

RESEARCH ARTICLE

Financial time series forecasting using twin support vector regression

Deepak Gupta¹, Mahardhika Pratama^{2*}, Zhenyuan Ma³, Jun Li⁴, Mukesh Prasad⁴

1 Department of Electronics and Computer Engineering, National Institute of Technology, Arunachal Pradesh, India, **2** School of Computer Science and Engineering, Nanyang Technological University, Singapore, Singapore, **3** School of Mathematics and System Sciences, Guangdong Polytechnic Normal University, Guangzhou, China, **4** Centre for Artificial Intelligence, School of Software, Faculty of Engineering and Technology, University of Technology Sydney, Sydney, Australia

* mpratama@ntu.edu.sg

Abstract

Financial time series forecasting is a crucial measure for improving and making more robust financial decisions throughout the world. Noisy data and non-stationarity information are the two key factors in financial time series prediction. This paper proposes twin support vector regression for financial time series prediction to deal with noisy data and nonstationary information. Various interesting financial time series datasets across a wide range of industries, such as information technology, the stock market, the banking sector, and the oil and petroleum sector, are used for numerical experiments. Further, to test the accuracy of the prediction of the time series, the root mean squared error and the standard deviation are computed, which clearly indicate the usefulness and applicability of the proposed method. The twin support vector regression is computationally faster than other standard support vector regression on the given 44 datasets.

OPEN ACCESS

Citation: Gupta D, Pratama M, Ma Z, Li J, Prasad M (2019) Financial time series forecasting using twin support vector regression. PLoS ONE 14(3): e0211402. <https://doi.org/10.1371/journal.pone.0211402>

Editor: Francisco Martínez-Álvarez, Pablo de Olavide University, SPAIN

Received: December 5, 2017

Accepted: December 21, 2018

Published: March 13, 2019

Copyright: © 2019 Gupta et al. This is an open access article distributed under the terms of the [Creative Commons Attribution License](https://creativecommons.org/licenses/by/4.0/), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

Data Availability Statement: All relevant data are available from <https://finance.yahoo.com>. To search for a data set, type the target company name in the search box, click the "Historical Data" tab, and select the desired time period. Then, set "Show" to "Historical Process" and "Frequency" to "Daily". Once the data set loads, you may hit "Apply" and download and utilize the data.

Funding: This work was supported in part by the Science and Engineering Research Board, Gov. of India (SERB) under the early career research award ECR/2016/001464 and Science and Technology

Introduction

For the last two decades in the machine learning area, support vector machines (SVMs) have been a computationally powerful kernel-based tool for various classification problems, such as pattern recognition and regression problems and function approximations [1]. It has the advantages over other methods, such as artificial neural networks (ANN), which focus on minimizing the empirical risk in the training phase, whereas SVM was developed on the structural risk minimization principle [1], which minimizes the upper bound on the generalization error. Another advantage of SVM is that it forms a convex optimization problem, a single large quadratic programming problem (QPP) that yields a unique global solution. The SVM has been applied in many fields to solve various well-known real-world problems ranging from image classification [2], remote sensing image classification [3], text characterization [4], biomedicine [5, 6], time series prediction [7, 8] and business prediction [9], which clearly justify its popularity.

To obtain an optimal regressor function for a given set of training data, support vector regression (SVR) was introduced by Vapnik [1], where training data points are in the input

Program of Guangzhou (No. 201704030133), “A Knowledge-Connection and Cognitive- Style based Mining System for Massive Open Online Courses and Its Application” (Code: PRO16-1300).

Competing interests: The authors have declared that no competing interests exist.

space or in a higher dimensional space via kernel mapping. The SVR has the advantage of better generalization performance than the other regression methods. However, standard SVM has a drawback in that it optimizes a computationally expensive cost function for large-scale datasets that have high training costs, i.e., $O(m^3)$, where m is the number of training samples. Due to this high training cost, it is not easy to find the optimal parameters from a large set of parameters. To address this issue, different variants of SVM have been proposed, such as chunking and decomposition methods [10, 11], exact SVM training algorithm SMO [12], approximate SVM training algorithms [13–15] and LS-SVM [16, 17].

Mangasarian and Wild [18] suggested a new method for binary classification as a generalized eigenvalue proximal support vector machine (GEPSVM) based on two nonparallel hyperplanes. To find the nonparallel hyperplanes, GEPSVM solves two eigenvalue problems based on the size of the input space dimensions. The GEPSVM outperforms the standard SVM in terms of computational speed and accuracy. Similarly, in the spirit of GEPSVM, twin support vector machine (TWSVM) has recently been proposed [19] for binary classification problems that consist of two nonparallel planes, for example, where each plane is closer to the data points of one of the two classes and as far as possible from the data points of the other class. In TWSVM, two QPPs of smaller size are solved to obtain two nonparallel hyperplanes instead of a QPP of large size. This strategy gives TWSVM good generalization ability, making it better than GEPSVM and approximately four times faster than the standard SVM. The main difference between GEPSVM and TWSVM is that GEPSVM solves two generalized eigenvalue problems to obtain the hyperplanes because TWSVM solves two related SVM-type problems to obtain the hyperplanes. Peng [20] recently proposed a twin support vector regression technique based on TWSVM in which an unknown regressor function is generated by the construction of nonparallel insensitive up and down bound functions. In this case, it solves a pair of two smaller sized QPPs unlike the large QPP solved in the case of SVR. To find the solution to this problem through machine learning approaches, various methods have been applied, such as artificial neural networks [21], statistical learning [22], fuzzy logic [23–26], neural networks [27–29], evolutionary algorithms [30] and hidden Markov models [31]. Eugene et al. [32], estimated that the factors for high expected returns that are due to future price increases are only offset through the decrementing of the current price. Therefore, expected returns based on the variable time generate temporary subsets of different prices. Lewellen et al. [33] proposed an approach for testing the prediction of aggregate financial ratios, named predictive regression, on small-scale sample biases. Goh et al. [34] tried to find the relationship between the U.S. and Chinese economic variables and predicted the economic variable for each country that justifies which country’s economic variables are greater than others. In 2017, Shen et al. [35] presented a novel method for predicting the Chinese stock returns for different asset values using the Baidu index. Similarly, Li et al. (2018) [36] found that idiosyncratic volatility significantly grows when internet stock message boards are already built up.

The prediction of stock market indices has been the focus of interest from the day the stock market came into existence. Researchers have several goals and motivations for trying to predict stock market prices. One of the motivations could be to make life easier and more luxurious. Many investment professionals, along with researchers, are trying to find a superior system that will yield high returns in terms of financial gain. There has been considerable work performed to predict the behavior of the stock market. To perform the financial time series prediction, various parameters are involved: (a) price of the last trade performed during the day, (b) total number of commodities traded during the day, and (c) lowest and highest traded price [37]. Because of these parameters, the nonlinearity and

uncertainty involved in the prediction of financial time series forecasting, this paper proposes TSVR to address these situations. To determine the effectiveness of TSVR on financial time series datasets, first, this paper discusses the formulation of TSVR and then the performance of the numerical experiments for various financial datasets. The experimental results of TSVR are compared with the standard SVR formulation with accuracy in terms of average RMSE and training time.

The remainder of this paper is organized as follows: Sections 2 and 3 discuss the formulation of SVR and TSVR, respectively. Section 4 shows the experimental results on different financial time series datasets of TSVR and comparison results with SVR. Finally, conclusions are drawn in section 5.

Support vector regression

This section describes the standard formulation of support vector regression (SVR). Assume that a set of training samples is $\{(x_i, y_i)\}_{i=1,2,\dots,m}$ where $x_i = (x_{i1}, x_{i2}, \dots, x_{in})^t \in R^n$ is the input example and $y_i \in R$ is the target value for $i = 1, 2, \dots, m$, where m corresponds to input training samples. Let matrix $D \in R^{m \times n}$ denote the input examples where x_i^t is the i -th row and $y = (y_1, \dots, y_m)^t$ is the vector of observed values. The main goal of SVR is to approximate the regression function $f(\cdot)$ in the form

$$f(x) = x^t w + b \tag{1}$$

where unknowns w is the vector and b is a scalar value.

Vapnik [1] suggested the formulations of SVR by introducing the ϵ -insensitive loss function and determining the unknown variables w and b by solving the following QPP:

$$\min_{(w, b, \xi_1, \xi_2) \in R^{n+1+m+m}} \frac{1}{2} w^t w + C(e^t \xi_1 + e^t \xi_2),$$

subject to:

$$y_i - x_i^t w - b \leq \epsilon + \xi_{1i},$$

$$x_i^t w + b - y_i \leq \epsilon + \xi_{2i}$$

and

$$\xi_{1i} \geq 0, \xi_{2i} \geq 0 \text{ for } i = 1, 2, \dots, m \tag{2}$$

where $\xi_1 = (\xi_{11}, \dots, \xi_{1m})^t$, $\xi_2 = (\xi_{21}, \dots, \xi_{2m})^t$ are slack variables in vector form, and $C > 0$ and $\epsilon > 0$ denote the input parameters.

Here, the solution of the above problem is obtained by introducing Lagrange multipliers

$$\min_{\lambda_1, \lambda_2 \in R^m} \frac{1}{2} \sum_{i,j=1}^m (\lambda_{1i} - \lambda_{2j})^t x_i^t x_j (\lambda_{1j} - \lambda_{2j}) + \epsilon \sum_{i=1}^m (\lambda_{1i} + \lambda_{2i}) - \sum_{i=1}^m y_i (\lambda_{1i} - \lambda_{2i})$$

subject to:

$$\sum_{i=1}^m (\lambda_{1i} - \lambda_{2i}) = 0$$

$$0 \leq \lambda_1, \lambda_2 \leq Ce, \tag{3}$$

where the Lagrange multipliers are $\lambda_1 = (\lambda_{11}, \dots, \lambda_{1m})^t$ and $\lambda_2 = (\lambda_{21}, \dots, \lambda_{2m})^t$ in R^m , which give

the solution to the above quadratic problem. Here, nonzero values of Lagrangian multipliers, which are known as support vectors in Eq (3) are useful for predicting the regression function, which is defined for any $x \in R^n$ as

$$f(x) = \sum_{i=1}^m (\lambda_{1i} - \lambda_{2i})(x^t x_i) + b \tag{4}$$

For a nonlinear regressor, the input data maps to a higher dimensional feature space using a kernel function $k(.,.)$ which is defined by the Gaussian kernel as $k(x_i, x_j) = \exp(-\mu \|x_i - x_j\|^2)$ for $i, j = 1, 2, \dots, m$ and μ is a parameter. The nonlinear case can be obtained as

$$\min_{\lambda_1, \lambda_2 \in R^m} \frac{1}{2} \sum_{i,j=1}^m (\lambda_{1i} - \lambda_{2i})^t k(x_i, x_j) (\lambda_{1j} - \lambda_{2j}) + \epsilon \sum_{i=1}^m (\lambda_{1i} + \lambda_{2i}) - \sum_{i=1}^m y_i (\lambda_{1i} - \lambda_{2i})$$

subject to:

$$\sum_{i=1}^m (\lambda_{1i} - \lambda_{2i}) = 0$$

$$0 \leq \lambda_1, \lambda_2 \leq Ce, \tag{5}$$

The nonlinear prediction function $f(.)$ is given by finding the value of λ_1 and λ_2 from the solution of the problem mentioned in Eq (5) for any $x \in R^n$,

$$f(x) = \sum_{i=1}^m (\lambda_{1i} - \lambda_{2i}) k(x, x_i) + b$$

Twin support vector machine

To further improve the generalization performance and training time of SVR, a new approach was discussed by Peng [20], termed TSVR. The TSVR constructs a pair of nonparallel hyperplanes such that one of the hyperplanes determines the ϵ -insensitive downbound $f_1(x) = x^t w_1 + b_1$ and another ϵ -insensitive upbound function $f_2(x) = x^t w_2 + b_2$ to identify the end regression function. The TSVR solves a pair of smaller QPPs of m constraints to identify the solution instead of solving a single large QPP with a $2m$ number of constraints.

The formulation of TSVR determines the regression function by the following pair of constrained QPPs as:

$$\min \frac{1}{2} \|y - \epsilon \epsilon_1 - (Dw_1 + eb_1)\|^2 + C_1 e^t \xi$$

subject to:

$$y - (Dw_1 + eb_1) \geq \epsilon \epsilon_1 - \xi, \xi \geq 0 \tag{6}$$

$$\min \frac{1}{2} \|y + \epsilon \epsilon_2 - (Dw_2 + eb_2)\|^2 + C_2 e^t \eta$$

subject to:

$$(Dw_2 + eb_2) - y \geq \epsilon \epsilon_2 - \eta, \eta \geq 0 \tag{7}$$

where $C_1, C_2 > 0$ and $\epsilon_1, \epsilon_2 \geq 0$ denote input parameters, $\xi = (\xi_1, \dots, \xi_m)^t$ and $\eta = (\eta_1, \dots, \eta_m)^t$ denote the vector of slack variables.

