SN Bus Econ (2021) 1:58
https://doi.org/10.1007/s43546-021-00065-6

ORIGINAL ARTICLE

Predicting the daily closing price of selected shares on the Dhaka Stock Exchange using machine learning techniques

Sharmin Islam[1] · **Md. Shakil Sikder**[1] · **Md. Farhad Hossain**[2] · **Partha Chakraborty**[3]

Received: 25 June 2020 / Accepted: 25 February 2021 / Published online: 24 March 2021

Abstract

One of the most challenging topics in financial market analysis is predicting stock prices. Factors like supply and demand in the market, market sentiment, and investor's expectations, economic and political shocks can affect stock prices. All these factors make stock prices volatile and chaotic. Several machine learning models have been developed to make more precise and accurate predictions. Support vector regression (SVR) and K-nearest neighbor (KNN) regression are the most popular machine learning techniques used for stock price prediction. Our study follows the hypothesis that the SVR algorithm is a more precise way of predicting the Dhaka Stock Exchange (DSE) than the KNN regression. This research has made predictions and compared prediction errors between SVR and KNN. The analysis has been conducted on recent years' data of some selected shares listed on the DSE. The performance of the models is measured in terms of their root mean squared error (RMSE), *R*-squared (R^2), adjusted *R*-squared score values. We have optimized our model performance by tuning different combinations of hyper-parameters. The best result was found with the linear SVR model in the case of BXPHARMA with the highest R-squared score of about 97.04%, and lowest RMSE of about 1.23, followed by the KNN regression model with an *R*-squared score of approximately 96.39% and RMSE of about 1.38. SVR has the lowest RMSE and highest *R*-squared values in all cases.

Keywords Stock market · Machine learning techniques · Regression · Linear SVR · KNN

✉ Sharmin Islam
sharminislameity@gmail.com

Extended author information available on the last page of the article

Introduction

Financial time series predicting is one of the most challenging applications of modern time series forecasting (Abecasis et al. 1999; Cao and Tay 2003). A company's stock or share or equity is a financial instrument representing the holder has proportionate ownership of issuing an organization's assets and its profit. A stock market is a place where investors can buy and sell ownership of such investible assets. It is the most profitable investment sector in the field of the financial market. In addition, as profits and risks are proportional, the higher the profits, the greater the risks. Stock markets are now an essential part of our financial system. The impact of stock markets on the economy is extensive. Making precise inferences about the future price of the stock market has apparent benefits. Supply and demand affect the price of a stock. If there is a greater supply (sell) of stock than demand (buy), then the stock price goes down. Conversely, if a bigger number of people buy a stock than sell it, there would be greater demand than supply, and the price of the stock would increase. There are many reasons behind the rise or fall of the supply and demand of a stock such as market sentiment and investor's expectations, economic and political shocks, which make the prices unstable and imbalanced. It is extremely difficult to take into consideration all those factors that can influence stock (Vainionpää and Davidsson 2014). For example, internal development, world events, inflation and interest rates, exchange rates, and lastly hype, etc. can affect a stock. However, the most important variable that is highly correlated with the next day's stock prices is A day's avg. price, Day's High, Day's Low, Year high, YTD change, P/E ratio, etc. (Mbeledogu 2012) and Open, High, Low and Close prices that are recorded on daily basis have a higher informational content that other intraday prices. By using computational intelligence, machine learning, and data mining that find correlations in large datasets, where humans are not capable of doing it, which can be used as a prediction method for stock prices as well as another uncertain sector in finance. Machine learning can derive hidden patterns from messed-up data. In this era of artificial intelligence, applications of machine learning and deep learning are ubiquitous. Smart phones are getting more powerful with limited resources; self-driving cars are already on the streets. Many multinational investment banks like JPMorgan Chase and Morgan Stanley now have their Machine Learning department to help them make appropriate decisions. So far, there has been much study on stock market prediction. It is extremely tough to take into account all the factors which influence stock prices. For this reason, developing the best model is a matter of decision. However, opening price, closing price, day's high, and day's low recorded daily are the essential variables that are positively correlated with the next day's stock prices and have higher information than others. Several methods, including statistical methods, machine learning models, and deep learning models, are used to predict stock prices. In this study, we compare K-Nearest Neighbor Regression (KNN Regression) and Support Vector Regression (SVR). The algorithms are used historical data to train a model that is expected to infer future prices

