Bangladeshi Stock Price Prediction and Analysis with Potent Machine Learning Approaches

Chapter · July 2020

DOI: 10.1007/978-3-030-52856-0_18

6 authors, including:

Sajib Das
Daffodil International University
1 PUBLICATION **0** CITATIONS
SEE PROFILE

Md. Shohel Arman
Daffodil International University
20 PUBLICATIONS **29** CITATIONS
SEE PROFILE

Syeda Sumbul Hossain
Daffodil International University
20 PUBLICATIONS **28** CITATIONS
SEE PROFILE

Md. Sanzidul Islam
Daffodil International University
29 PUBLICATIONS **160** CITATIONS
SEE PROFILE

Bangladeshi Stock Price Prediction and Analysis with Potent Machine Learning Approaches

Sajib Das, Md. Shohel Arman(✉), Syeda Sumbul Hossain, Md. Sanzidul Islam, Farhan Anan Himu, and Asif Khan Shakir

Department of Software Engineering, Daffodil International University, Dhaka, Bangladesh
sshuvo27@gmail.com

Abstract. Stock price forecasting, is one of the most significant financial complexities, since data are not reliable and noisy, impacting many factors. This article offers a machine learning model for the stock price prediction using Support Vector Machine-Regression (SVR) with two different kernels which are Radial Basis Function (RBF) and linear kernel. This study shows the Prediction and accuracy comparison between Support Vector Regression (SVR) and Linear Regression (LR) and also the accuracy comparison for different kernels of Support vector Regression (SVR). The model has used sum squared error (SSE) to determine the accuracy of each algorithm; which has shown significant improvement than the other studies. This analysis is conducted on the price data of about five years of Grameenphone listed on Dhaka Stock Exchange (DSE). The highest accuracy was found with Linear Regression model in every case with the highest accuracy of about 97.07% followed by SVR (Linear) model and SVR (radial basis function) model with the highest accuracy rate of about 97.06% and 96.82%. In some cases the accuracy of SVR (radial basis function) was higher than SVR (linear). But it was the Linear Regression which had the highest accuracy of all in every case.

Keywords: Machine learning · Stock price prediction · Support vector regression · Linear regression

1 Introduction

The stock market refers to the selection and exchange of stocks in which common shares of public companies are bought, exchanged and issued. The shares of the company are all shares in which the company's ownership is split. In proportion to the number of shares in total, a single share of the stock represents fractional ownership. The prices of stocks shift with market forces every day. This means that share prices are changing due to demand and supply. If more people would like to purchase a stock (demand) than sell it (supply), the price will increase. On the other hand, if more people wanted to sell a stock than purchase it, there would be more supply than demand, and the price would fall [1]. In other words the more a stock has been transacted the more is valuable.

T. Bhuiyan et al. (Eds.): ICONCS 2020, LNICST 325, pp. 230–240, 2020.
https://doi.org/10.1007/978-3-030-52856-0_18

For years, stock price forecasts have focused because they can generate substantial profit. Investors have tried to predict the trends using various methods and bet in the markets. Technical analysis like RSI (Relative Strength Index), MFI (Money Flow Index), MACD (Moving Average Convergence/Divergence) etc. [2] and fundamental analysis like Investor sentiment analysis [3], EPS (Earnings per Share), Net asset value etc. are used in analyzing the trends of stock prices. Different machine learning algorithms have also used in forecasting stock prices. Tough it's a tricky task to as predict stock prices as the prices follow a random pattern.

We have collected the stock price data of GrameenPhone of about last five years (1/1/2014–21/11/2019) form Stock Bangladesh website (stockbangladesh.com). The dataset contains six columns named Date, Open, High, Low, Close and volume. Where the Date stands for a particular date, the Open indicate the opening price of a stock on the particular date, the High and Low stands for the highest and lowest trading price of on that day. The Close indicates the closing price of the day and the volume indicates the numbers of share transacted on the particular day.