To find the solution of the above primal-based QPPs shown in Eqs (6) and (7), we convert the QPPs into dual forms by using the Lagrange multipliers $\lambda_1 = (\lambda_{11}, \dots, \lambda_{1m})^t$, $v_1 = (v_{11}, \dots, v_{1m})^t$ and $\lambda_2 = (\lambda_{21}, \dots, \lambda_{2m})^t$, $v_2 = (v_{21}, \dots, v_{2m})^t$. The Lagrangian functions of Eqs (6) and (7) are given by Eqs (8) and (9), respectively.

$$L_1(w_1, b_1, \zeta, \lambda_1, v_1) = \frac{1}{2} \|y - e\epsilon_1 - (Dw_1 + eb_1)\|^2 + C_1 e^t \zeta - \lambda_1 (y - (Dw_1 + eb_1) - e\epsilon_1 + \zeta) - v_1^t \zeta \tag{8}$$

$$L_2(w_2, b_2, \eta, \lambda_2, v_2) = \frac{1}{2} \|y + e\epsilon_2 - (Dw_2 + eb_2)\|^2 + C_2 e^t \eta - \lambda_2 ((Dw_2 + eb_2) - y - e\epsilon_2 + \eta) - v_2^t \eta \tag{9}$$

By applying the KKT conditions for the Lagrangian function as shown in Eq (8), we obtain:

$$-D^t(y - Dw_1 - eb_1 - e\epsilon_1) + D^t \lambda_1 = 0, \tag{10}$$

$$-e^t(y - Dw_1 - eb_1 - e\epsilon_1) + e^t \lambda_1 = 0, \tag{11}$$

$$C_1 e - \lambda_1 - v_1 = 0, \tag{12}$$

$$y - (Dw_1 + eb_1) \geq e\epsilon_1 - \zeta, \zeta \geq 0, \tag{13}$$

$$\lambda_1^t (y - (Dw_1 + eb_1) \geq e\epsilon_1 - \zeta) = 0, \lambda_1 \geq 0, \tag{14}$$

$$v_1^t \zeta = 0, v_1 \geq 0, \tag{15}$$

Since $v_1 \geq 0$, we have

$$0 \leq \lambda_1 \leq C_1 e. \tag{16}$$

Similarly, for the Lagrangian function as shown in Eq (9), we obtain

$$-D^t(y - Dw_2 - eb_2 + e\epsilon_2) - D^t \lambda_2 = 0, \tag{17}$$

$$-e^t(y - Dw_2 - eb_2 + e\epsilon_2) - e^t \lambda_2 = 0, \tag{18}$$

$$C_2 e - \lambda_2 - v_2 = 0, \tag{19}$$

$$(Dw_2 + eb_2) - y \geq e\epsilon_2 - \eta, \eta \geq 0, \tag{20}$$

$$\lambda_2^t ((Dw_2 + eb_2) - y \geq e\epsilon_2 - \eta) = 0, \lambda_2 \geq 0, \tag{21}$$

$$v_2^t \eta = 0, v_2 \geq 0, \tag{22}$$

Since $v_2 \geq 0$, we have

$$0 \leq \lambda_2 \leq C_2 e. \tag{23}$$

Combining Eq (10) with Eq (11) and Eq (17) with Eq (18), we obtain

$$-\begin{bmatrix} D^t \\ e^t \end{bmatrix} \left\{ (y - e\epsilon_1) - [D \quad e] \begin{bmatrix} w_1 \\ b_1 \end{bmatrix} \right\} + \begin{bmatrix} D^t \\ e^t \end{bmatrix} \lambda_1 = 0 \tag{24}$$

$$-\begin{bmatrix} D^t \\ e^t \end{bmatrix} \left\{ (y + e\epsilon_2) - [D \quad e] \begin{bmatrix} w_2 \\ b_2 \end{bmatrix} \right\} - \begin{bmatrix} D^t \\ e^t \end{bmatrix} \lambda_2 = 0 \tag{25}$$

Let us define,

$$S = [D \quad e], \quad u_1 = [w_1^t \quad b_1^t]^t, \quad u_2 = [w_2^t \quad b_2^t]^t, \quad f_1 = y - e\epsilon_1, \quad f_2 = y + e\epsilon_2, \tag{26}$$

and then we have,

$$-S^t f_1 + S^t S u_1 + S^t \lambda_1 = 0,$$

i.e.,

$$u_1 = (S^t S)^{-1} S^t (f_1 - \lambda_1). \tag{27}$$

and

$$-S^t f_2 + S^t S u_2 - S^t \lambda_2 = 0,$$

, i.e.,

$$u_2 = (S^t S)^{-1} S^t (f_2 + \lambda_2). \tag{28}$$

Here, note that $S^t S$ is positive semidefinite, but to overcome the situation in which its inverse does not exist, σI is introduced as a regularization term, so that $(S^t S + \sigma I)$ becomes positive definite where σ is a very small positive number, such as $\sigma = 1e-7$. Thus, we have

$$u_1 = (S^t S + \sigma I)^{-1} S^t (f_1 - \lambda_1) \tag{29}$$

$$u_2 = (S^t S + \sigma I)^{-1} S^t (f_2 + \lambda_2) \tag{30}$$

Substituting Eq (29) into the primal Lagrangian function Eq (8) and using Eqs (13) to (16), the dual problem of Eq (6) is obtained as

$$\max -\frac{1}{2} \lambda_1^t S (S^t S)^{-1} S^t \lambda_1 + f_1^t S (S^t S)^{-1} S^t \lambda_1 - f_1^t \lambda_1$$

subject to:

$$0 \leq \lambda_1 \leq eC_1 \tag{31}$$

Similarly, substituting Eq (30) into the primal Lagrangian function Eq (9) and using Eq (20) to (23), the dual problem of Eq (7) is obtained as

$$\max -\frac{1}{2} \lambda_2^t S (S^t S)^{-1} S^t \lambda_2 - f_2^t S (S^t S)^{-1} S^t \lambda_2 + f_2^t \lambda_2$$

subject to:

$$0 \leq \lambda_2 \leq eC_2 \tag{32}$$

The vectors λ_1 and λ_2 are calculated by solving the dual QPPs Eqs (31) and (32). Finally, in the output for any data point $x \in R^n$, the end regressor $f(\cdot)$ is given by:

$$f(x) = \frac{1}{2}(f_1(x) + f_2(x)). \tag{33}$$

To extend TSVR to a nonlinear case, TSVR finds the regression function by solving the following primal problems:

$$\min \frac{1}{2} \|y - e\epsilon_1 - (K(D, D^t)w_1 + eb_1)\|^2 + C_1 e^t \xi$$

subject to:

$$y - (K(D, D^t)w_1 + eb_1) \geq e\epsilon_1 - \xi, \quad \xi \geq 0 \tag{34}$$

and

$$\min \frac{1}{2} \|y + e\epsilon_2 - (K(D, D^t)w_2 + eb_2)\|^2 + C_2 e^t \eta$$

subject to:

$$(K(D, D^t)w_2 + eb_2) - y \geq e\epsilon_2 - \eta, \quad \eta \geq 0 \tag{35}$$

where the kernel matrix $K(D, D^t)$ of order m whose (i, j) element is given by $K(D, D^t)_{ij} = k(x_i, x_j) \in R$, and where $k(x_i, x_j)$ is a nonlinear kernel function. For a vector $x \in R^n$, we define

$$k(x^t, D^t) = (k(x, x_1), \dots, k(x, x_m))$$

in a similar manner, the dual formulations of QPPs Eqs (34) and (35) are given by Eqs (36) and (37), respectively.

$$\max -\frac{1}{2} \lambda_1^t T (T^t T)^{-1} T^t \lambda_1 + f_1^t T (T^t T)^{-1} T^t \lambda_1 - f_1^t \lambda_1$$

subject to:

$$0 \leq \lambda_1 \leq eC_1 \tag{36}$$

and

$$\max -\frac{1}{2} \lambda_2^t T (T^t T)^{-1} T^t \lambda_2 - f_2^t T (T^t T)^{-1} T^t \lambda_2 + f_2^t \lambda_2$$

subject to:

$$0 \leq \lambda_2 \leq eC_2 \tag{37}$$

where $T = [K(D, D^t) \ e]$. After resolving Eqs (36) and (37), we find the value of u_1 and u_2 as

$$u_1 = (T^t T + \sigma I)^{-1} T^t (f_1 - \lambda_1) \tag{38}$$

$$u_2 = (T^t T + \sigma I)^{-1} T^t (f_2 + \lambda_2) \tag{39}$$

Finally, for any data sample $x \in R^n$, the end regression function $f(\cdot)$ is given by:

$$f(x) = \frac{1}{2}([K(x^t, D^t) \mathbf{1}](u_1 + u_2)) \tag{40}$$

Numerical experiments

In this section, various numerical experiments are conducted to test the generalization performance and the computational efficiency of the TSVR on standard datasets and compared with SVR. This paper considered 44 benchmark datasets and divided them into two groups. The first group has a combination of 24 individual company stocks, and the second group has 20 stock market index datasets from the Yahoo financial website, i.e., <http://finance.yahoo.com> [38]. Individual company stock datasets are AT&T Inc. (T), Infosys Limited (INFY), Apple, Inc. (AAPL), Facebook, Inc. (FB), Cisco Systems, Inc. (CSCO), Alphabet, Inc. (Goog), Citigroup, Inc. (C), HSBC Holding Plc (HSBC), ICICI Bank, Ltd. (IBN), Royal Bank of Canada (RY), Royal Bank of Scotland (RBS), State Bank of India (SBIN.NS), Punjab National Bank (PNB.NS), International Business Machines Corporation (IBM), Microsoft Corporation (MSFT), Tata Consultancy Services Limited (TCS.BO), Oracle Corporation (ORCL), Bharat Petroleum Corporation Limited (BPCL.NS), Oil India Limited (OIL.NS), Oil and Natural Gas Corporation (ONGC.NS), Royal Dutch Shell Plc (RDS-B), Exxon Mobil Corporation (XOM), Sinopec Shanghai Petrochemical Company Limited (SHI), Hindustan Petroleum Corporation Limited (HINDPETRO.NS) and the stock market index datasets are S&P BSE SENSEX (BSESN), NIFTY 50 (NSEI), CAC 40 (FCHI), ESTX 50 PR.EUR (STOXX50E), KOSPI Composite (KS11), IBEX 35 (IBEX), Nikkei 225 (N225), AEX (AEX), DAX PERFORMANCE (GDAXI), IBOVESPA (BVSP), S&P/TSX Composite (GSPTSE), IPC MEXICO (MXX), SMI PR (SSMI), Dow Jones Industrial Average (DJI), HANG SENG INDEX (HSI), TSEC weighted index (TWII), NASDAQ Composite (IXIC), BEL 20 (BFX), Austrian Traded Index in EUR (ATX), Jakarta Composite Index (JKSE). The details of these datasets are listed in Table 1 and Table 2, respectively.

All computations are carried out on a PC with Windows 7 OS, with a 32 bit, 3.10 GHz Intel core i5-2400 processor with 4 GB of RAM under the MATLAB R2012b environment. This paper used the MOSEK optimization toolbox to solve the quadratic programming problem in SVR and TSVR formulations, which is taken from <http://www.mosek.com> [39].

All the datasets are normalized in the following manner so that each feature value lies in [0, 1]:

$$\bar{d}_{ij} = \frac{d_{ij} - d_j^{\min}}{d_j^{\max} - d_j^{\min}}$$

where \bar{d}_{ij} is the normalized value corresponding to d_{ij} and $d_j^{\max} = \max_{i=1}^m (d_{ij})$ and $d_j^{\min} = \min_{i=1}^m (d_{ij})$ denote the maximum and minimum values of the j -th feature of A , respectively. To measure the prediction performance, this paper considered the root mean square error (RMSE), which is given by

$$RMSE = \sqrt{\frac{1}{P} \sum_{i=1}^P (y_i - \tilde{y}_i)^2},$$

where the total number of test samples is denoted by P , and \tilde{y}_i is the predicted value corresponding to the observed values. To construct a nonlinear regressor, we use a Gaussian kernel

$$k(x, y) = \exp(-\mu \|x - y\|^2)$$

Table 1. Individual stock financial details with their stock exchanges, types and listing abbreviations.

Company name	Registered stock exchange	Listing abbreviation
AT&T Inc.	Equity-NYSE	T
Infosys Limited	Equity-NYSE	INFY
Apple Inc.	Equity-NASDAQ	AAPL
Facebook Inc.	Equity-NASDAQ	FB
Cisco Systems, Inc.	Equity-NASDAQ	CSCO
Alphabet Inc.	Equity-NASDAQ	Goog
Citigroup Inc.	Equity-NYSE	C
HSBC Holding Plc	Equity-NYSE	HSBC
ICICI Bank Ltd.	Equity-NYSE	IBN
Royal Bank of Canada	Equity-NYSE	RY
Royal Bank of Scotland	Equity-NYSE	RBS
State Bank of India	Equity-NSE	SBIN.NS
Punjab National Bank	Equity-NSE	PNB.NS
International Business Machines Corporation	Equity-NYSE	IBM
Microsoft Corporation	Equity-NASDAQ	MSFT
Tata Consultancy Services Limited	Equity-BSE	TCS.BO
Oracle Corporation	Equity-NYSE	ORCL
Bharat Petroleum Corporation Limited	Equity-NSE	BPCL.NS
Oil India Limited	Equity-NSE	OIL.NS
Oil and Natural Gas Corporation	Equity-NSE	ONGC.NS
Royal Dutch Shell Plc	Equity-NYSE	RDS-B
Exxon Mobil Corporation	Equity-NYSE	XOM
Sinopec Shanghai Petrochemical Company Limited	Equity-NYSE	SHI
Hindustan Petroleum Corporation Limited	Equity-NSE	HINDPETRO.NS

<https://doi.org/10.1371/journal.pone.0211402.t001>

where vector $x, y \in R^m$ and $\mu > 0$. The optimal parameter values of $C = C_1 = C_2$ are selected from the sets $\{10^{-5}, \dots, 10^5\}$ and μ from the set $\{2^{-5}, \dots, 2^5\}$ for the training using 10-fold cross validation. By using the optimal values, the whole dataset is divided into 10 equal parts at random, out of which one part is used for testing and the remaining parts for the training to obtain the computational test accuracy. Finally, to measure the prediction, the average RMSE of the test accuracies is considered.

Individual stocks datasets of company

Individual company stocks such as SBIN.NS, PNB.NS, BPCL.NS, OIL.NS, TCS.BO, HINDPETRO.NS, ONGC.NS consist of 735 closing prices, while T, INFY, AAPL, FB, CSCO, Goog, C, HSBC, IBN, RY, RBS, IBM, MSFT, ORCL, RDS-B, XOM, SHI have a total of 751 closing prices starting from 01-01-2015 to 31-12-2017. The current value is predicted by the previous five closing prices.

