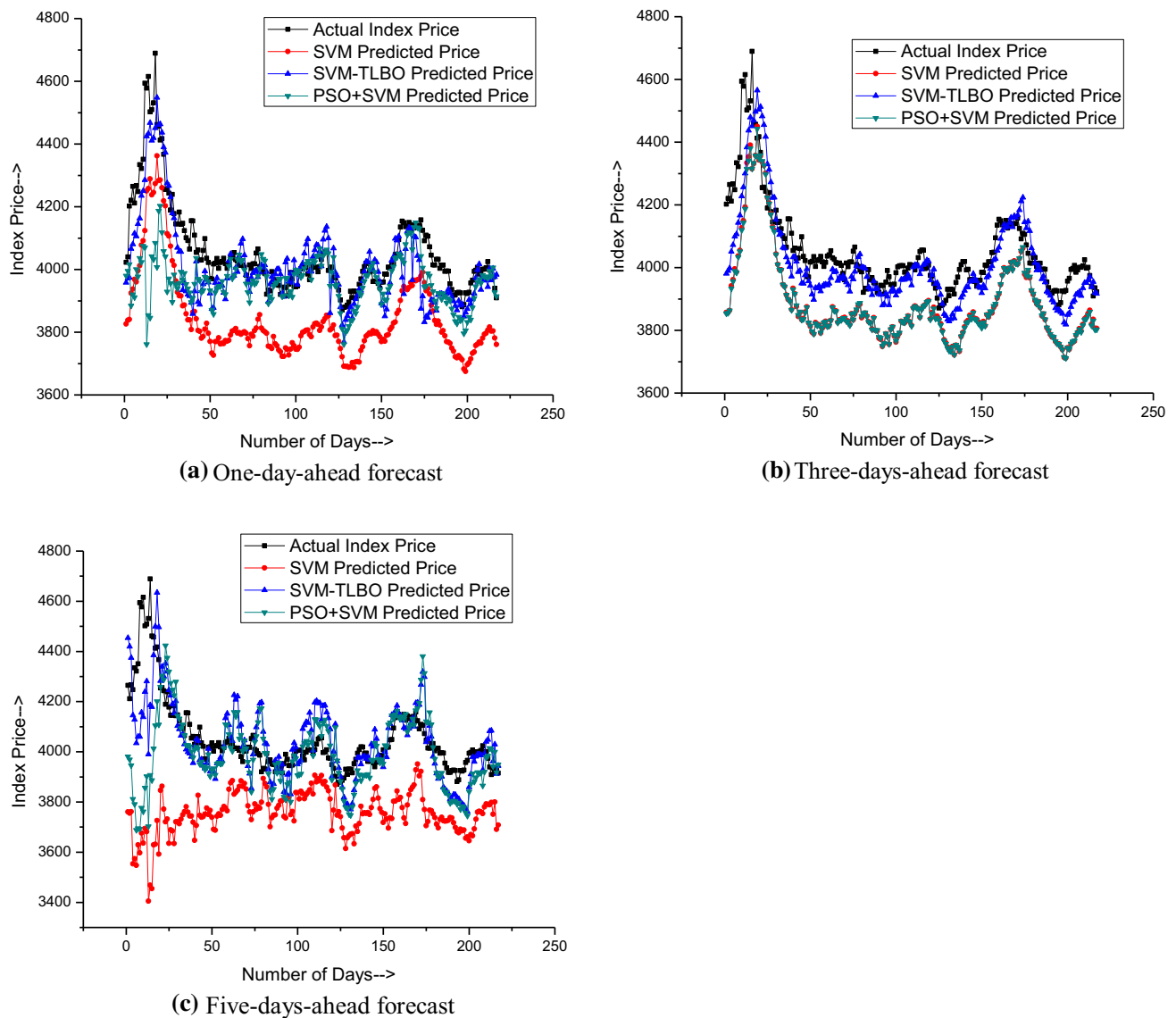
## 4 Results and discussion

The RMSE results of the SVM regression model in the training and testing phases, and the final values of  $C$  and  $\sigma^2$  are presented in Table 5 for all three forecasting cases.

The results for the proposed SVM–TLBO hybrid regression model, and the optimal parameters are summarized in Table 6.