We have evaluated the performance of SVR (Support vector machine-Regression) with Linear & Radial Basis Function (RBF) kernels and Linear Regression with the previous price data.

2 Literature Review

Various algorithms for machine learning are used to predict stock price trends. Some of them are ANN (Artificial Neural Networks) [4–7], GA (Genetic Algorithm) [6], LS-SVM (Least Square Support Vector Machine) [2, 8], Trend Estimation with Linear Regression (TELR) [9], SVM (support vector machines) [5, 7, 10] with different kernels, KNN (K Nearest Neighbors) [7] structural support vector machines (SSVMs) [11]. Some statistical analyses are also used like Autoregressive Integrated Moving Average (ARIMA) [12]. But none of them were able to give quite promising prediction due to the non-linearity of the data.

Studies have tried to predict stock prices using Artificial Neural Network. In a study in 2017 [5] on Korean stock market ANN was used to predict the stock price [13] with the highest accuracy of 81.34% for 20 days and 83.01% for 30 days moving average. In another study [7] ANN was used and got 86.69% average accuracy based on experiment carried out on three different stocks. In 2018 this study [4] ANN was used and the best SSE score was 0.6271104815 for Apple, 0.0121281374 for Pepsi, 0.0335425612 for IBM, 0.016770174 for McDonald and 0.0211154625 for LG. In this study [5] author used ANN and achieved 96.10% accuracy for 5-fold average prediction (Mj'). Mj is a model where $1 \leq j \leq J$, for each classifier. And Mj' is a variant model as a substitution of Mj. 92.81% accuracy was achieved from model Mj.

Least square support vector machine (LS-SVM) is another popular machine learning used to predict stock prices. In this study [8] author a LS-SVM model along with Particle swarm Optimization (PSO) and the accuracy rate of the system was around 90.5–93%. Another research [1] used LS-SVM with PSO optimization and LS_SVM to predict stock prices. The Mean Squared Error (MSE) was 0.5317 for Adobe, 0.6314 for oracle 0.7725 for HP and 0.7905 for American Express with PSO-LS-SVM where MSE was 0.5703 for Adobe, 0.8829 for Oracle 1.2537 for HP and 1.0663 for American Express.

Other potent machine learning algorithms like SVM was used in this study [7] and found average accuracy of 89.33% after applying the algorithm for three company stocks. For BSE-Sensex the accuracy was 90.10% using polynomial kernel where c = 1 and degree = 1 and for RBF kernel the accuracy was 88.08 where c = 1 gamma = 4. For Infosys the accuracy was 89.59% with polynomial kernel where c = 1 and degree = 1 and 87.80% for RBF kernel where c = 5 and gamma = 1.5. Another research [10] that used SVM with RBF kernel has the mean accuracy of 53.3% to 56.8% for 10 days. And the accuracy rate can be lower than 50% for different dataset. A modified linear regression algorithm Trend Estimation with Linear Regression was used in this study [8] and Mean Absolute Percentage Error (MAPE) was 5.41% for bank data and 5.42% for overall stock data. K Nearest Neighbors or KNN is also been used to predict stock prices. A study that used KNN [6] has an accuracy of 83.52% with KNN. Structural support vector machines (SSVMs) is used in a research and the accuracy with training samples was higher than 78% and the accuracy with testing samples was about 50%.

3 Methodology

As our Fig. 1 shows our proposed model has five stages. First we collected the data and pre-process it. Then we optimized the parameters that we are going to use for our training algorithm. Then we tested our dataset with Linear Regression and support vector regression with 2 different kernels using the parameters that we optimized. Then we extracted our output and tested the accuracy of our algorithms with SSE. Finally we visualized the comparison between the actual values and the predicted values.