Linear case. In the linear case, Table 3 shows the average RMSE for the optimal parameter values with standard deviation and the training time in seconds. Fig 1 shows the absolute prediction error of SVR and TSVR for the linear kernel on the SHI dataset. Fig 2 shows the actual and predicted values of SVR and TSVR for the linear kernel on the SHI dataset. To verify the performance of both algorithms statistically on 24 individual stock datasets, we perform a simple, nonparametric safe test, i.e., the Friedman test with the corresponding post hoc test [40].

Table 2. Financial stock market index details with their stock exchanges, types and listing abbreviations.

Stock market index name	Registered stock exchange	Listing abbreviation
S&P BSE SENSEX	Index-Bombay Stock Exchange	BSESN
NIFTY 50	Index-National Stock Exchange	NSEI
CAC 40	Index-Paris Stock Exchange	FCHI
ESTX 50 PR.EUR	Index-Zurich Stock Exchange	STOXX50E
KOSPI Composite Index	Index-Korea Stock Exchange	KS11
IBEX 35.	Index-Madrid Stock Exchange	IBEX
Nikkei 225	Index-Osaka Stock Exchange	N225
AEX-INDEXT	Index-Amsterdam Stock Exchange	AEX
DAX PERFORMANCE-INDEXT	Index-Xetra, Frankfurt Stock Exchange	GDAXI
IBOVESPA	Index-Sao Paulo Stock Exchange	BVSP
S&P/TSX Composite index	Index-Toronto Stock Exchange	GSPTSE
IPC MEXICO	Index-Mexico Stock Exchange	MXX
SMI PR	Index-VTX,SIX Swiss Exchange	SSMI
Dow Jones Industrial Average	Index-New York Stock Exchange	DJI
HANG SENG INDEX	Index-Hong Kong Stock Exchange	HSI
TSEC weighted index	Index-Taiwan Stock Exchange	TWII
NASDAQ Composite	Index-Nasdaq GIDS, American stock exchange	IXIC
BEL 20	Index-Brussels Stock Exchange	BFX
Austrian Traded Index in EUR	Index-Vienna Stock Exchange	ATX
Jakarta Composite Index	Index-Jakarta Stock Exchange	JKSE

<https://doi.org/10.1371/journal.pone.0211402.t002>

For this, the average rank of 24 datasets for the linear case is tabulated in Table 4. The Friedman statistic [40] can be computed under the null hypothesis, as shown in Table 4.

$$\chi_F^2 = \frac{12 \times 24}{2 \times (2 + 1)} \left[(1.416667^2 + 1.583333^2) - \frac{2 \times (2 + 1)^2}{4} \right] \cong 0.6667$$

$$F_F = \frac{(24 - 1) \times 0.6667}{24 \times (2 - 1) - 0.6667} \cong 0.6572$$

where F_F is distributed according to the F -distribution with (1, 23), which has the critical value 4.2793 for the level of significance $\alpha = 0.05$. Here, F_F is lower than the critical value, i.e., $0.6572 < 4.2793$, so there is no significant difference between these two algorithms for the linear case.

Nonlinear case. In the nonlinear case, Table 5 shows the average RMSE for the optimal parameter values with the standard deviation and the training time in seconds. From Table 5, we can conclude that TSVR gives better results in 19 cases out of 24 datasets in terms of average RMSE of test accuracy, which signifies the performance of TSVR in comparison to SVR in terms of prediction. Additionally, it shows the superiority of TSVR with respect to SVR in terms of computational time.

Similar to linear case, for individual stocks, the Friedman statistic can be computed under the null hypothesis from Table 4, which shows that both algorithms have a similar performance:

$$\chi_F^2 = \frac{12 \times 24}{2 \times (2 + 1)} \left[(1.791667^2 + 1.208333^2) - \frac{2 \times (2 + 1)^2}{4} \right] \cong 8.1667$$

$$F_F = \frac{(24 - 1) \times 8.1667}{24 \times (2 - 1) - 8.1667} \cong 11.8632$$

Table 3. Performance comparison of TSVR with SVR on individual companies' stock datasets using a linear kernel. RMSE is used for comparison. Time is used for the training in seconds.

Dataset	SVR (C) Time	TSVR (C1 = C2) Time
C (751X6)	0.01865+0.00425 (10 ⁻²) 10.9164	0.01866+0.00415 (10 ⁻²) 1.37938
HSBC (751X6)	0.02305+0.00393 (10 ⁻²) 11.2698	0.02306+0.004 (10 ⁻¹) 1.35164
IBN (751X6)	0.02234+0.00383 (10 ⁻¹) 11.155	0.02232+0.00389 (10 ⁻¹) 1.30003
PNB_NS (735X6)	0.02609+0.01228 (10 ⁻¹) 10.5834	0.026+0.0123 (10 ⁻²) 1.6011
RY (751X6)	0.0186+0.00467 (10 ⁻¹) 11.6193	0.05662+0.10278 (10 ⁻⁵) 1.48767
RBS (751X6)	0.01923+0.00868 (10 ⁻²) 11.5917	0.01923+0.00869 (10 ⁻¹) 1.05937
SBIN_NS (735X6)	0.02872+0.00915 (10 ⁻⁰) 10.885	0.02885+0.00922 (10 ⁻¹) 1.72065
AAPL (751X6)	0.01989+0.0037 (10 ⁻³) 11.6285	0.0199+0.00368 (10 ⁻²) 1.05923
AT&T (751X6)	0.03191+0.00567 (10 ⁻²) 11.6016	0.11026+0.25187 (10 ⁻⁵) 1.31281
CSCO (751X6)	0.02219+0.00506 (10 ⁻¹) 11.0575	0.02221+0.00506 (10 ⁻²) 0.95512
FB (751X6)	0.01507+0.00474 (10 ⁻¹) 11.1297	0.01506+0.00476 (10 ⁻²) 1.23056
GOOG (751X6)	0.01647+0.00412 (10 ⁻¹) 11.1765	0.09118+0.23693 (10 ⁻⁵) 1.24606
IBM (751X6)	0.02789+0.0057 (10 ⁻²) 10.8029	0.02781+0.00573 (10 ⁻²) 0.98122
INFY (751X6)	0.04022+0.01021 (10 ⁻²) 10.7731	0.04016+0.01022 (10 ⁻⁰) 1.00669
MSFT (751X6)	0.01601+0.00416 (10 ⁻¹) 10.9651	0.01604+0.00411 (10 ⁻²) 1.04679
ORCL (751X6)	0.02717+0.00563 (10 ⁻¹) 10.9254	0.02719+0.00563 (10 ⁻²) 1.56594
TCS_BO (735X6)	0.02581+0.0391 (10 ⁻¹) 10.387	0.02301+0.03058 (10 ⁻¹) 1.01678
BPCL_NS (735X6)	0.02017+0.00389 (10 ⁻¹) 10.3464	0.02012+0.00392 (10 ⁻²) 1.23517
HINDPETRO_NS (735X6)	0.01594+0.00654 (10 ⁻²) 10.8712	0.02444+0.02563 (10 ⁻⁵) 1.12206
OIL_NS (735X6)	0.02393+0.00654 (10 ⁻²) 10.3806	0.02389+0.00651 (10 ⁻¹) 1.06545

(Continued)

Table 3. (Continued)

Dataset	SVR (C) Time	TSVR (C1 = C2) Time
ONGC_NS (735X6)	0.02515+0.00599 (10 ⁻¹) 10.3224	0.02521+0.00602 (10 ⁻²) 1.23187
RDS_B (751X6)	0.02553+0.00797 (10 ⁻¹) 10.8088	0.0255+0.00802 (10 ⁻¹) 1.34066
SHI (751X6)	0.03325+0.01405 (10 ⁻²) 10.7923	0.03331+0.01461 (10 ⁻²) 1.1263
XOM (751X6)	0.03385+0.01181 (10 ⁻³) 10.7708	0.0339+0.0119 (10 ⁻²) 1.2065

<https://doi.org/10.1371/journal.pone.0211402.t003>

where F_F is the distribution according to the F -distribution and $(1, 1 \times 23) = (1, 23)$ is the degree of freedom. Here, 4.2793 is the critical value of $F(1, 23)$ for the level of significance at $\alpha = 0.05$. Since the value of $F_F = 11.8632 > 4.2793$, we reject the null hypothesis. Furthermore, we performed pairwise comparisons using the Nemenyi post hoc test of all reported methods and verified the significant difference between their average ranks by computing the critical difference (CD) at $p = 0.10$. The difference between their ranks should be at least $1.645 \sqrt{\frac{2 \times (2+1)}{6 \times 24}} \approx 0.3358$.

Since the difference between the average ranks of TSVR with SVR ($1.791667 - 1.208333 = 0.583334$) is greater than 0.3358, we conclude that TSVR is significantly better than SVR for individual stock datasets. For the non-linear case, the absolute prediction error of SVR and

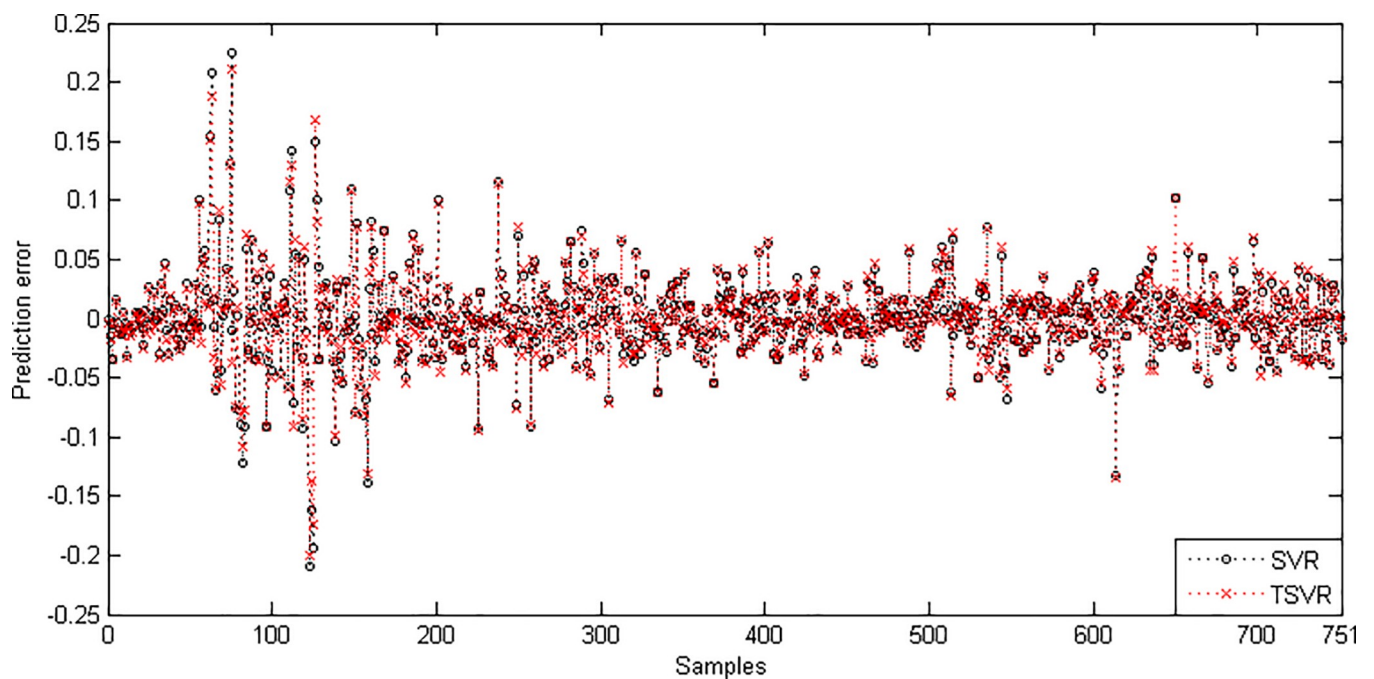


Fig 1. Prediction error plots using a linear kernel on the SHI dataset.

<https://doi.org/10.1371/journal.pone.0211402.g001>

TSVR is shown in Figs 3 and 4 for the FB and RY datasets, respectively. Additionally, the actual and predicted values of SVR and TSVR are plotted in Figs 5 and 6 for the FB and RY datasets, respectively. It can easily be observed that TSVR is in close agreement with the observed values compared to SVR.

Stock market index datasets

Stock market index datasets such as BSESN and HSI consist of 733 closing prices, while DJI and IXIC have 751 closing prices; the FCHI and IBEX datasets consist of 763 closing prices; the JKSE and TWII datasets consist of 724 closing prices; MXX and SSMI have 750 closing points; AEX consists of 763 closing points; ATX consists of 737 closing points; BFX consists of 762 closing points; BVSP consists of 738 closing points and GDAXI, GSPTSE, KS11, N225, NSEI, STOXX50E consist of 755, 748, 728, 732, 731, 745 closing points, respectively, from 01-01-2015 to 31-12-2017. The current value is predicted by using the previous five closing prices.