### 4.1 Comparisons of results

The RMSE, MAE, and NMSE values presented in Table 7 show that the SVM–TLBO hybrid regression model outperformed the standard SVM regression and PSO + SVM hybrid approaches in all three forecasting cases. With regard to the DS performance metric, SVM–TLBO performed better than the standard SVM and PSO + SVM models in two forecasting cases (3 and 5 days ahead), but standard SVM performed better for the 1-day-ahead forecast. Financial market practitioners evaluate forecasting



**Fig. 4** Actual and predicted futures index prices using the SVM regression, PSO + SVM hybrid model, and SVM-TLBO hybrid regression models, for the **a** 1-day-ahead forecast, **b** 3-days-ahead forecast, and **c** 5-days-ahead forecast

**Table 5** Model performance and final parameter settings using the standard SVM regression model

Forecasting cases	Final value		Training error RMSE	Testing error RMSE
	$C$	$\sigma^2$		
1-day-ahead forecast	100	0.01	0.0601	0.0992
3-days-ahead forecast	10000	0.0001	0.0614	0.0922
5-days-ahead forecast	0.1	1	0.0613	0.1403

**Table 6** Model performance and optimal parameters achieved by proposed SVM-TLBO hybrid regression model

Forecasting cases	Optimal value		Training error RMSE	Testing error RMSE
	$C$	$\sigma^2$		
1-day-ahead forecast	2376.1150	2.1725	0.0363	0.0440
3-days-ahead forecast	25.5730	0.0234	0.0333	0.0408
5-days-ahead forecast	198.8539	0.1818	0.0434	0.0599

models using both minimum forecast error and directional accuracy [18]. The aim is to get a directional accuracy of over 50 % [43]. In our study, the DS values for the SVM–TLBO hybrid and standard SVM methods were greater than 50 % in all cases. The DS values for the PSO + SVM hybrid approach was greater than 50 % for the 1-day-ahead and 3-days-ahead forecasts, but it was 48.15 % for the 5-days-ahead forecasts. The number in bold is the best performance.

Figure 4 shows the actual futures index prices, and the prices predicted using standard SVM regression, the PSO + SVM hybrid model, and the proposed SVM–TLBO hybrid regression model for the three types of forecasts. Table 8 represents the forecasting results in terms of index prices for a few data samples using the standard SVM, PSO + SVM hybrid, and SVM–TLBO hybrid regression model. It can be clearly seen from Table 8 that the index prices from the proposed SVM–TLBO hybrid model were more accurate than standard SVM, and were much better than the PSO + SVM hybrid model.

## 5 Conclusions and future work

In this research, we examined the feasibility of applying the newly developed novel TLBO algorithm to select optimal free parameters for an SVM regression model of financial time-series data. We used multicommodity futures index data collected from MCX. Our experimental results show that our proposed SVM–TLBO hybrid regression model effectively found the optimal parameters, and produced better predictions than the standard SVM method. The proposed model improved the MAE result by 65.87 % (for the 1-day-ahead forecast), 55.83 % (for the 3-days-ahead forecast), and 67.03 % (for the 5-days-ahead forecast), when compared with standard SVM regression. The proposed model also improved the RMSE result by 55.64 % (1 day ahead), 55.74 % (3 days ahead), and 57.3 % (for 5 days ahead), when compared with standard SVM regression. There were similar improvements in terms of MAE and RMSE when we compared the proposed SVM–TLBO hybrid regression method with the PSO + SVM hybrid model. Moreover, our experiments demonstrate that the proposed SVM–TLBO hybrid regression model is more efficient than the standard SVM and PSO + SVM hybrid models for financial time series forecasting. The proposed model avoids user-specified control parameters, which are required when using optimization methods such as PSO, GAs, and ACO.

In our current model, we selected the technical indicators (features) using previous research in this area and expert feedback. We could enhance the accuracy of the forecast by including efficient macroeconomic features.

**Table 7** Comparison of the results of the standard SVM, PSO + SVM hybrid, and SVM–TLBO hybrid regression models

Forecasting Cases	RMSE			MAE			NMSE			DS		
	SVM	PSO + SVM	SVM–TLBO	SVM	PSO + SVM	SVM–TLBO	SVM	PSO + SVM	SVM–TLBO	SVM	PSO + SVM	SVM–TLBO
1-day-ahead forecast	0.0992	0.074	<b>0.0440</b>	0.0967	0.046	<b>0.0330</b>	4.32E – 05	2.39E – 05	<b>8.49E – 06</b>	<b>58.79</b>	58.33	56.02
3-days-ahead forecast	0.0922	0.0825	<b>0.0408</b>	0.0754	0.0756	<b>0.0333</b>	2.98E – 05	3.07E – 05	<b>7.37E – 06</b>	51.54	50.0	<b>52.85</b>
5-days-ahead forecast	0.1403	0.0901	<b>0.0599</b>	0.1341	0.0547	<b>0.0442</b>	1.13E – 04	5.27E – 05	<b>1.58E – 05</b>	50.93	48.15	<b>52.53</b>