given recent price information. This study has been conducted on three well-known Dhaka Stock Exchange (DSE), named BXPHARMA, SQPHARMA, and GP Company Limited. The most recent historical time series data are collected from the DSE (01-Jan-2016 to 31-Jan-2019) daily for experimentations. We have taken the closing price as the target variable. Except for the volume variable, we have taken the rest as independent variables. In the KNN Algorithm case, we have determined the appropriate value of K through the iteration process. In the SVR Algorithm, we have made models through different combinations of hyper-parameters and made predictions. We have verified the RMSE of each of them. Models are trained to make short-term predictions for one day on daily stock exchange data. The main contribution of this research work is using support vector regression (SVR) and K-nearest neighbor (KNN) regression for stock price prediction as well as optimizing the model performance by tuning different combinations of hyper-parameters. Finally, finding the best result with the linear SVR model compared to other non-linear SVR models whereas models are trained to make short-term predictions for one day on daily stock exchange data.

Review of literature

In recent years (Tsai and Wang 2009; Jaman et al. 2009), many types of research on share market prediction have been performed using computational intelligence methods with higher prediction results. Neural networks and support vector machines are the most useful among those (Chen and Ho 2005). The stock market prediction has been an essential exertion in business and finance for many years. Many existing studies on machine learning models, deep learning models, time-series models, and comparative studies have been conducted to predict the stock prices. Wang (2014) used SVM to predict S&P 500 stock prices. They showed that SVM performs better than standard statistical methods. Chen et al. (2006) used Neural Networks to predict the SP 500 Index's future values. Their paper compares Neural Networks performance against an ARIMA model, and the Neural Network model outperformed the ARIMA model only in a stable market. When dealing with volatile markets, the Neural Network system only showed 23% accuracy and the ARIMA 42%. Lakshminarayanan (Sai et al. 2019) has made a comparative study of SVM, naïve Bayes, and random forest and found that each model's performance is improved when technical parameters are represented as deterministic data. Alkhatib et al. (2013) compared KNN against non-linear regression for a sample of six major companies listed on the Jordanian stock exchange for predicting next-day price. They have used various evaluation matrices, including the total sum of squared error, average error, cumulative closing price when sorted using predicted values, k values, and training root mean squared (RMS) errors. The author used second-order polynomial (quadratic) non-linear regression and found that the k-nearest neighbor algorithm performs better than non-linear regression. Phayung Meesad (Meesad and Rasel 2013) used SVR with different kinds of windowing operators. They showed that SVR with a rectangular window and flatten window is reasonable to predict for 1-day and

5-day ahead stock prices of DSE. Md. Anwar and Rahman (2019) applied technical analysis in several machine learning models. Their paper compares the performance of the RNN-LSTM hybrid model, DNN, random forest, logistic regression, and SVM. They found the highest accuracy in deep neural network (DNN), which was 94.29%. But, we compare the performance of the KNN regression and linear support vector regression (SVR). Many studies found that SVR is better but did not find whether it is linear or nonlinear and also did not compare with KNN. We found that linear SVR is better than KNN.

Data and methodology

Research data

This study considered the Dhaka Stock Exchange's daily closing price index conveniently from three companies with trading code BXPHARMA, SQPHARMA, and GP from the DSE website. We have collected the most recent dataset (01-January-2016 to 31-January-2019). The training data points cover the period from 01-January-2016 up to the end of 31-December-2017, while the data points starting from 01-January-2018 up to 31-January-2019 are used as the test data (Fig. 1).

Project architecture

Our study involves a list of processes required to be performed in a stepwise manner, depicted in Fig. 2. Based on this flowchart, project architecture is built. We