Linear case. For the linear kernel, Table 6 shows the average RMSE for the optimal parameter values with the standard deviation and the training time in seconds. We can conclude that TSVR gives better results in 13 cases out of 20 datasets in terms of average RMSE of test accuracy. Additionally, the training time of TSVR is lower than that of SVR. The Friedman statistical non-parametric post hoc test is performed on the average rank of 20 financial datasets from Table 7. The Friedman statistic [40] can be computed under the null hypothesis for the linear case:

$$\chi^2_F = \frac{12 \times 20}{2 \times (2 + 1)} \left[(1.65^2 + 1.35^2) - \frac{2 \times (2 + 1)^2}{4} \right] \cong 1.80$$

$$F_F = \frac{(20 - 1) \times 1.8}{20 \times (2 - 1) - 1.8} \cong 1.8791$$

where F_F is distributed according to the F -distribution with (1,19), which has the critical value 4.3807 for the level of significance $\alpha = 0.05$. Here, F_F is less than the critical value, so there is no

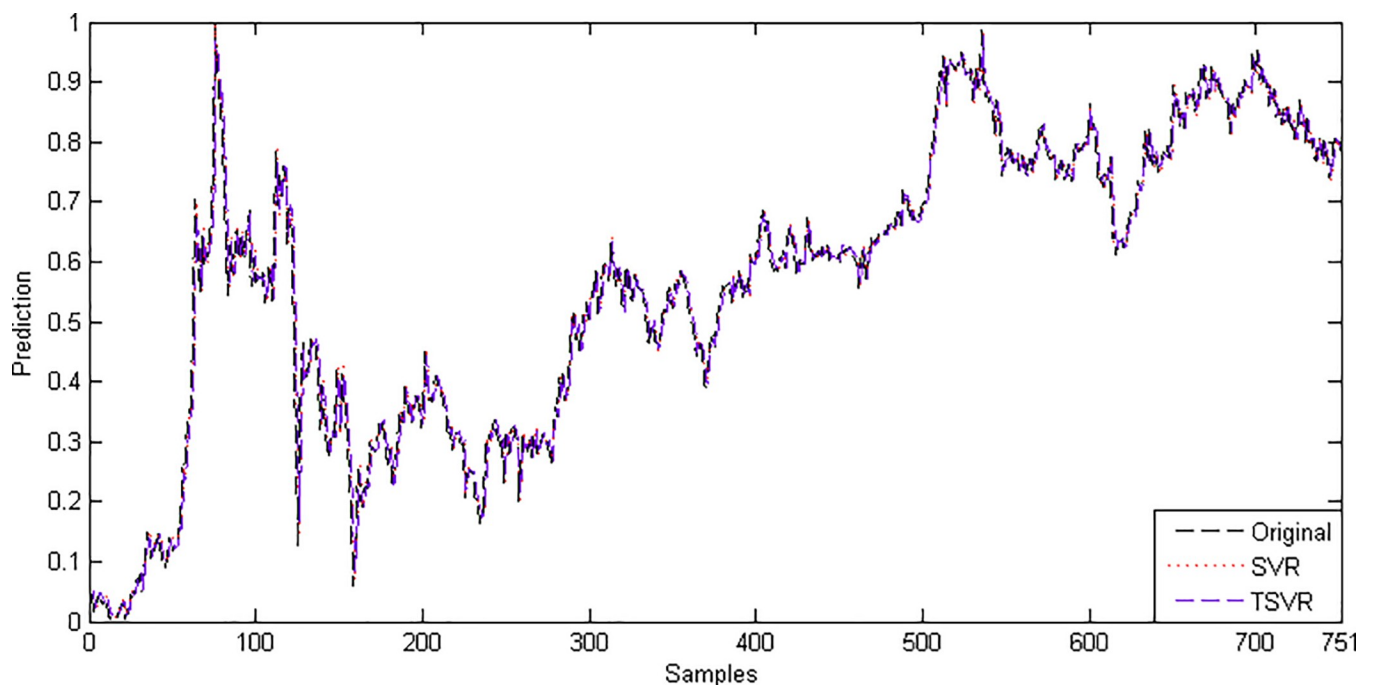


Fig 2. Predicted and actual values using a linear kernel on the SHI dataset.

<https://doi.org/10.1371/journal.pone.0211402.g002>

Table 4. Average ranks of TSVR with SVR on individual companies' stocks using a linear and Gaussian kernel.

Dataset	Linear		Non-Linear	
	SVR	TSVR	SVR	TSVR
C	1	2	2	1
HSBC	1	2	1	2
IBN	2	1	2	1
PNB_NS	2	1	2	1
RY	1	2	2	1
RBS	2	1	2	1
SBIN_NS	1	2	1	2
AAPL	1	2	2	1
AT&T	1	2	1	2
CSCO	1	2	2	1
FB	2	1	2	1
GOOG	1	2	2	1
IBM	2	1	1	2
INFY	2	1	2	1
MSFT	1	2	2	1
ORCL	1	2	2	1
TCS_BO	2	1	2	1
BPCL_NS	2	1	2	1
HINDPETRO_NS	1	2	2	1
OIL_NS	2	1	2	1
ONGC_NS	1	2	2	1
RDS_B	2	1	2	1
SHI	1	2	2	1
XOM	1	2	1	2
Average rank	1.416667	1.583333	1.791667	1.208333

<https://doi.org/10.1371/journal.pone.0211402.t004>

significant difference between these two algorithms for the linear case. Fig 7 shows the absolute prediction error plot of SVR and TSVR for the linear kernel on the BFX dataset. Fig 8 also shows the actual and predicted values of SVR and TSVR for the linear kernel on the market stock index BFX dataset. One can easily conclude that TSVR is in close agreement with the target values compared to SVR.

Nonlinear case. For the non-linear kernel, Table 8 shows the average RMSE for the optimal parameter value with the standard deviation and the training time in seconds. We can conclude that TSVR gives better results in 19 out of 20 datasets in terms of average RMSE of test accuracy. The training time of TSVR is less than that of SVR due to solving a pair of smaller-sized QPPs instead of a large QPP, as in the case of SVR. This shows the superiority of TSVR with respect to SVR.

In the nonlinear case for different stock market index datasets, the Friedman statistic can be computed under the null hypothesis from Table 7 as:

$$\chi_F^2 = \frac{12 \times 20}{2 \times (2 + 1)} \left[(1.95^2 + 1.05^2) - \frac{2 \times (2 + 1)^2}{4} \right] \cong 16.2$$

$$F_F = \frac{(20 - 1) \times 16.2}{20 \times (2 - 1) - 16.2} \cong 81$$

Table 5. Performance comparison of TSVR with SVR on individual companies' stock datasets using a Gaussian kernel. RMSE is used for comparison. Time is used for the training in seconds.

Dataset	SVR (C,μ) Time	TSVR (C1 = C2,μ) Time
C (751X6)	0.0197+0.00459 (10 ⁰ ,2 ⁻¹) 12.8056	0.01925+0.00438 (10 ³ ,2 ⁻²) 1.84222
HSBC (751X6)	0.02342+0.00369 (10 ⁻¹ ,2 ⁻²) 12.7637	0.02355+0.00378 (10 ² ,2 ⁻³) 1.46775
IBN (751X6)	0.02394+0.00612 (10 ⁰ ,2 ⁻⁴) 12.6499	0.02252+0.00407 (10 ² ,2 ⁻⁵) 1.69161
PNB_NS (735X6)	0.02619+0.01202 (10 ⁻¹ ,2 ⁻²) 12.1768	0.02598+0.01208 (10 ³ ,2 ⁻⁵) 1.45579
RY (751X6)	0.02098+0.00585 (10 ⁰ ,2 ⁻⁵) 12.7447	0.01911+0.00521 (10 ¹ ,2 ⁻⁵) 1.70301
RBS (751X6)	0.01948+0.00876 (10 ⁻¹ ,2 ⁻²) 12.5943	0.01939+0.00868 (10 ³ ,2 ⁻⁵) 1.5127
SBIN_NS (735X6)	0.02908+0.00925 (10 ⁻¹ ,2 ⁻²) 11.704	0.02912+0.00981 (10 ² ,2 ⁻⁵) 1.46473
AAPL (751X6)	0.0207+0.00421 (10 ⁻¹ ,2 ⁻²) 12.3965	0.01995+0.00371 (10 ² ,2 ⁻⁵) 1.53151
AT&T (751X6)	0.03185+0.0057 (10 ⁻¹ ,2 ⁻²) 12.3917	0.03192+0.00601 (10 ² ,2 ⁻⁵) 1.49755
CSCO (751X6)	0.02362+0.00534 (10 ⁰ ,2 ⁻⁵) 12.3661	0.02243+0.00511 (10 ³ ,2 ⁻⁵) 1.81103
FB (751X6)	0.01743+0.00519 (10 ⁰ ,2 ⁻⁵) 12.2778	0.01515+0.00465 (10 ² ,2 ⁻⁴) 1.79247
GOOG (751X6)	0.01828+0.00648 (10 ⁻¹ ,2 ⁻²) 12.3224	0.01659+0.00417 (10 ² ,2 ⁻⁴) 1.50103
IBM (751X6)	0.02855+0.00581 (10 ⁰ ,2 ⁻⁵) 12.1636	0.21208+0.12217 (10 ⁻³ ,2 ⁻³) 1.73274
INFY (751X6)	0.0402+0.01014 (10 ⁰ ,2 ⁻⁴) 12.2526	0.04002+0.01014 (10 ¹ ,2 ⁻⁵) 1.69419
MSFT (751X6)	0.01793+0.00522 (10 ⁰ ,2 ⁻⁵) 12.3601	0.01629+0.00434 (10 ³ ,2 ⁻⁵) 1.74959
ORCL (751X6)	0.02844+0.00647 (10 ⁻¹ ,2 ⁻⁵) 12.2863	0.02717+0.00566 (10 ² ,2 ⁻⁵) 1.5399
TCS_BO (735X6)	0.0199+0.02908 (10 ⁻¹ ,2 ⁻²) 11.7124	0.01963+0.02914 (10 ⁰ ,2 ⁻¹) 1.57151
BPCL_NS (735X6)	0.0204+0.00377 (10 ⁰ ,2 ⁻²) 11.7141	0.02023+0.00395 (10 ² ,2 ⁻⁵) 1.61242

(Continued)

Table 5. (Continued)

Dataset	SVR (C,μ) Time	TSVR (C1 = C2,μ) Time
HINDPETRO_NS (735X6)	0.01869+0.00916 (10 ^{1,2} ^-4) 11.8947	0.01607+0.00664 (10 ^{3,2} ^-3) 1.52778
OIL_NS (735X6)	0.02512+0.00797 (10 ^{0,2} ^-2) 11.7162	0.02407+0.0067 (10 ^{2,2} ^-5) 1.63295
ONGC_NS (735X6)	0.02644+0.00678 (10 ^{-1,2} ^-4) 11.7554	0.02581+0.00658 (10 ^{2,2} ^-5) 1.37471
RDS_B (751X6)	0.02737+0.01047 (10 ^{-1,2} ^-4) 12.3922	0.02587+0.00841 (10 ^{1,2} ^-5) 1.48654
SHI (751X6)	0.03433+0.01577 (10 ^{-1,2} ^-4) 12.3041	0.03366+0.01511 (10 ^{2,2} ^-5) 1.45092
XOM (751X6)	0.03391+0.01177 (10 ^{0,2} ^-4) 12.3424	0.03395+0.01186 (10 ^{2,2} ^-5) 1.7403

<https://doi.org/10.1371/journal.pone.0211402.t005>

where F_F is the distribution according to the F -distribution with $(1,1 \times 19) = (1,19)$ as the degree of freedom. Here, 4.3807 is the critical value of $F(1,19)$ for the level of significance at $\alpha = 0.05$. Since the value of $F_F = 81 > 4.3807$ is rejected, we reject the null hypothesis. Similar to the previous case, we perform pairwise comparisons using the Nemenyi post hoc test for all reported

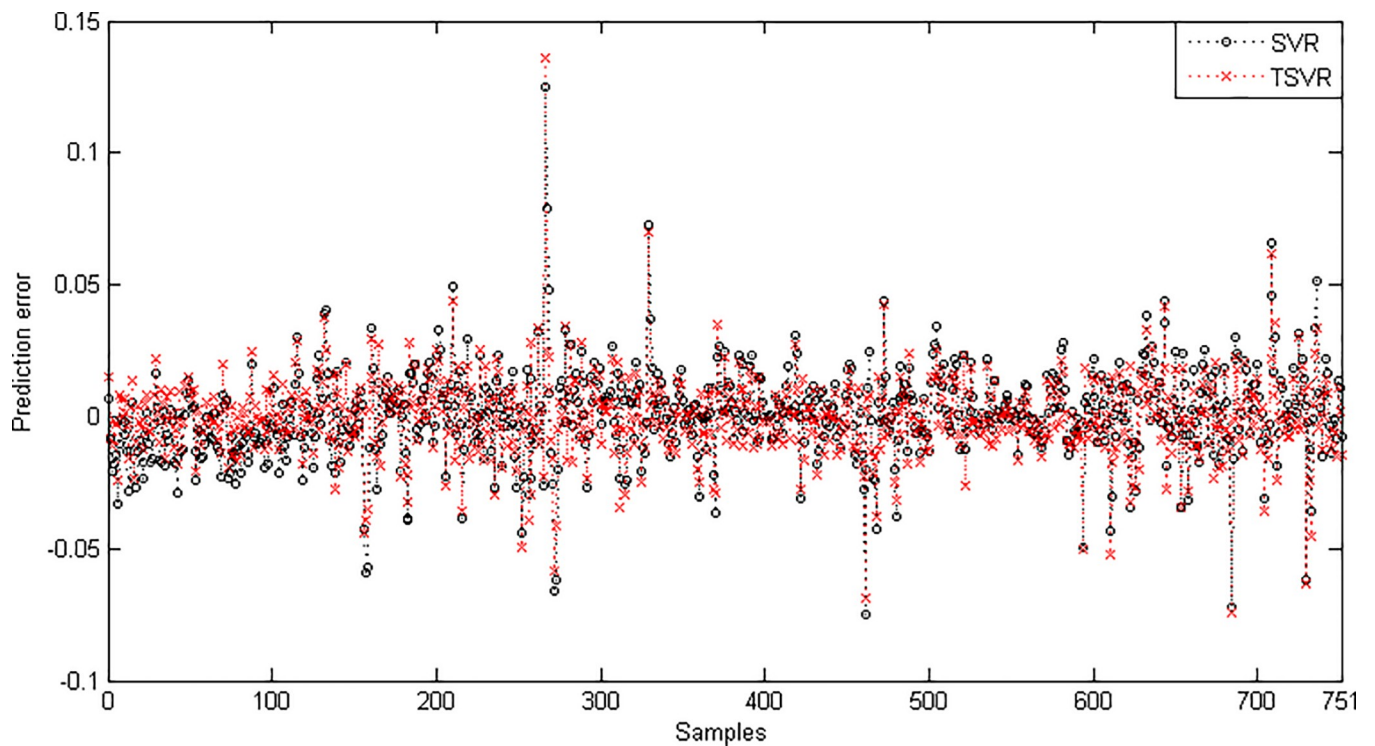


Fig 3. Prediction error plots using a Gaussian kernel on the FB dataset.