Bold values represent the best performance

**Table 8** Forecasting results using the SVM regression, SVM-TLBO hybrid regression, and PSO + SVM hybrid models

Test day	Actual index price	SVM predicted price	SVM–TLBO predicted price (Proposed model)	PSO + SVM predicted price (Model proposed by Lin et al. [21])
1-day-ahead forecast case				
1	4022.8	3825.91	3958.05	3980.30
20	4456.91	4283.19	4461.05	4187.16
40	4155.58	3808.17	3894.42	3895.84
60	4040.45	3803.68	3975.80	3970.81
80	4004.05	3813.38	4013.54	4048.85
100	3944.44	3745.54	3995.06	3999.12
120	3973.19	3789.21	3990.55	3947.49
140	4008.8	3772.55	3993.99	3962.22
160	4094.73	3869.55	4106.00	4049.30
180	4013.89	3800.26	3922.44	3904.81
200	3922.77	3695.89	3874.01	3832.20
215	3974.47	3804.05	4003.03	4006.06
3-days-ahead forecast case				
1	4202.29	3856.72	3980.66	3853.62
20	4456.91	4357.65	4496.56	4356.84
40	4054.45	3934.26	4032.90	3930.90
60	4036.33	3812.34	3943.46	3812.67
80	3986.59	3839.92	4001.38	3845.25
100	3983.33	3762.45	3911.36	3766.66
120	3975.41	3835.40	3959.49	3834.11
140	4019.34	3810.88	3926.60	3808.09
160	4153.92	3915.55	4064.03	3918.23
180	4032.41	3914.36	4044.75	3917.08
200	3957.01	3744.60	3849.69	3744.27
215	3909.04	3835.386	3954.89	3831.10
5-days-ahead forecast case				
1	4264.49	3760.65	4453.95	3981.29
20	4367.21	3592.28	4496.83	4108.07
40	4029.58	3647.12	4019.41	4009.60
60	4025.99	3884.69	4043.30	4002.39
80	3920.69	3799.40	4197.11	4173.62
100	4005.5	3837.05	3950.66	3906.84
120	3948.8	3685.89	3900.63	3888.38
140	3994.8	3754.86	3973.07	3907.38
160	4149.41	3810.52	4153.90	4138.01
180	4032.41	3765.33	4059.89	4109.45
200	3994.14	3645.71	3858.94	3841.44
215	3933.27	3691.15	3912.75	3917.35

The proposed model can also be applied to other domains in the future, to validate and extend the model.

**Acknowledgments** We would like to express our gratitude to the National Institute of Science and Technology (NIST), for the facilities and resources provided at the Data Science Laboratory at NIST for the development of this study. The authors would also like to thank the editor and the anonymous reviewers for their innovative suggestions that improved the quality of this manuscript.

## References

1. Bose S (2008) Commodity futures market in India: a study of trends in the national multi-commodity indices. *ICRA Bull Money Finance* 3(3):125–158
2. Cao LJ, Tay FEH (2003) Support vector machine with adaptive parameters in financial time series forecasting. *IEEE Trans Neural Networks* 14(6):1506–1518