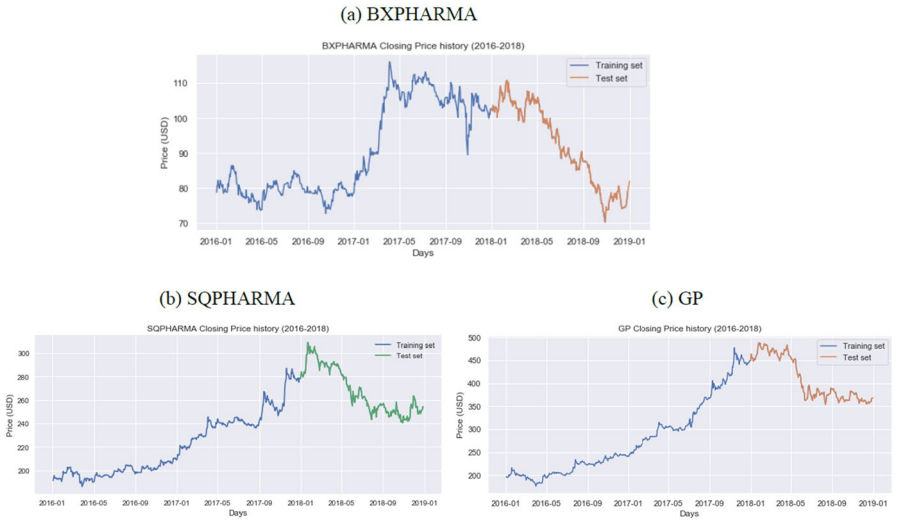

Fig. 1 Train–test split of selected shares

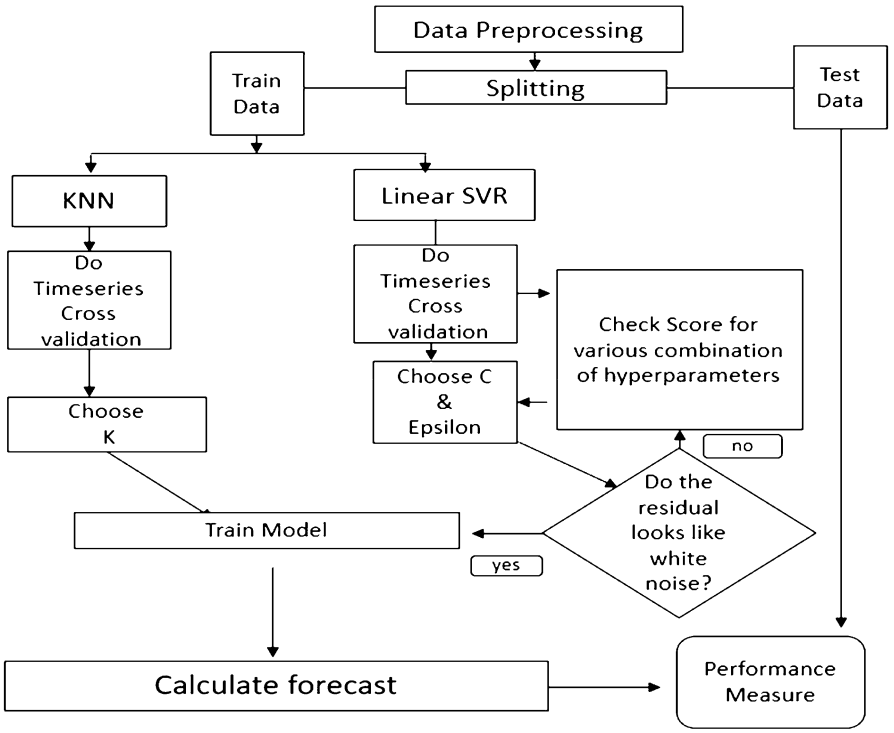

Fig. 2 Data analysis architecture

performed the train–test split of the dataset while keeping the dataset's time order (i.e., without shuffling the dataset). To find the best model, we run a loop to compare a different combination of hyper-parameters, i.e., the values of $\in$ and c in the case of Linear SVR and the values of k in the case of KNN with their evaluation matrices. After training our model with optimal hyper-parameters, we apply this to our testing set to evaluate our model performance.

All the programs used to generate the results in this study are written in Python version 3.7.0. in Anaconda's Jupiter notebook editor. We used different built-in library files such as Scikit-Learn, NumPy, Pandas, Matplotlib, etc.