<https://doi.org/10.1371/journal.pone.0211402.g003>

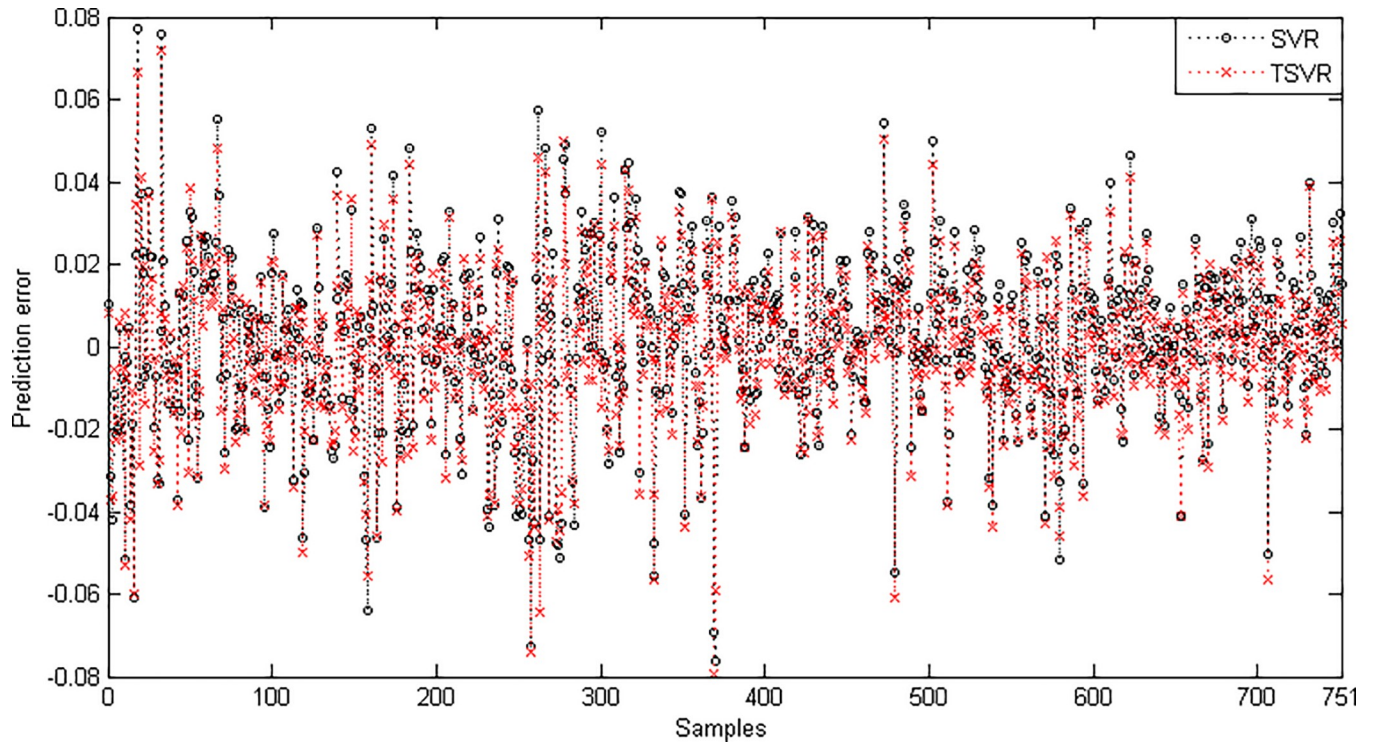


Fig 4. Prediction error plots using a Gaussian kernel on the RY dataset.

<https://doi.org/10.1371/journal.pone.0211402.g004>

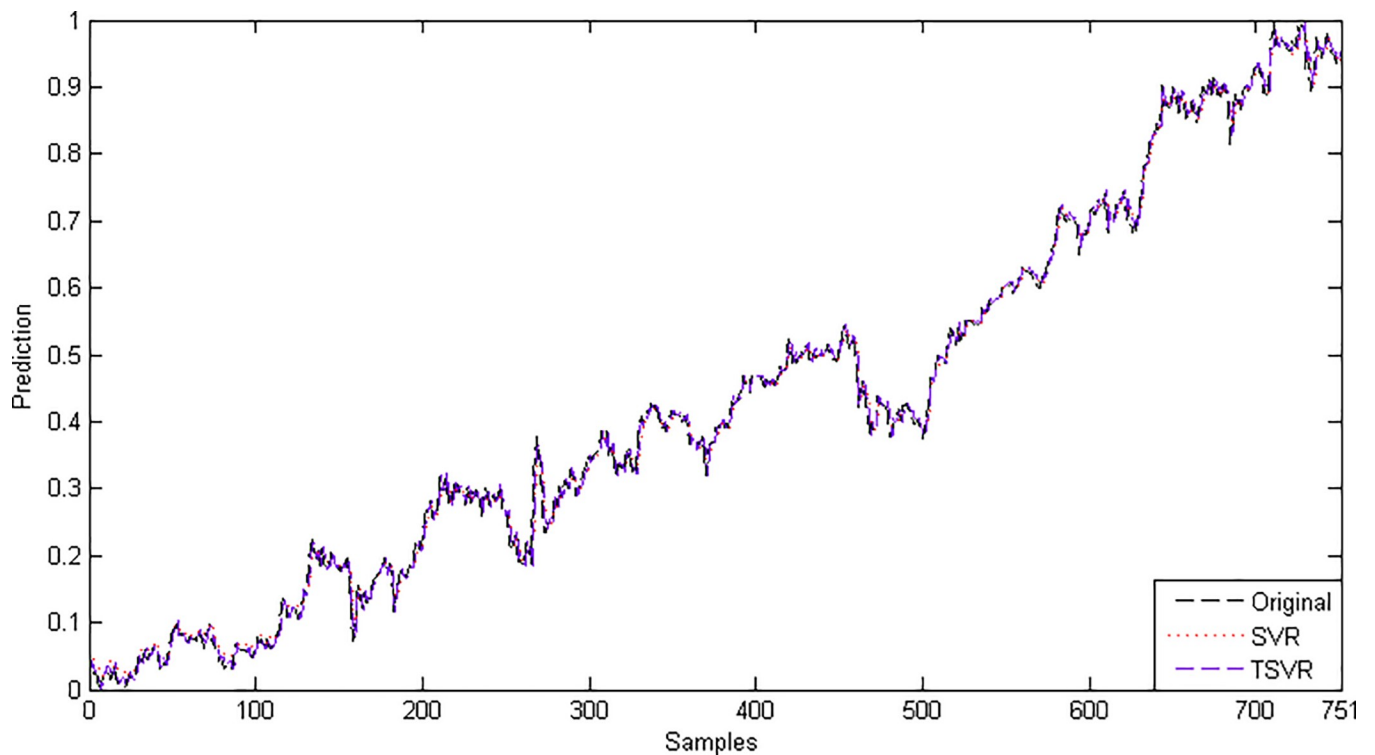


Fig 5. Predicted and actual values using a Gaussian kernel on the FB dataset.

<https://doi.org/10.1371/journal.pone.0211402.g005>

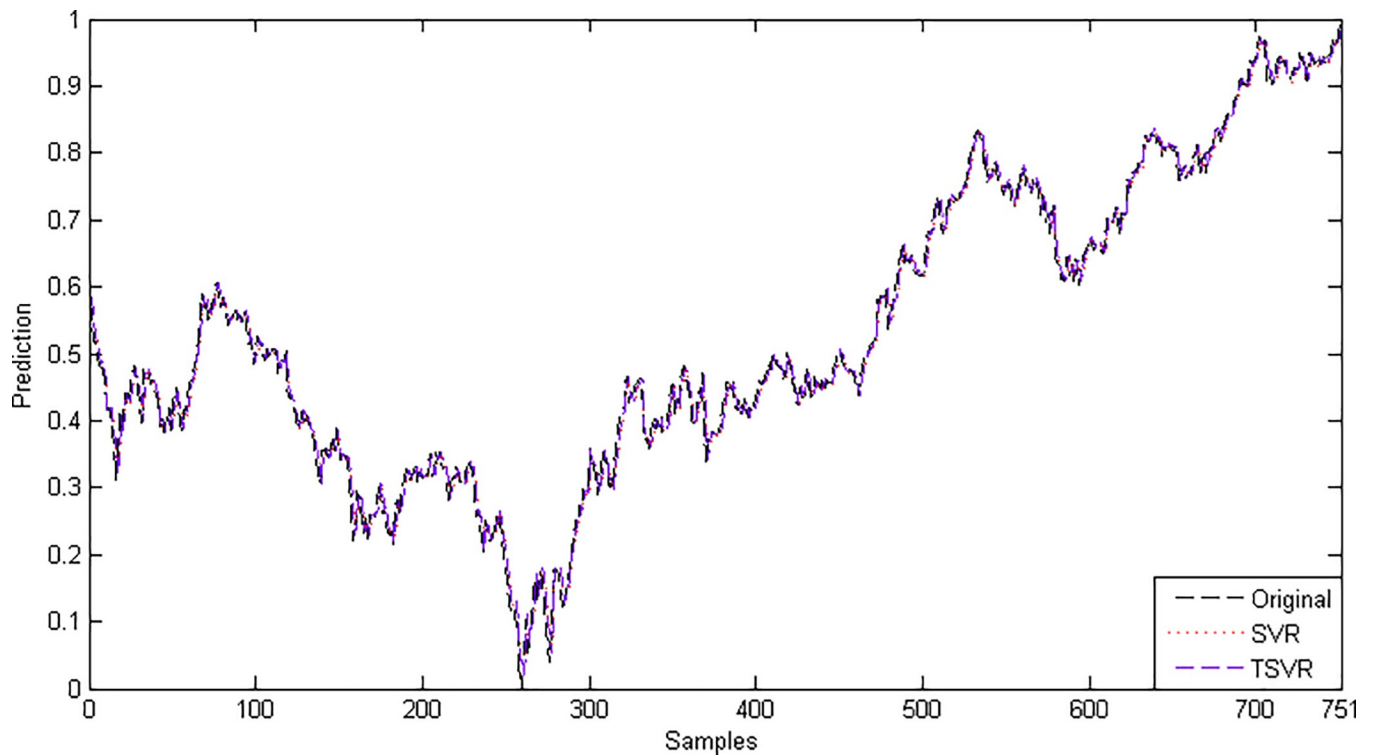


Fig 6. Predicted and actual values using a Gaussian kernel on the RY dataset.

<https://doi.org/10.1371/journal.pone.0211402.g006>

Table 6. Performance comparison of TSVR with SVR on stock market index datasets using a linear kernel. RMSE is used for comparison. Time is used for the training in seconds.

Dataset	SVR (C) Time	TSVR (C1 = C2) Time
AEX (763X6)	0.02683+0.01051 (10 ⁻¹) 11.8233	0.02678+0.01061 (10 ⁵) 1.47306
ATX (737X6)	0.01886+0.00414 (10 ⁻²) 10.3641	0.01885+0.0043 (10 ¹) 1.1216
BFX (762X6)	0.03424+0.01144 (10 ⁻¹) 11.3085	0.03545+0.01039 (10 ³) 1.16305
BSESN (733X6)	0.02062+0.00448 (10 ⁻¹) 10.2492	0.02071+0.00445 (10 ¹) 1.22084
BVSP (738X6)	0.01993+0.00365 (10 ⁻²) 10.4724	0.01997+0.00379 (10 ²) 0.97825
DJI (751X6)	0.01413+0.00492 (10 ⁻¹) 10.8441	0.01419+0.0048 (10 ²) 1.39238
FCHI (763X6)	0.03166+0.01213 (10 ⁻²) 11.1665	0.03159+0.01216 (10 ²) 0.93741

(Continued)

Table 6. (Continued)

Dataset	SVR (C) Time	TSVR (C1 = C2) Time
GDAXI (755X6)	0.02591+0.00872 (10 ⁻²) 10.8492	0.02586+0.00873 (10 ⁻¹) 1.12026
GSPTSE (748X6)	0.02208+0.00768 (10 ⁻¹) 10.6209	0.02214+0.00779 (10 ⁻²) 1.28185
HSI (733X6)	0.02125+0.00607 (10 ⁻²) 10.2733	0.0212+0.00608 (10 ⁻¹) 1.26684
IBEX (763X6)	0.02829+0.00918 (10 ⁻²) 11.1037	0.02828+0.0091 (10 ⁻¹) 1.44011
IXIC (751X6)	0.0165+0.00475 (10 ⁻¹) 10.8561	0.01645+0.00473 (10 ⁻²) 1.10158
JKSE (724X6)	0.01871+0.0053 (10 ⁻¹) 10.4427	0.18938+0.36737 (10 ⁻⁵) 1.19995
KS11 (728X6)	0.02053+0.00366 (10 ⁻²) 10.1628	0.02052+0.00367 (10 ⁻²) 0.90443
MXX (750X6)	0.03059+0.00594 (10 ⁻¹) 10.6947	0.03052+0.006 (10 ⁻¹) 1.47527
N225 (732X6)	0.02757+0.01059 (10 ⁻¹) 10.2778	0.02753+0.01071 (10 ⁻¹) 1.14582
NSEI (731X6)	0.01992+0.00419 (10 ⁻¹) 10.1286	0.01994+0.00419 (10 ⁻¹) 1.18078
SSMI (750X6)	0.0402+0.0164 (10 ⁻¹) 10.8077	0.04008+0.01626 (10 ⁻¹) 1.32179
STOXX50E (745X6)	0.032+0.01324 (10 ⁻²) 10.5735	0.03193+0.01327 (10 ⁻¹) 1.1432
TWII (724X6)	0.02051+0.00474 (10 ⁻¹) 10.0368	0.02049+0.00477 (10 ⁻²) 1.23588

<https://doi.org/10.1371/journal.pone.0211402.t006>

methods and verify the significant critical difference between their average ranks. The difference between their ranks should be at least $1.645\sqrt{\frac{2 \times (2+1)}{6 \times 20}} \approx 0.3678$ at $p = 0.10$.