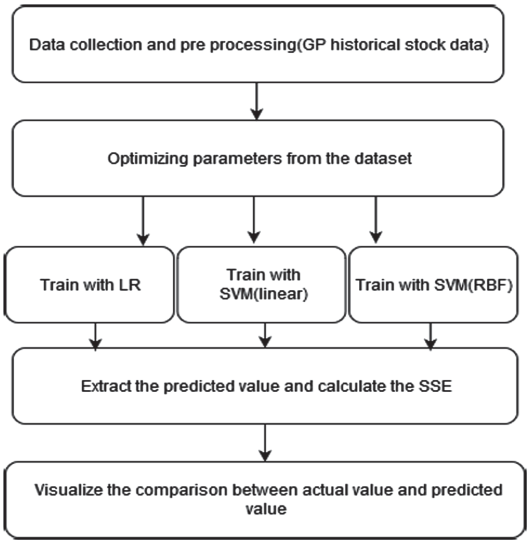

Fig. 1. The proposed model

3.1 Data Collection and Pre-processing

We collected the data form stockbangladesh.com. The website is an open source website and contains the previous price data of all the companies listed in Dhaka Stock Exchange. We have collected the price data of Grameenphone from 1st January 2014 to 21st November 2019 where the data size was 1413 meaning that we have the price information of 1413 days for the stock. We got the dataset in the recent price to the oldest price form. To Process the data to our desired order we had to flip the data to get the oldest price to recent price order. We checked for null values; there was none so we used the dataset as it was (Fig. 2).

Date	Open	High	Low	Close	Volume
1/1/2014	79.5	81.6	79	79.5	210400
2/1/2014	82.5	85.6	80.5	83.8	751200
6/1/2014	85.4	85.4	82.1	83.9	194800
7/1/2014	79.5	85.8	75.5	85.2	377400
8/1/2014	85.9	89	84.9	88	641200
9/1/2014	88.3	90.4	85	86.2	378800
12/1/2014	87.9	89.3	86.5	87.4	368800
13/01/201	87.5	89.3	83.2	83.4	493400
15/01/201	83.4	84.8	81	82	540200

Fig. 2. Sample data of GP

3.2 Optimizing Parameters from the Dataset

We have used the Close column of our dataset as the input parameter to predict the stock prices. The close prices for different dates were used as the dependent variable to predict stock prices. We have split the data into 80% test data and 20% train data and stored them into different variables to use them as parameters.

3.3 Linear Regression

We have used linear regression algorithm to train our model. Linear regression is a statistical method for modeling a relationship between two variables that corresponds to the observed data on a linear equation. One variable is regarded as an explanatory variable and the other as a dependent variable. The linear regression method can be used to predict under the assumption that the correlation between the variables will continue in the future. A linear regression Eq. (1) is as follows

$$Y_i = \beta_0 + \beta_1 X_i + \varepsilon_i \tag{1}$$

Here, Y_i is the outcome (predicted output) of the dependent variable for the i^{th} test unit, X_i is the independent variable which is used for prediction for i^{th} test unit. $\beta_0 + \beta_1 X_i$ is the linear relation between Y_i and X_i β_0 is the intercept or the mean of Y when X = 0 and β_1 is the slope or the change in mean when X increases by 1. The ε_i represents the error term. In Our model we have used the data close column as our independent variable. So in our model $\text{Close}_i = X_i$.

The β_0 and β_1 parameters are unknown. They are estimated by using least square method. The least square method (2) is as follows:

$$b = \frac{n(\sum XY) - (\sum X).(\sum Y)}{n(\sum X^2) - (\sum X)^2} \tag{2}$$

Considering the least square the best fitted line is taken and uses that as function for the prediction.

3.4 Support Vector Regression

Support Vector Regression (SVR) has the same principal that Support Vector Machine (SVM) uses except for a few minor differences. At first, as output is a real number, the information at hand which has endless possibilities becomes very hard to predict. In the case of regression, the SVM is approximated by a range of tolerance (epsilon), which would already have requested the problem. The main idea, however, is always the same: to mitigate error, to individualize the hyperplane which maximizes the margin, taking into account which part of the error is tolerated.

The fundamental idea behind SVM is for training information from the input field to be transformed into a higher dimension of function Φ and then a separating hyperplane with maximum margin in the function space is constructed as shown in Fig. 3.