3. Chauhan P, Pant M, Deep K (2015) Parameter optimization of multi-pass turning using chaotic PSO. *Int J Mach Learn Cybern* 6:319–337. doi:[10.1007/s13042-013-0221-1](https://doi.org/10.1007/s13042-013-0221-1)
4. Cherkassy V, Ma Y (2004) Practical selection of SVM parameters and noise estimation for SVM regression. *Neural Netw* 17(1):113–126
5. Chih-Chung C, Chin-Jen (2001) LIBSVM: a library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27. (Software available at <http://www.csie.ntu.edu.tw/~cjlin/libsvmLIBSVM>)
6. Chih-Ming H (2013) A hybrid procedure with feature selection for resolving stock/futures price forecasting problems. *Neural Comput Appl* 2013:651–671. doi:[10.1007/s00521-011-07214](https://doi.org/10.1007/s00521-011-07214)
7. Gestel TV, Suykens JAK, Baestaens D-E, Lambrechts A, Lanckriet G, Vandaele B, Moor BD, Vandewalle J (2001) Financial time-series prediction using least squares support vector machines within the evidence framework. *IEEE Trans Neural Netw* 12(4):809–821
8. Huang CL, Tsai CY (2009) A hybrid SOFM-SVR with a filter based feature selection for stock market forecasting. *Expert Syst Appl* 36(2):1529–1539. doi:[10.1016/j.eswa.2007.11.062](https://doi.org/10.1016/j.eswa.2007.11.062)
9. Ince H, Trafalis TB (2008) Short term forecasting with support vector machines and application to stock price prediction. *Int J Gen Syst* 37(6):677–687. doi:[10.1080/03081070601068595](https://doi.org/10.1080/03081070601068595)
10. Ito K, Nakano R (2005) Optimizing support vector regression hyper-parameters based on cross-validation. *Proc Int Jt Conf Neural Netw* 3:871–876
11. Jain SK, Patnaik A, Sinha SN (2013) Design of custom-made stacked patch antennas: a machine learning approach. *Int J Mach Learn Cybern* 4:189–194. doi:[10.1007/s13042-012-0084-x](https://doi.org/10.1007/s13042-012-0084-x)
12. Jiang M, Jiang S, Zhu L, Wang Y, Huang W, Zhang H (2013) Study on parameter optimization for support vector regression in solving the inverse ECG problem. *Comput Math Methods Med*. doi:[10.1155/2013/158056](https://doi.org/10.1155/2013/158056) Article ID 158059
13. Keerthi SS (2002) Efficient tuning of SVM hyperparameters using radius/margin bound and iterative algorithms. *IEEE Trans Neural Netw* 13(5):1225–1229
14. Kim K (2003) Financial time series forecasting using support vector machines. *Neurocomputing* 55:307–319
15. Kim KJ, Han I (2000) Genetic algorithms approach to feature discretization in artificial neural networks for the prediction of stock price index. *Expert Syst Appl* 19(2):125–132
16. Kim KJ, Lee WB (2004) Stock market prediction using artificial neural networks with optimal feature transformation. *Neural Comput Appl* 13(3):255–260. doi:[10.1007/s00521-004-0428-x](https://doi.org/10.1007/s00521-004-0428-x)
17. Lai RK, Fan CY, Huang WH, Chang PC (2009) Evolving and clustering fuzzy decision tree for financial time series data forecasting. *Expert Syst Appl* 36(2):3761–3773. doi:[10.1016/j.eswa.2008.02.025](https://doi.org/10.1016/j.eswa.2008.02.025)
18. Leung MT, Daouk H, Chen AS (2000) Forecasting stock indices: a comparison of classification and level estimation models. *Int J Forecast* 16:173–190
19. Liang X, Zhang HS, Mao JG, Chen Y (2009) Improving option price forecasts with neural networks and support vector regressions. *Neurocomputing* 72(13–15):3055–3065. doi:[10.1016/j.neucom.2009.03.015](https://doi.org/10.1016/j.neucom.2009.03.015)
20. Lin H-T, Lin C-J (2003) A study on sigmoid kernels for SVM and the training of non-PSD kernels by SMO-type methods. Technical report, University of National Taiwan, Department of Computer Science and Information Engineering, March, pp 1–32
21. Lin SW, Ying K-C, Chen S-C, Lee Z-J (2008) Particle swarm optimization for parameter determination and feature selection of support vector machines. *Expert Syst Appl* 35:1817–1824
22. Liu S, Tian L, Huang Y (2014) A comparative study on prediction of throughput in coal ports among three models. *Int J Mach Learn Cybern* 5:125–133. doi:[10.1007/s13042-013-0201-5](https://doi.org/10.1007/s13042-013-0201-5)
23. Liu Z, Wu Q, Zhang Y, Philip Chen CL (2011) Adaptive least squares support vector machines filter for hand tremor canceling in microsurgery. *Int J Mach Learn Cybern* 2:37–47. doi:[10.1007/s13042-011-0012-5](https://doi.org/10.1007/s13042-011-0012-5)