Since the difference between the average ranks of TSVR with SVR (1.95–1.05 = 0.90) is greater than 0.3678, we conclude that TSVR is significantly better than SVR for stock market index datasets. For the non-linear case, the absolute prediction error of SVR and TSVR is shown in Figs 9, 10 and 11 for the BVSP, DJI and IXIC datasets, respectively. The actual and predicted values of SVR and TSVR are plotted in Figs 12, 13 and 14 for the BVSP, DJI and IXIC datasets, respectively. It can easily be observed from these figures that TSVR is in close

Table 7. Average ranks of TSVR with SVR on stock market index datasets using a linear and Gaussian kernel.

Dataset	Linear		Non-Linear	
	SVR	TSVR	SVR	TSVR
AEX	2	1	2	1
ATX	2	1	2	1
BFX	1	2	2	1
BSESN	1	2	2	1
BVSP	1	2	2	1
DJI	1	2	2	1
FCHI	2	1	2	1
GDAXI	2	1	2	1
GSPTSE	1	2	2	1
HIS	2	1	2	1
IBEX	2	1	2	1
IXIC	2	1	2	1
JKSE	1	2	2	1
KS11	2	1	2	1
MXX	2	1	2	1
N225	2	1	2	1
NSEI	1	2	2	1
SSMI	2	1	1	2
STOXX50E	2	1	2	1
TWII	2	1	2	1
Average rank	1.65	1.35	1.95	1.05

<https://doi.org/10.1371/journal.pone.0211402.t007>

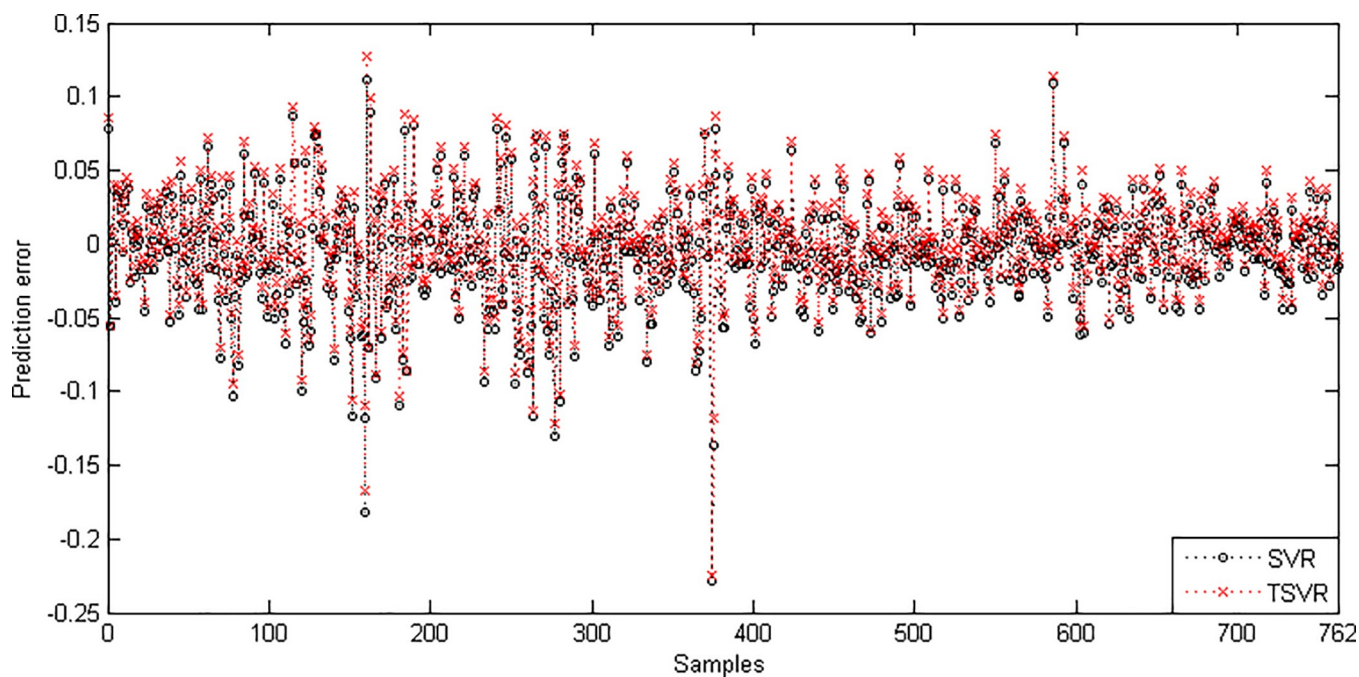


Fig 7. Prediction error plots using a linear kernel on the BFX dataset.

<https://doi.org/10.1371/journal.pone.0211402.g007>

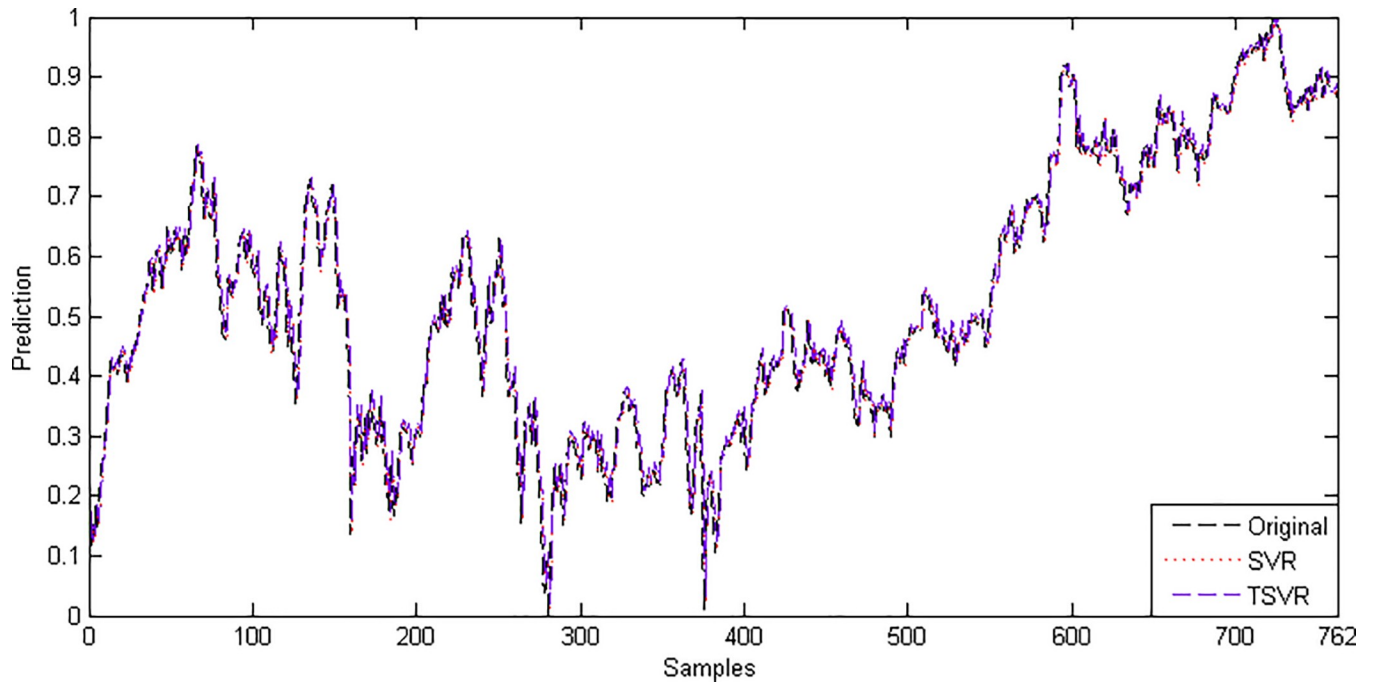


Fig 8. Predicted and actual values using a linear kernel on the BFX dataset.

<https://doi.org/10.1371/journal.pone.0211402.g008>

Table 8. Performance comparison of TSVR with SVR on stock market index datasets using a Gaussian kernel. RMSE is used for comparison. Time is used for the training in seconds.

AEX (763X6)	0.02765+0.01023 ($10^0, 2^{\wedge}-2$) 12.781	0.02698+0.0106 ($10^2, 2^{\wedge}-5$) 1.73396
ATX (737X6)	0.01949+0.00416 ($10^{\wedge}-1, 2^{\wedge}-2$) 11.907	0.01892+0.00422 ($10^2, 2^{\wedge}-5$) 1.41487
BFX (762X6)	0.03466+0.01048 ($10^{\wedge}-2, 2^{\wedge}-1$) 12.787	0.03395+0.0117 ($10^2, 2^{\wedge}-5$) 1.40597
BSESN (733X6)	0.02264+0.00551 ($10^0, 2^{\wedge}-2$) 11.8247	0.02073+0.00453 ($10^3, 2^{\wedge}-4$) 1.56612
BVSP (738X6)	0.0222+0.00447 ($10^{\wedge}-1, 2^{\wedge}-5$) 11.9909	0.02005+0.00391 ($10^2, 2^{\wedge}-5$) 1.43526
DJI (751X6)	0.01721+0.00561 ($10^0, 2^{\wedge}-5$) 12.3971	0.0155+0.00503 ($10^2, 2^{\wedge}-5$) 1.74354
FCHI (763X6)	0.03171+0.01203 ($10^{\wedge}-1, 2^{\wedge}-2$) 12.6618	0.03156+0.01218 ($10^2, 2^{\wedge}-5$) 1.53077
GDAXI (755X6)	0.02662+0.00822 ($10^{\wedge}-1, 2^{\wedge}-2$) 12.4439	0.02601+0.00867 ($10^2, 2^{\wedge}-5$) 1.49756
GSPTSE (748X6)	0.02627+0.0173 ($10^{\wedge}-1, 2^{\wedge}-2$) 12.2009	0.02301+0.00875 ($10^3, 2^{\wedge}-5$) 1.44924

(Continued)

Table 8. (Continued)

HSI (733X6)	0.02189+0.00633 (10 ⁻¹ ,2 ⁻²) 11.7218	0.02156+0.00623 (10 ² ,2 ⁻⁵) 1.38225
IBEX (763X6)	0.0285+0.00925 (10 ⁻¹ ,2 ⁻⁴) 12.6977	0.02842+0.00915 (10 ² ,2 ⁻⁵) 1.47464
IXIC (751X6)	0.01906+0.00513 (10 ⁰ ,2 ⁻⁵) 12.3108	0.01681+0.0047 (10 ² ,2 ⁻⁵) 1.69016
JKSE (724X6)	0.01922+0.00522 (10 ⁰ ,2 ⁻²) 11.3828	0.01893+0.00533 (10 ² ,2 ⁻²) 1.57983
KS11 (728X6)	0.02197+0.00448 (10 ⁻¹ ,2 ⁻⁴) 11.6135	0.02073+0.00373 (10 ² ,2 ⁻⁵) 1.34512
MXX (750X6)	0.03145+0.00605 (10 ⁰ ,2 ⁻⁵) 12.4093	0.03082+0.0058 (10 ² ,2 ⁻³) 1.73352
N225 (732X6)	0.02952+0.01077 (10 ⁻¹ ,2 ⁻⁵) 12.0234	0.02839+0.01029 (10 ² ,2 ⁻⁵) 1.3794
NSEI (731X6)	0.02206+0.00591 (10 ⁻² ,2 ⁻¹) 11.9166	0.02013+0.00444 (10 ³ ,2 ⁻⁵) 1.29858
SSMI (750X6)	0.04002+0.01628 (10 ⁻¹ ,2 ⁻⁴) 12.3919	0.04007+0.0161 (10 ² ,2 ⁻⁵) 1.42911
STOXX50E (745X6)	0.03218+0.01336 (10 ⁰ ,2 ⁻⁴) 12.4306	0.03204+0.01328 (10 ² ,2 ⁻⁵) 1.66912
TWII (724X6)	0.02084+0.0046 (10 ⁻¹ ,2 ⁻¹) 11.495	0.02057+0.00472 (10 ² ,2 ⁻⁵) 1.34332

<https://doi.org/10.1371/journal.pone.0211402.t008>

agreement with the desired output in comparison to SVR, which clearly demonstrates the applicability and usefulness of TSVR.

Conclusion

In this paper, support vector regression and twin support vector regression formulations are discussed in detail and applied to an individual companies' stock indices in the area of information technology industries, banking, oil, and petroleum industry and stock market index datasets of different countries to predict stock prices. Here, a pair of smaller sized QPPs is solved instead of a single large sized QPP, as in the case of SVR, thus yielding a reduction in the cost of the system. To verify the effectiveness of TSVR, we performed numerical experiments for both linear and Gaussian kernels on financial time series datasets. In experimental results, TSVR shows better learning speed for both linear and Gaussian kernels with the ability to predict having a better generalization ability than SVR. In fact, the computation time of the TSVR is approximately four times lower than the standard SVR in terms of learning speed, which clearly indicates its existence and usability. In future work, a new model that is able to handle noise and outliers for predicting the prices of stock indices can be explored.