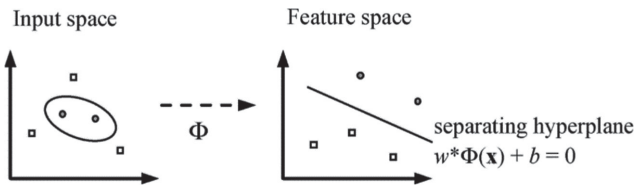

Fig. 3. Using a kernel function SVM mapped the data into a higher dimensional space and separated them using a hyperplane.

When it comes to SVR we can consider a set of training data $\{(x_1, y_1) \ldots (x_l, y_l)\}$ where each $x_i \subset R^n$ which represents the sample input space and has an adequate target value $y_i \subset R$ for i = 1,…,l, where l is the size of training data [14]. The standard SVR [16] estimation function (1) is as follows:

$$f(x) = (w \cdot \Phi(x)) + b \tag{3}$$

Where $w \subset R_n$, $b \subset R$ and Φ indicates a nonlinear conversion from R_n to a high-dimensional space. Our objective is to find the value of w and b so that we can determine

values of x by reducing the risk of regression.

$$R_{\text{reg}}(f) = C\sum_{i=0}^{l}\Gamma(f(x_i) - y_i) + \frac{1}{2}\|w\|^2 \tag{4}$$

Here Γ() is a cost function C is a constant and the data points vector w may be formulated as:

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

The most widely used cost function is the $\mathscr{E}$-insensitive loss function [15].The function is in this following form:

$$\Gamma(f(x) - y)) = \begin{cases} |f(x) - y| - \varepsilon \\ 0, \end{cases} \tag{6}$$

The regression risk in (4) and the -insensitive loss function (6) can be minimized by solving the quadratic optimization problem.

ζ is a slack variable used to calculate errors outside $\mathscr{E}$ tube. In Fig. 4 SVR is fitting a tube with radius $\mathscr{E}$ to the data and positive slack ζ is measuring the points that are outside the tube

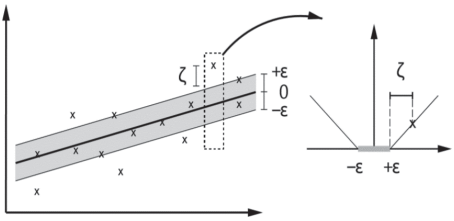

Fig. 4. The configuration of the soft margin loss applies to a linear SV system

The standard formula can be rewritten by substituting (5) to (3):

$$\begin{aligned} f(x) &= \sum_{i=1}^{l}(\alpha_i - \alpha_i^*)(\Phi(x_i)\cdot\Phi(x)) + b \\ &= \sum_{i=1}^{l}(\alpha_i - \alpha_i^*)k(x_i, x) + b. \end{aligned} \tag{7}$$

In (6) the function k (x_i,x) can substitute the dot product known as the kernel function. The kernel function is the idea is to map the non-linear data set into a higher dimensional space where a hyperplane can be found that separates the samples. There are different types of kernels in SVM. The kernels that we used for our model are Radial basis function kernel of RBF and linear kernel.

3.4.1 Radial Basis Function Kernel

The Radial basis function kernel, or RBF kernel also known as Gaussian kernel, is a kernel that is in the form of a radial basis function. The RBF kernel on two samples x and x′, interpreted in some input space as feature vectors, is defined as follows:

$$K(\mathrm{x}, \mathrm{x}') = \exp\left(\frac{\left\|\mathrm{x} - \mathrm{x}'\right\|^2}{2\sigma^2}\right) \tag{8}$$

It can also be interpreted as:

$$K(\mathrm{x}, \mathrm{x}') = \exp(-\gamma \left\|\mathrm{x} - \mathrm{x}'\right\|^2) \tag{9}$$

Radial basis function takes parameters like gamma and c. The parameter gamma describes the degree to which the effect of a single example of learning exceeds. The C parameter works against maximizing the margin of the decision function from accurate identification of training instances. We have set the values of our parameters by gamma = 0.1 and c = le3.