K-nearest neighbor (KNN) regression

A simple process of KNN regression is to calculate the average of the number target of K neighborhoods. Another approach uses the reverse distance weighting average of neighboring neighborhoods K (http://www.saedsayad.com, n.d.). KNN is a simple algorithm that stores all available events and predicts a numerical target based

on the degree of similarity (e.g., distance functions). We performed the process described below:

- *Choosing features and targets* Transform the data set into supervised learning that defines features (X) and output variables called targets (y). This mapping function aims to maximize the mapping so well that when having new input data (X), it is possible to predict the output variables (y) for that data. We define the current closing price as feature X, and the next day's closing price as a strategy.
- *Tuning the values of k* k is the most crucial task in KNN and a prediction point. We use time-series cross-validation and iteratively check the root mean square error to determine an appropriate value of k. Table 1 shows the root mean square errors for different values of k. KNN identifies the training observation N_0 closest to the prediction point x_0. In addition, KNN estimates $(f(x)_0)$ using the average of all the responses in N_0,.i.e.

$$f(x_0) = \frac{1}{k} \sum_{x_i \in N_0} y_i.$$

- *Choosing an appropriate k* In the case of k-nearest neighbors, we can derive an explicit analytical expression for the total error as a summation of bias and variance

$$\begin{aligned} \mathrm{MSE}(f(x_0)) &= (f(x) - \frac{1}{k} \sum_{x_i \in N_0} f(x_i))^2 + \frac{\sigma^2}{k} + \sigma^2 \\ &= \mathrm{Bias}^2 + \mathrm{Variance} + \mathrm{irreduciable}_{\mathrm{error}}. \end{aligned}$$

From the above equation, it is clear that, as we increase the value of k, the variance will continue to decrease, but bias is likely to increase if the function $f(x)$ is suitably smooth. For any given training set, the best choice of k would be the optimal trade-off between bias and variance.

Linear support vector regression

In linear support vector regression (Cortes and Vapnik 1995), for a given set of training data $(x_1, y_1), (x_2, y_2) \ldots (x_n, y_n)$ where $x \in R^m$ and $y \in R$, our goal is to find a function $f(x)$ that has at most ε deviation from the obtained targets y_i for all the training data, and at the same time is as flat as possible. Support Vector Regression (SVR) is formulated as an optimization problem using a convex ε-insensitive loss function to be

Table 1 Effect of C and ε in model fitting

	Large C	Small C	Large ε	Small ε
Variance	High	Low	Low	High
Bias	Low	High	High	Low

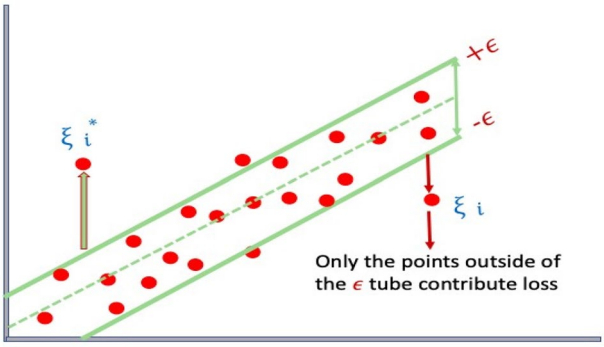

Fig. 3 One-dimensional linear ε insensitive SVR

minimized and finding the flattest tube containing most of the training data. The insensitive zone is referred to as ε tube.

Any training points outside of this ε tube contribute to the actual loss. Figure 3 shows the structure of the one-dimensional linear ε-insensitive support vector regression.

For time-series data, an N-instance sample series is described as

$$(X, Y) = \left(x_t, y_t\right) | x_t \in R^m, y_t \in R, t = 1, \ldots, N.$$

In the financial time series, it can be assumed that all information can be suppressed at a price. Therefore, y_t usually refers to the price at t, and x_t refers to the prices of the days preceding p. To analyze this series, one can evaluate a function, f

$$y_t = f\left(x_t\right) + \varepsilon_t.$$

From the given N-instance sample series, where ε_t is the noise at time t.

Typically, the objective of SVR is to estimates a linear function.

$$f(x) = w_1\varphi_1(x) + \cdots + w_m\varphi_m(x) + b = W^T\varphi(x) + b,$$

in a feature space, R^m, by minimizing the following regression risk:

$$R_{\text{reg}}(f) = \frac{1}{2}w^{\mathrm{T}}w + C\sum_{i=1}^{N} l(f(x_i) - y_i).$$

The superscript T denotes the transpose, φ is a mapping function in the feature space, and b is biased in R. The term $\frac{1}{2}w^{\mathrm{T}}w$ is a complexity term determining the flatness of the function in R^m, C is a regularized constant, and l is a cost function. Generally, the ε-insensitive loss function is used as the cost function. This function does not consider data points in the range of ε-margin, i.e., $\pm\varepsilon$. Therefore, it can reduce the effect of those data points lying in the ε-margin to the approximation

function and control the solution's sparsity. C determines the trade-off between the model complexity (flatness) and the degree to which deviations larger than ε are tolerated in optimization formulation. It is the penalty factor for the error term. If it is too large, we may have over-fit, and if it is too small, we may have under-fit.