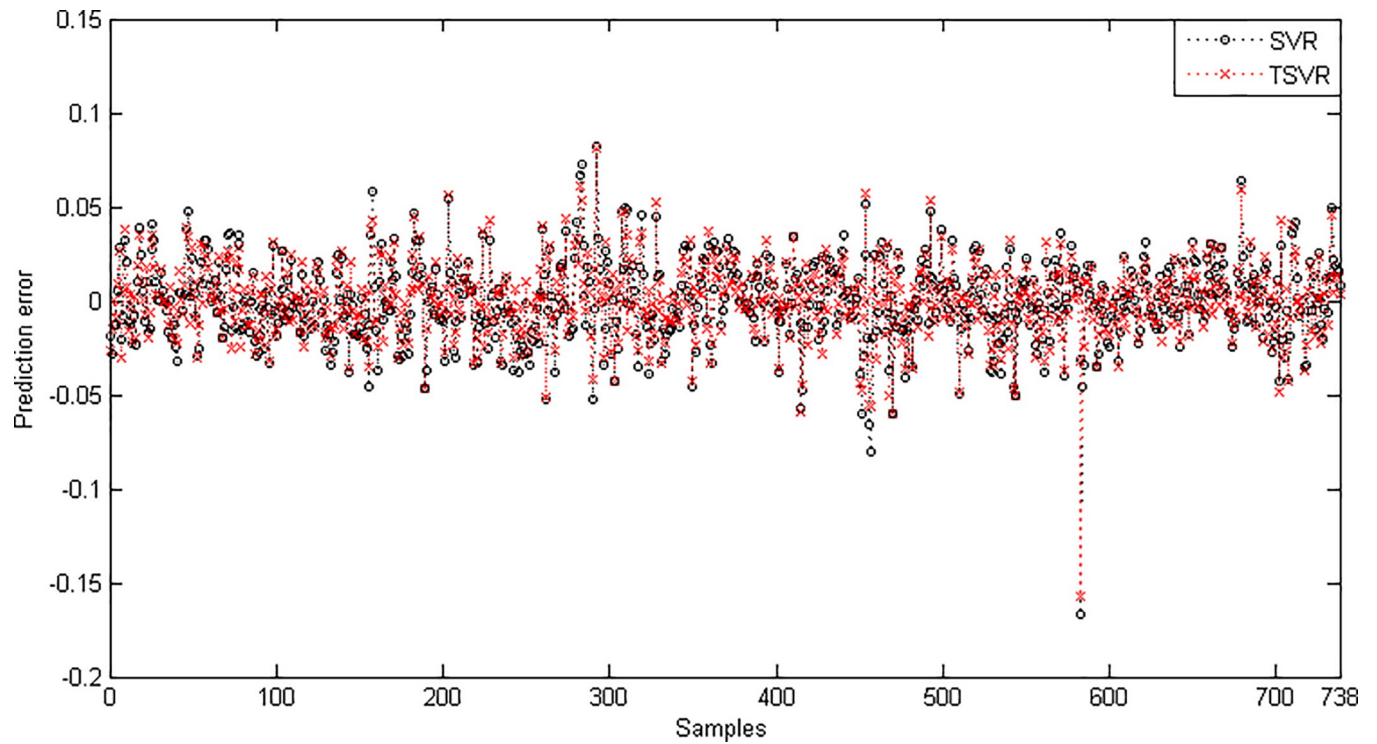


Fig 9. Prediction error plots using a Gaussian kernel on the BVSP dataset.

<https://doi.org/10.1371/journal.pone.0211402.g009>

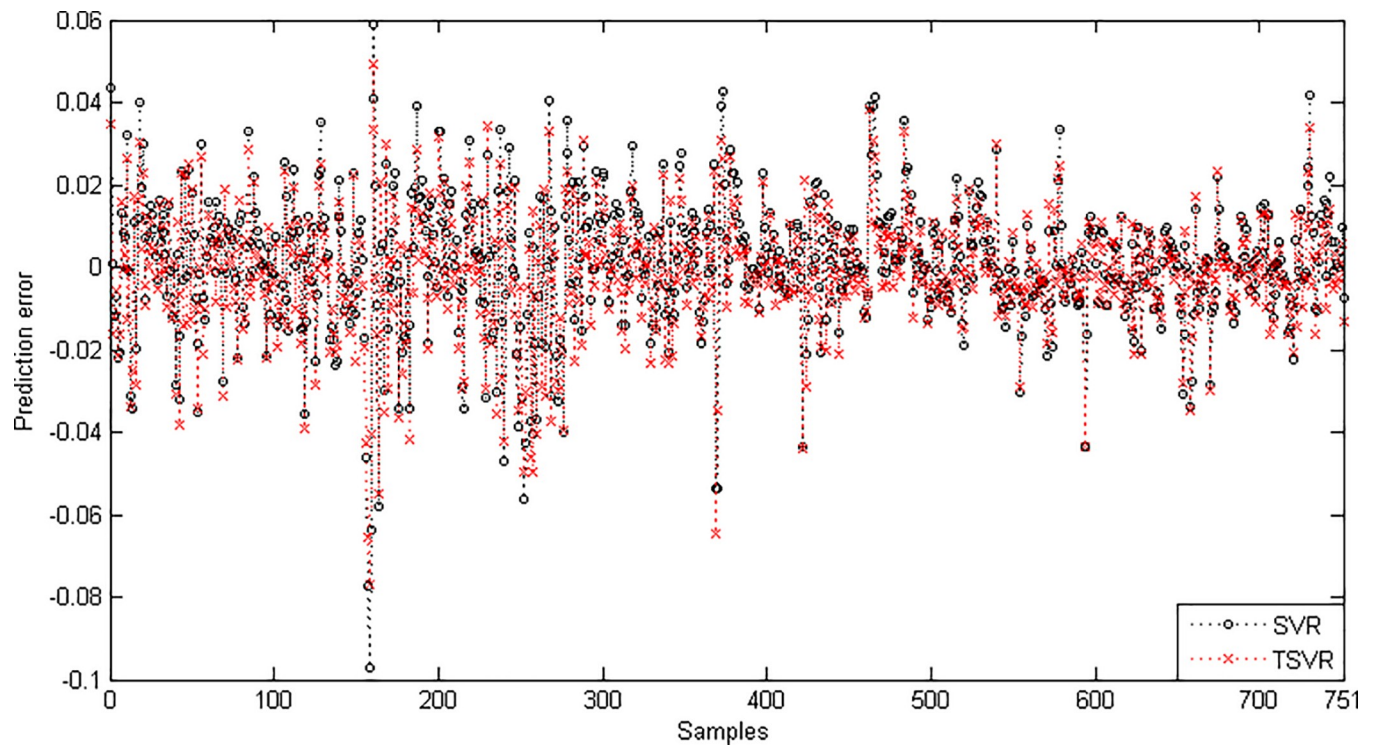


Fig 10. Prediction error plots using a Gaussian kernel on the DJI dataset.

<https://doi.org/10.1371/journal.pone.0211402.g010>

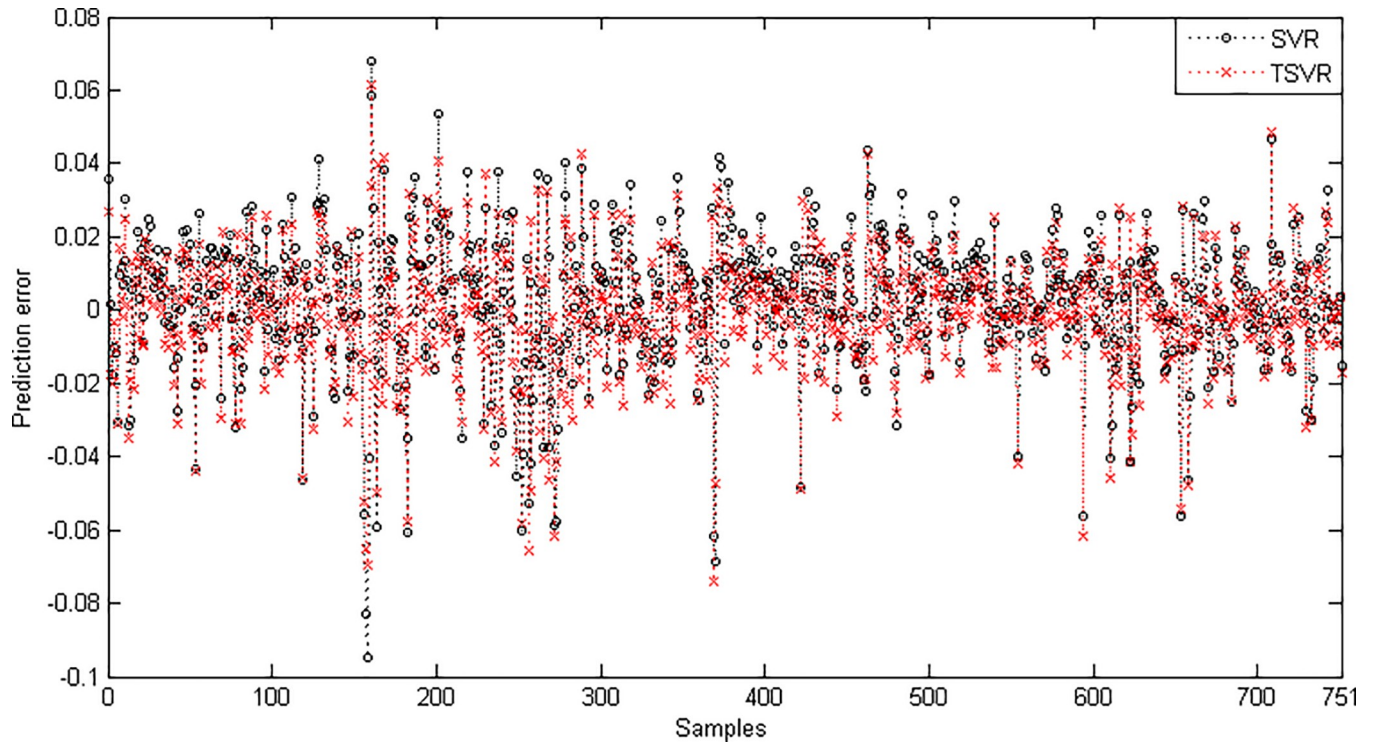


Fig 11. Prediction error plots using a Gaussian kernel on the IXIC dataset.

<https://doi.org/10.1371/journal.pone.0211402.g011>

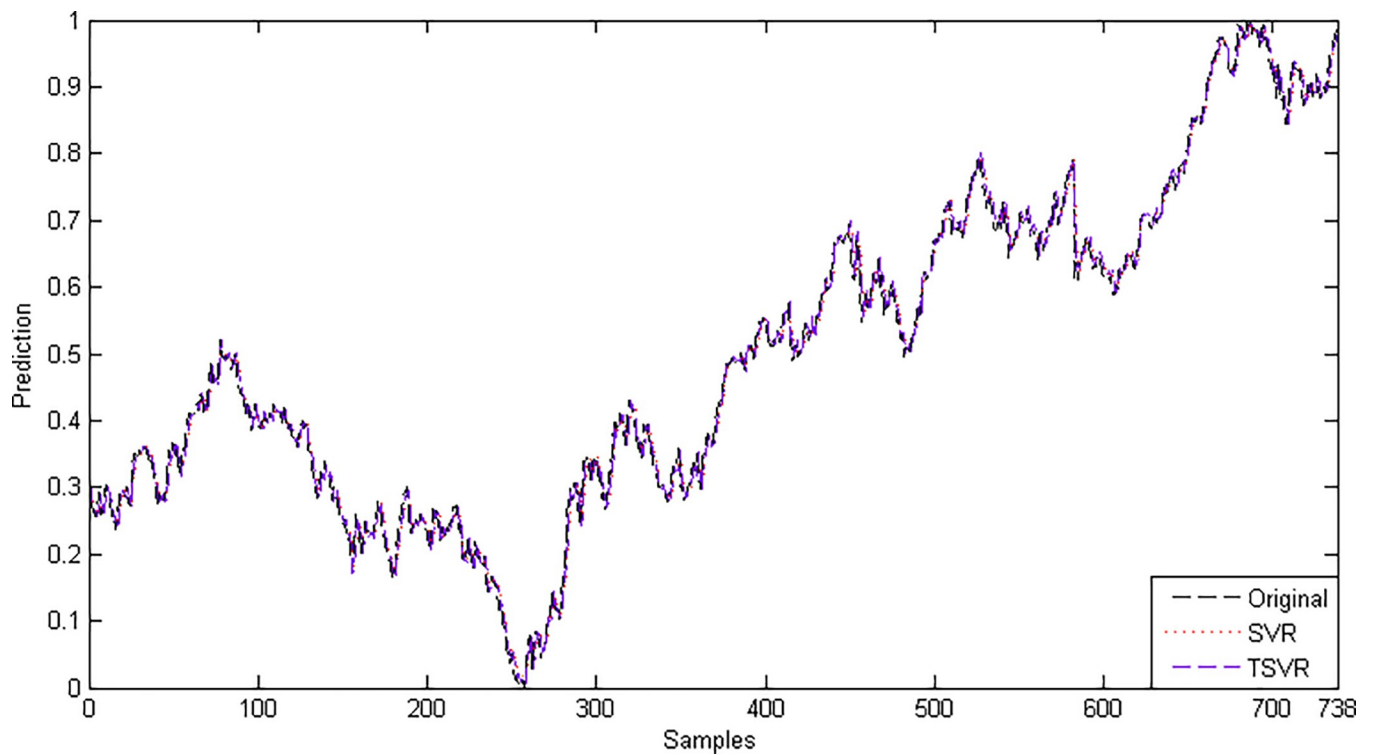


Fig 12. Predicted and actual values using a Gaussian kernel on the BVSP dataset.