3.4.2 Linear Kernel

The Linear kernel function is the simplest kernel function. The function is given by the internal product (x, y) and an optional constant C. Algorithms using linear kernel functions are often the same as their non-kernel counterparts. It can be interpreted as:

$$k(x, y) = x^T y + c \tag{10}$$

3.5 Extracting Predicted Value and Calculating SSE

After training our dataset with the algorithms we extract the forecasted values that our algorithms predicted. We calculated the accuracy by calculating the SSE where the value is between 0 to 1. 1 is the best possible value that we can get meaning an accuracy of hundred percent. The formula for calculating SSE is:

$$SSE = \sum_{i=1}^{n} (Y_i - \hat{Y}_i)^2 \tag{11}$$

Where $\mathrm{Y_i}$ is the actual value and $\hat{Y}_i$ is the predicted value.

3.6 Visualizing the Comparison Between Actual and Predicted Values

Finally we plotted our value on a graph and compared the predicted values in respect of the actual values. We have compared the values for 10 days, 20 days and 30 days for linear regression and for SVR with both the RBF and linear kernels.

4 Result and Discussion

After training and test our model we have extracted the predicted value for different days. We have predicted the closing price for a 10 days, 20 days and 60 days period. The accuracy for predicting the price for less number of days was much higher than the accuracy for a higher number of days. The accuracy and predicted price changes after each iteration. We have discussed about the accuracy and the predicted price that we got for a random iteration. After running our program we found an accuracy of 97.07% with linear Regression 97.06 with SVR (RBF) and 96.82 with SVR (linear) for 10 days. We have plotted the comparison between the actual value and predicted value using graphical representations. We have used matplotlib; a python library to visualize the graph for the values. The 10 day comparison graph among the Linear Regression, SVR (RBF) and SVR (linear) is shown below:

Figure 5 shows the comparison between actual value and forecasted value for 10 days with LR. It has a 94.07% accuracy which is pretty high for a continuous valued prediction. The graph shows that the predicted value is almost as same as the actual value which is pretty impressive. In Fig. 6 we can see the graphical comparison between the actual and predicted value for SVR (linear). The accuracy is 97.06% which is pretty similar to the LR prediction and the predicted values are very close to the actual values. In Diagram 6 the graph shows the comparison between actual and predicted values with SVR (RBF). The patterns are quite close for actual and predicted values but still not as accurate as LR and SVR (linear). The accuracy is 96.86% which is still pretty high (Fig. 7).

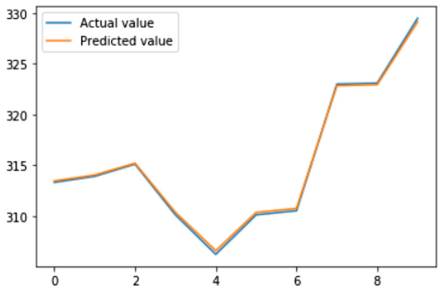

Fig. 5. Comparison for 10 days (LR)

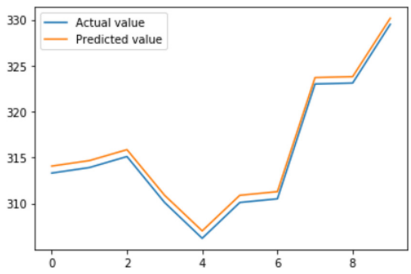

Fig. 6. Comparison for 10 days SVR (linear)

We have predicted the closing price for 30 days period too. For 30 days The Linear regression performed the best with the highest accuracy of 91.22% while the SVR (linear) has an accuracy of about 90.70% and SVR (RBF) has an accuracy of about 87.50%. The accuracy for all of the algorithms are still quite good.

Figure 8 shows that the actual and predict values are very close. Figure 9 shows the predicted prices followed the same trend as the actual prices. In Fig. 10 with SVR (RBF) the prices wasn't quite accurate but still pretty good for a continuous valued prediction.