ε controls the width of the ε insensitive zone used to fit the training data, i.e., how much error we are willing to allow per training data instance (Theodoridis 2020). So, the range is 0 to max allowed error. The ε can affect the number of support vectors used to construct the regression function. If we allow an immense value of ε, we will end up with fewer support vectors; on the other hand, we may have under-fit and allow smaller results. If we allow a smaller value of ε, we will have a larger number of support vectors. If ε is larger than the target values range, we cannot expect a good result. If ε is zero, we can expect over-fitting. Hence, both C and ε values affect model complexity but in a different way. Table 1 describes the effect of the hyper-parameters in model fitting.

SVR models are highly sensitive to tuning hyper-parameters. We can use a grid-search cross-validation iterative process to tune the hyper-parameters.

Empirical results and analysis

This study follows mainly K-nearest neighbor (KNN) and support vector regression (SVR) in which algorithms are presented with historical stock data. The algorithms use historical data to train a model that is expected to infer future prices given recent price information. The models are trained on daily stock exchange data to make short-term predictions for one day ahead which is also done by previous studies. This study intends to predict three selected companies of the Dhaka Stock Exchange's future values based on their historical prices. Two machine learning algorithms will be used to solve the problem of predicting stock prices. All the algorithm parameters are optimized using cross-validation with the grid-search algorithm and iterative process. The predictions are then based on the parameter and their forecasting/prediction errors are compared to find out the best. Table 1 shows that different RMSE with K and suggests that Ks values will be 15 for

Table 2 K vs. RMSE

Trading code	k	RMSE	Trading code	k	RMSE	Trading code	k	RMSE
BXPHARMA	1	2.199	SQPHARMA	1	8.12	GP	1	11.05
	2	1.716		2	7.26		2	7.90
	5	1.49		4	7.34		5	8.07
	10	1.42		6	7.09		10	9.17
	15	1.38		10	7.52		15	9.81
	20	1.39		15	8.09		20	9.87
	25	1.43		20	8.49		25	10.89

BXPHARMA, 6 for SQPHARMA, and 2 for GP for their minimum root mean square error (RMSE) (Table 2).

The study also runs a loop to find out the best value of the linear SVR hyper-parameters c and ε. At first, the set default value of C and then iterate various values of ε. At the second stage, set the value of ε as the optimum value found first and then iterate various C and finally fit models considering the different combinations of C and ε simultaneously. Also, draw a residual plot to check the distribution of random error. Figure 3 shows both train R^2 and test R^2 simultaneously for each of the companies that indicate a few values of the hyper-parameters C and ε for BXPHARMA 1–10 and 0–2, for SQPHARMA 5–15 and 0–2, and GP10-15 and 0–4, respectively. This study also inspects the values for each of the iterations by considering each hyper-parameter simultaneously with the help of a cross-validation grid-search algorithm.

Summarized results from the cross-validation iterations, shown in Fig. 4

Residual plots of a series of errors should like random. If there are patterns in the errors, we can use one error to predict another. When the residuals are randomly scattered around zero for the entire range of fitted values, i.e., approximate center on zero, they indicate that the model's predictions are correct on average rather than systematically too high or low. The residuals plots in Fig. 5 show that the residuals are not systematically high or low, and the residuals are approximately centered on zero throughout the range of fitted values. In other words, the model is correct on average for all fitted values (Fig. 6).

Combining the above results, we get the optimum value of the hyper-parameters for the desired model; results are described in Tables 3, 4, and 5.

There exists one that stands out; the SVR model has the lowest value for RMSE in all the three stocks. The higher value of RMSE is for KNN than Linear SVR when we are trying to forecast GP and SQPHARMA. To better understand the dimension of the two algorithms' errors, the daily predictions made by each of the mares are compared against their actual values of the training data sets. After the training with KNN and SVR, the predicted price and the actual price for the test data are exhibited in the following figures.