24. Musa AB (2013) Comparative study on classification performance between support vector machine and logistic regression. *Int J Mach Learn Cybern* 4:13–24. doi:[10.1007/s13042-012-0068-x](https://doi.org/10.1007/s13042-012-0068-x)
25. Nicholas IS, Ravi S (2009) Time series prediction using support vector machines: a survey. *IEEE Comput Intel Mag* 4(2):24–38. doi:[10.1109/MCI.2009.932254](https://doi.org/10.1109/MCI.2009.932254)
26. Pawar PV, Rao RV (2013) Parameter optimization of machining using teaching-learning-based optimization algorithm. *Int J Adv Manuf Technol* 67:995–1006
27. Rao RV, Kalyankar VD (2012) Parameter optimization of machining processes using a new optimization algorithm. *Mater Manuf Process* 27(9):978–985
28. Rao RV, Kalyankar VD, Waghmare G (2014) Parameters optimization of selected casting processes using teaching-learning-based optimization algorithm. *Appl Math Model* 38:5592–5608. doi:[10.1016/j.apm.2014.04.036](https://doi.org/10.1016/j.apm.2014.04.036)
29. Rao RV, Patel V (2014) A multi-objective improved teaching-learning based optimization algorithm for unconstrained and constrained optimization problems. *Int J Ind Eng Comput* 5:1–22. doi:[10.5267/j.ijiec.2013.09.007](https://doi.org/10.5267/j.ijiec.2013.09.007)
30. Rao RV, Savsani VJ, Vakharia DP (2011) Teaching-learning-based optimization: a novel method for constrained mechanical design optimization problems. *Comput Aided Des* 43(3):303–315
31. Rao RV, Savsani VJ, Vakharia DP (2012) Teaching-learning-based optimization: a optimization method for continuous non-linear large scale problems. *Inf Sci* 183:1–15
32. Rao RV, Waghmare GG (2014) A comparative study of a teaching-learning-based optimization algorithm on multi-objective unconstrained and constrained functions. *J King Saud Univ Comput Inf Sci* 26(3):332–346. doi:[10.1016/j.jksuci.2013.12.004](https://doi.org/10.1016/j.jksuci.2013.12.004)
33. Rao RV, Waghmare GG (2015) Multi-objective design optimization of a plate-fin heat sink using a teaching-learning-based optimization algorithm. *Appl Therm Eng* 76:521–529. doi:[10.1016/j.applthermaleng.2014.11.052](https://doi.org/10.1016/j.applthermaleng.2014.11.052)
34. Refenes AP, Zaprantis AD, Francis G (1995) Modeling stock returns in the framework of APT: a comparative study with regression models. In: Refenes AP (ed) *Neural Networks in the Capital Markets*. Wiley, Chichester, pp 101–125
35. Satapathy SC, Naik A, Parvathi A (2013) A teaching learning based optimization based on orthogonal design for solving global optimization problems. *SpringerPlus* 2013(2):130. doi:[10.1186/2193-1801-2-130](https://doi.org/10.1186/2193-1801-2-130)
36. Shazly MRE, Shazly HEE (1999) Forecasting currency prices using genetically evolved neural network architecture. *Int Rev Financ Anal* 8(1):67–82
37. Steiner M, Wittkemper HG (1995) Neural networks as an alternative stock market model. In: Refenes AP (ed) *Neural Networks in the Capital Markets*. Wiley, Chichester, pp 135–147
38. Tay FEH, Cao L (2002) Modified support vector machines in financial time series forecasting. *Neurocomputing* 48:847–861
39. Tsang PM, Kwok P, Choy SO, Kwan R, Ng SC, Mak J, Tsang J, Koong K, Wong TL (2007) Design and implementation of NN5 for Hong Kong stock price forecasting. *Eng Appl Artif Intell* 20(4):453–461. doi:[10.1016/j.engappai.2006.10.002](https://doi.org/10.1016/j.engappai.2006.10.002)
40. Tsiouris G, Zeidenberg M (1995) Testing the efficient markets hypothesis with gradient descent algorithms. *Neural Networks in the Capital Markets*, pp 127–136
41. Vapnik V (1995) *The Nature of Statistical Learning Theory*. Springer, New York (ISBN 0-387-94559-8)
42. Wittkemper HG, Steiner M (1996) Using neural networks to forecast the systematic risk of stocks. *Eur J Oper Res* 90:577–588



43. Wun-Hua C, Jen-Ying S, Soushan W (2006) Comparison of support vector machines and back propagation neural networks in forecasting the six major Asian stock markets. *Int J Electron Finance* 1(1):49–67
44. Yang Y, Wang G, Yang Y (2014) Parameters optimization of polygonal fuzzy neural networks based on GA-BP hybrid algorithm. *Int J Mach Learn Cybern* 5:815–822. doi:[10.1007/s13042-013-0224-y](https://doi.org/10.1007/s13042-013-0224-y)
45. Zou F, Wang L, Hei X, Chen D, Yang D (2014) Teaching-learning-based optimization with dynamic group strategy for global optimization. *Inf Sci* 273:112–131. doi:[10.1016/j.ins.2014.03.038](https://doi.org/10.1016/j.ins.2014.03.038)