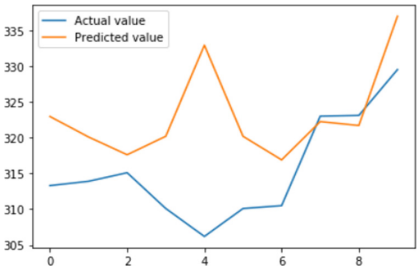

Fig. 7. Comparison for 10 days SVR (RBF)

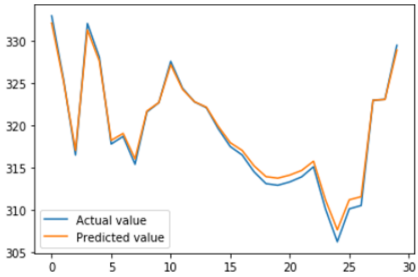

Fig. 8. Comparison for 30 days (LR)

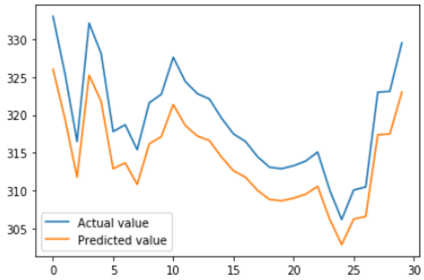

Fig. 9. Comparison for 30 days SVR (linear)

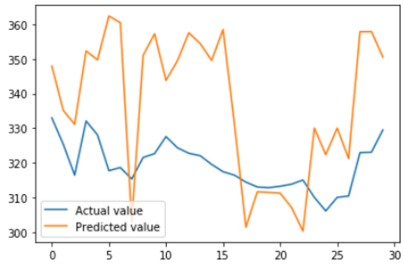

Fig. 10. Comparison for 30 days SVR (RBF)

For 60 days prediction like other prediction LR has the best performance with an accuracy of 79.82% while SVR (RBF) has a better accuracy than SVR (linear) with 78.50% and SVR (linear) have an accuracy of 77.53%. The comparison in graphical representation is shown below:

In Fig. 11 we can see that the predicted price trend followed the actual price trend. The accuracy of linear is much lower than the 10 days or 30 days prediction accuracy but it's still pretty close. For SVR (linear) the trend is similar but the values are a little far from the actual price point. For SVR (RBF) the price point are pretty close in some points but still has a lower accuracy than LR (Figs. 12 and 13).

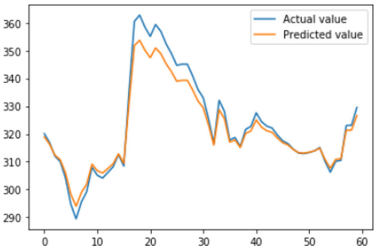

Fig. 11. Comparison for 60 days (LR)

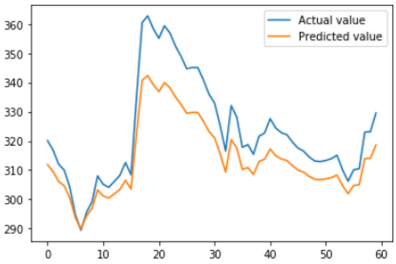

Fig. 12. Comparison for 60 days SVR (linear)

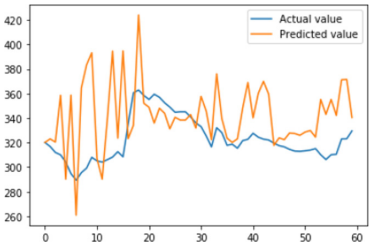

Fig. 13. Comparison for 60 days SVR (RBF)

The accuracy comparison for 10, 30 and 60 days are given at the table:

Days	LR	SVR (linear)	SVR (RBF)
10	97.07%	97.06%	96.82%
30	91.22%	90.70%	87.50%
60	79.82%	77.53%	78.50%

As per our table we can see that Liner Regression algorithm has the best performance among all. SVR (linear) and SVR (RBF) have pretty much similar performance with varying performance for different days.