<https://doi.org/10.1371/journal.pone.0211402.g012>

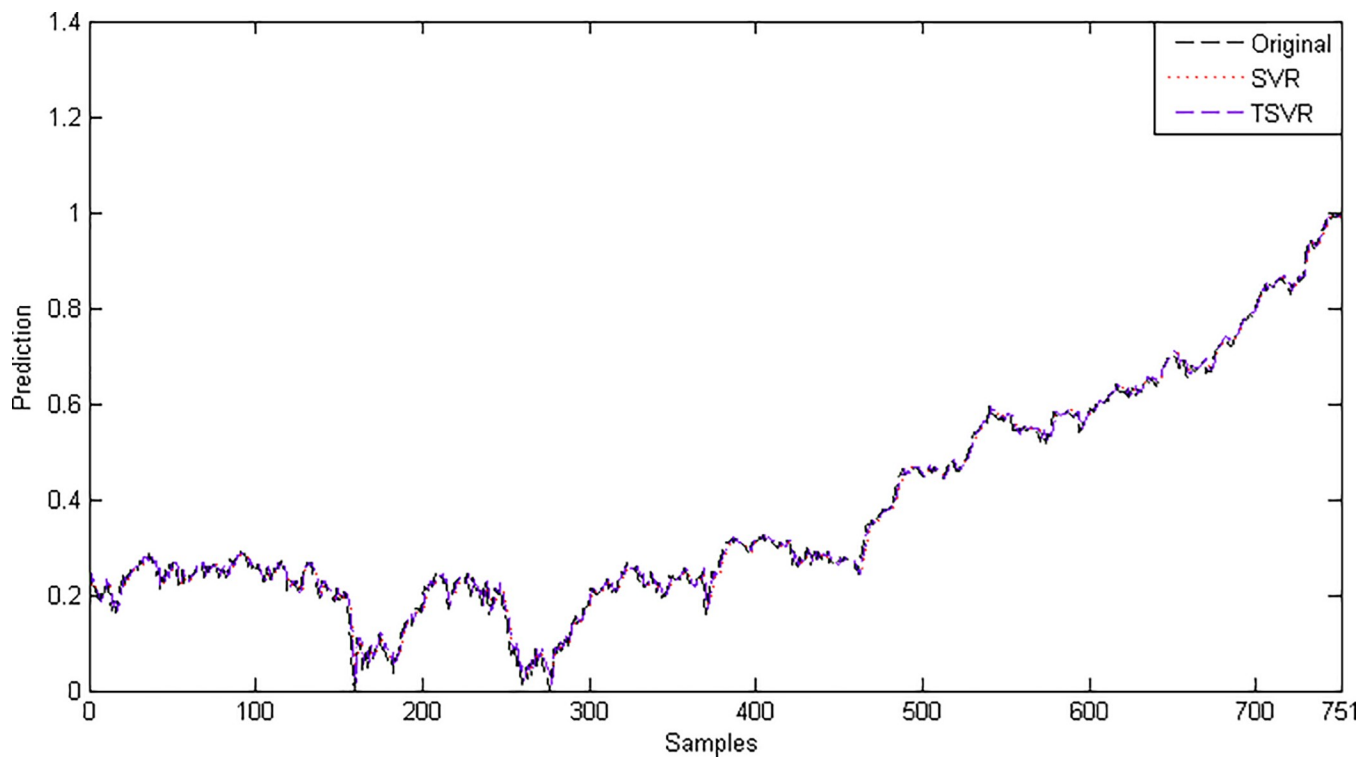


Fig 13. Predicted and actual values using a Gaussian kernel on the DJI dataset.

<https://doi.org/10.1371/journal.pone.0211402.g013>

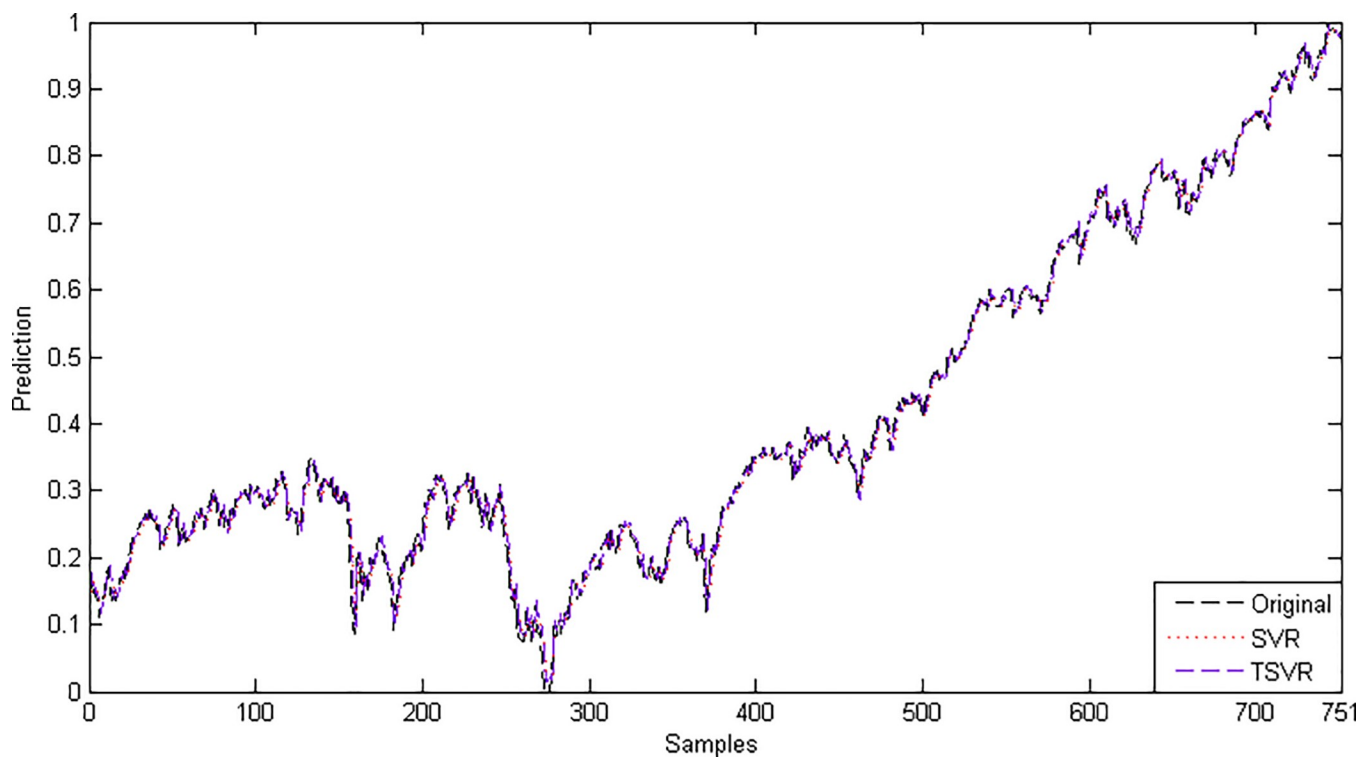


Fig 14. Predicted and actual values using a Gaussian kernel on the IXIC dataset.

<https://doi.org/10.1371/journal.pone.0211402.g014>

Author Contributions

Conceptualization: Deepak Gupta.

Data curation: Deepak Gupta.

Formal analysis: Mukesh Prasad.

Funding acquisition: Mahardhika Pratama.

Investigation: Deepak Gupta.

Supervision: Jun Li.

Validation: Mahardhika Pratama, Mukesh Prasad.

Visualization: Mukesh Prasad.

Writing – original draft: Deepak Gupta.

Writing – review & editing: Mahardhika Pratama, Zhenyuan Ma, Jun Li, Mukesh Prasad.

References

1. Vapnik VN. (2000). *The nature of statistical learning theory*, 2nd ed., Springer, New York
2. Osuna E, Freund R, Girosi F. (1997). Training support vector machines: An application to face detection, in *Proceedings of Computer Vision and Pattern Recognition*, 130–136
3. Huang C, Davis LS, Townshend JRG. 2002. An assessment of support vector machines for land cover classification. *International Journal of Remote Sensing*, 23, 725–749.
4. Joachims T. (1998). Text categorization with support vector machines: learning with many relevant features, In: *European Conference on Machine Learning No.10*, Chemnitz, Germany, 137–142
5. Brown MPS, Grundy WN, Lin D, Cristianini N, Sugnet CW, Furey TS, et al. (2000), Knowledge-based analysis of microarray gene expression data using support vector machine, *Proceedings of the National Academy of Sciences of USA*, 97(1), 262–267
6. Guyon I, Weston J, Barnhill S, Vapnik V. (2002). Gene selection for cancer classification using support vector machine, *Machine Learning*, 46, 389–422
7. Mukherjee S, Osuna E, Girosi F. (1997). Nonlinear prediction of chaotic time series using support vector machines, In: *NNSP'97: Neural Networks for Signal Processing VII: in Proc. of IEEE Signal Processing Society Workshop*, Amelia Island, FL, USA, 511–520
8. Muller KR, Smola AJ, Ratsch G, Schölkopf B, Kohlmorgen J. (1999). Using support vector machines for time series prediction, In: *Schölkopf B, Burges CJC, Smola AJ (Eds.), Advances in Kernel Methods—Support Vector Learning*, MIT Press, Cambridge, MA, 243–254
9. Lin F, Yeh C, Lee M. (2011). The use of manifold learning and support vector machines in the prediction of business failure, *Knowledge-Based Systems*, 24(1), 95–101
10. Boser BE, Guyon IM, Vapnik VN. (1992). A training algorithm for optimal margin classifiers. In *proceedings of the Annual Conference on Computational Learning Theory*, Haussler D, Ed., ACM Press, Pittsburgh, PA, pp. 144–152
11. Joachims T. (1999). Making large-scale SVM learning practical. In *Advances in Kernel Methods—Support Vector Learning*, Schölkopf B, Burges CJC, Smola AJ, Eds., MIT Press, Cambridge, MA, pp. 169–184,
12. Platt J. (1999). Fast training of support vector machines using sequential minimal optimization. In *Advances in Kernel Methods—Support Vector Learning*, Schölkopf B., Burge C.J.C., and Smola A.J., Eds., pp. 185–208, MIT Press, Cambridge, MA
13. Achlioptas D, McSherry F, Schölkopf B. (2002). Sampling techniques for kernel methods, In *Advances in Neural Information Processing Systems 14*, Dietterich T. G., Becker S., and Ghahramani Z., Eds., MIT Press, Cambridge, MA.
14. Fine S, Scheinberg K. (2001). Efficient SVM training using low-rank kernel representations, *Journal of Machine Learning Research*, 2:243–264
15. Tsang IW, Kwok JT, Cheung PM. (2005). Very large SVM training using core vector machines. In *proceedings of the Tenth International Workshop on Artificial Intelligence and Statistics*, Barbados.

16. Suykens JAK, Vandewalle J. (1999). Least squares support vector machine classifiers. *Neural Processing Letters*, 9(3), 293–300.
17. Suykens JAK, Lukas L, Van DP, Moor BD, Vandewalle J. (1999). Least squares support vector machine classifiers: a large scale algorithm, *European Conference on Circuit Theory and Design, (ECCTD'99)*, Stresa Italy, pp.839-842.
18. Mangasarian OL, Wild EW. (2006). Multisurface Proximal Support Vector Classification via Generalized Eigenvalues, *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 28(1), 69–74.
19. Jayadeva KR, Chandra S. (2007). Twin support vector machines for pattern classification, *IEEE Transaction on Pattern Analysis and Machine Intelligence (TPAMI)*, 29, 905–910.
20. Peng X (2010a). TSVR: An efficient twin support vector machine for regression, *Neural Networks*, 23 (3), 365–372
21. Emad WS, Danil VP, and Donald CW. (1998). Comparative study of stock trend prediction using time delay, recurrent and probabilistic neural networks, *IEEE Trans. on Neural Networks*, 9(6): 1456–1470. <https://doi.org/10.1109/72.728395> PMID: 18255823
22. Cao L, Tay FEH. (2001) Financial forecasting using support vector machines, *Neural Computing and Application*, 10: 184–192.
23. Kuo RJ, Chen CH, Hwang YC. (2001) An intelligent stock trading decision support system through integration of genetic algorithm based fuzzy neural network and artificial neural network, *Fuzzy Sets and Systems*, 118(1): 21–45.
24. Prasad M, Li DL, Lin CT, Singh J, Prakash S. (2015). Designing Mamdani-Type Fuzzy Reasoning for Visualizing Prediction Problems Based on Collaborative Fuzzy Clustering. *IAENG International Journal of Computer Science*, 42 (4).
25. Singh J, Prasad M, Prasad OK, Er MJ, Saxena A, Lin CT. (2016). A Novel Fuzzy Logic Model for Pseudo Relevance Feedback based Query Expansion. *International Journal of Fuzzy Systems*, 18 (6), 980–989.
26. Prasad M, Liu YT, Li DL, Lin CT, Shah RR, Kaiwartya OP. (2016). A New Mechanism for Data Visualization with TSK-type Preprocessed Collaborative Fuzzy Rule based System. *Journal of Artificial Intelligence and Soft Computing Research*.
27. Prasad M, Lin YY, Lin CT, Er MJ. (2015). A New Data-Driven Neural Fuzzy System with Collaborative Fuzzy Clustering Mechanism. *Neurocomputing*, 167, 558–568, 2015.
28. Lin CT, Prasad M, Saxena A. (2015). An Improved Polynomial Neural Network Classifier Using Real Coded Genetic Algorithm. *IEEE Transaction on Systems, Man and Cybernetics: Systems*, 45 (11), 1389–1401.
29. Prasad M, Lin CT, Hong CT, Chang JY. (2017). Soft Boosted Self Constructive Neuro Fuzzy Inference Network. *IEEE Transaction on Systems, Man and Cybernetics: Systems*, 47 (3), 584–588.
30. Kim KJ, Han I. (2000). Genetic algorithms approach to feature discretization in artificial neural networks for the prediction of stock price index, *Expert Systems with Applications*, 19: 125–132.
31. Hassan MR, Nath B. (2005). Stock market forecasting using hidden Markov model: a new approach, *Proc. 5th International Conference on Intelligent Systems Design and Applications*, 192–196.
32. Fama EF, French KR. Dividend yields and expected stock returns. *Journal of financial economics* 22.1 (1988): 3–25.
33. Lewellen J. "Predicting returns with financial ratios." *Journal of Financial Economics* 74.2 (2004): 209–235.
34. Goh JC, Jiang F, Tu J, Wang Y. Can US economic variables predict the Chinese stock market?. *Pacific-Basin Finance Journal* 22 (2013): 69–87.
35. Shen D, Zhang Y, Xiong X, Zhang W. Baidu index and predictability of Chinese stock returns. *Financial Innovation* (2017) 3:4.
36. Xiao L, Shen D, Zhang W. "Do Chinese internet stock message boards convey firm-specific information?" *Pacific-Basin Finance Journal* 49 (2018): 1–14.
37. Pissarenko D. (2002). Neural networks for financial time series prediction: Overview over recent research BSc (Hones) Computer Studies Thesis, University of Derby in Austria. <<http://citeseer.ist.psu.edu/pissarenko02neural.html>>.
38. <http://finance.yahoo.com>
39. <http://www.mosek.com>
40. Demšar J. Statistical comparisons of classifiers over multiple data sets. *Journal of Machine learning research* 7. Jan, pp. 1–30 (2006).