5 Conclusion

Linear Regression model has performed the best to predict stock price. For fewer days it has a tremendous performance. SVR (linear) and SVR (RBF) has a quite impressive performance too. But with other parameters the performance of SVR may be improved. Using technical and fundamental indicators like RSI, MACD, investor sentiment percentage, company background etc. as parameters might improve the prediction performance as these have an effect on stock price movement. Using these parameters in potent machine learning algorithms might increase the accuracy of price prediction.

References

1. David, R.H.: Forces That Move Stock Prices. https://www.investopedia.com/articles/basics/04/100804.asp. Accessed 20 Nov 2019
2. Hegazy, O., Soliman, O.S, Salam, M.A.: A machine learning model for stock market prediction. arXiv preprint arXiv:1402.7351 (2014)
3. Guo, K., Sun, Y., Qian, X.: Can investor sentiment be used to predict the stock price? Dynamic analysis based on China stock market. Phys. A **469**, 390–396 (2017)
4. Ebadati, O.M.E., Mortazavi, M.T.: An efficient hybrid machine learning method for time series stock market forecasting. Neural Netw. World **28**(1), 41–55 (2017)
5. Pyo, S., Lee, J., Cha, M., Jang, H.: Predictability of machine learning techniques to forecast the trends of market index prices: hypothesis testing for the Korean stock markets. PLoS ONE **12**(11), e0188107 (2017)
6. Qian, B., Rasheed, K.: Stock market prediction with multiple classifiers. Appl. Intell. **26**(1), 25–33 (2007)
7. Patel, J., Shah, S., Thakkar, P., Kotecha, K.: Predicting stock and stock price index movement using trend deterministic data preparation and machine learning techniques. Expert Syst. Appl. **42**(1), 259–268 (2015)
8. Akash, A., Rajaji, S., Aravinth, R., Vendhan, V., Veerapandi, D.: Stock market trend prediction using machine learning. Int. J. Innov. Res. Comput. Commun. Eng. **7**, 1000–1006 (2017). https://doi.org/10.15680/ijircce.2019.0702085
9. Efat, M.I.A., Bashar, R., Uddin, K.I., Bhuiyan, T.: Trend estimation of stock market: an intelligent decision system. In: International Conference on Cyber Security and Computer Science (2018)
10. Madge, S., Bhatt, S.: Predicting stock price direction using support vector machines. Independent work report spring (2015)
11. Leung, C.K.S., MacKinnon, R.K., Wang, Y.: A machine learning approach for stock price prediction. In: Proceedings of the 18th International Database Engineering & Applications Symposium, pp. 274–277. ACM (2014)
12. Müller, K.-R., Smola, A.J., Rätsch, G., Schölkopf, B., Kohlmorgen, J., Vapnik, V.: Predicting time series with support vector machines. In: Gerstner, W., Germond, A., Hasler, M., Nicoud, J.-D. (eds.) ICANN 1997. LNCS, vol. 1327, pp. 999–1004. Springer, Heidelberg (1997). https://doi.org/10.1007/BFb0020283
13. Kamalakannan, J., Sengupta, I., Chaudhury, S.: Stock market prediction using time series analysis. In: 2018 IADS International Conference on Computing, Communications & Data Engineering (CCODE), pp. 7–8 (2018)
14. Müller, K.R., Smola, A., Rätsch, G., Schölkopf, B., Kohlmorgen, J., Vapnik, V.: Using support vector machines for time series prediction. In: Advances in Kernel Methods—Support Vector Learning, pp. 243–254 (2018)
15. Edwards, R.D., Magee, J., Bassetti, W.C.: Technical Analysis of Stock Trends. CRC Press, Boca Raton (2018)
16. Cherkassky, V., Ma, Y.: Practical selection of SVM parameters and noise estimation for SVM regression. Neural Netw. **17**(1), 113–126 (2004)