Figure 7 illustrates the actual data and predicted data. The blue line is the actual value, and the red line is the predicted value of our method. From the above figures, it can be observed that in all the cases, the tendencies of the predicted value curve are identical to that of the actual value curve. The predicted curve fits the actual curve in most of the period. Therefore, SVR predicts the actual data very well than KNN.

It is evident from Fig. 8 that the SVR model has the lowest value for RMSE in all the three stocks compare to KNN. So, SVR performed better.

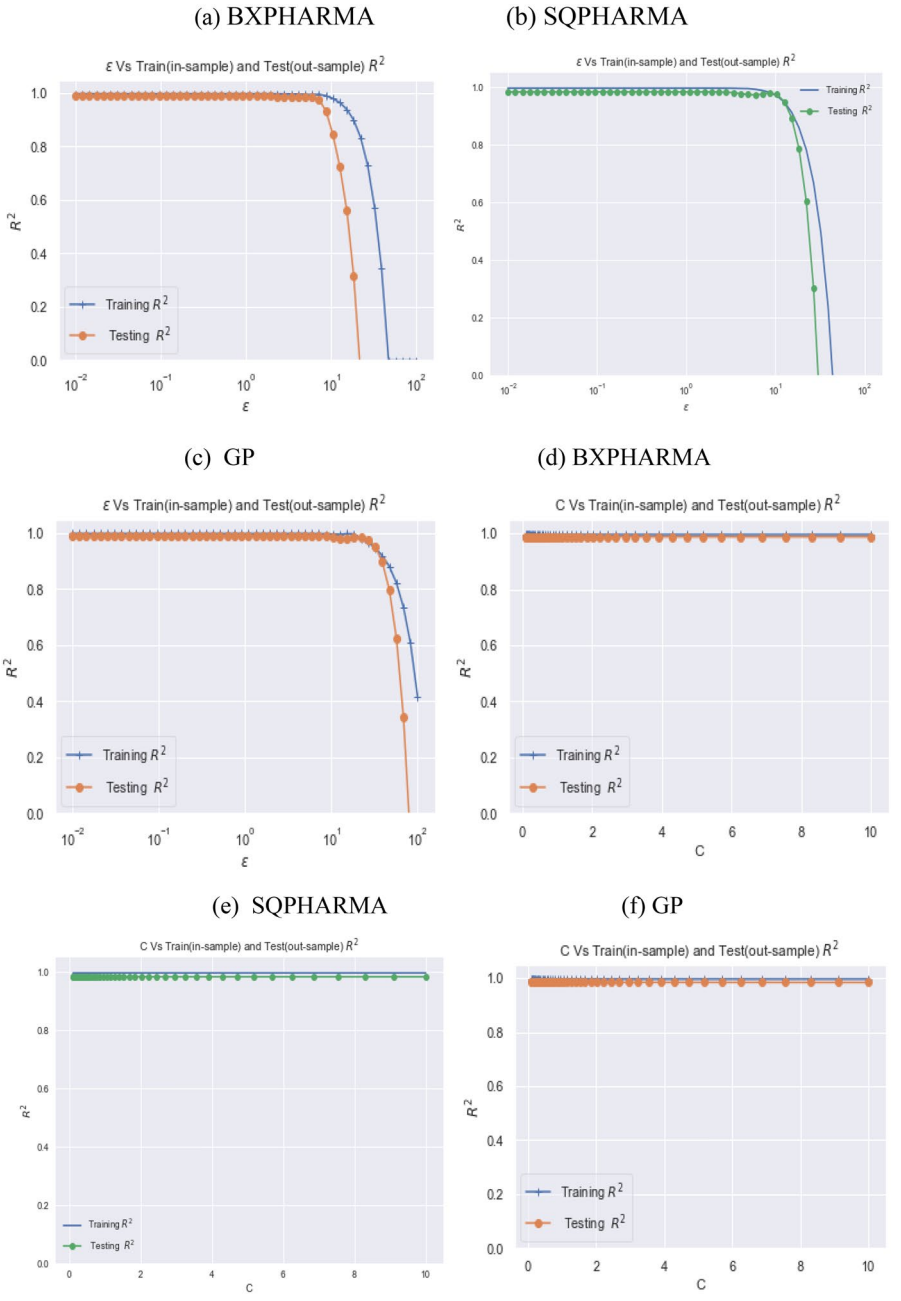

Fig. 4 Score of C and ε with different combinations

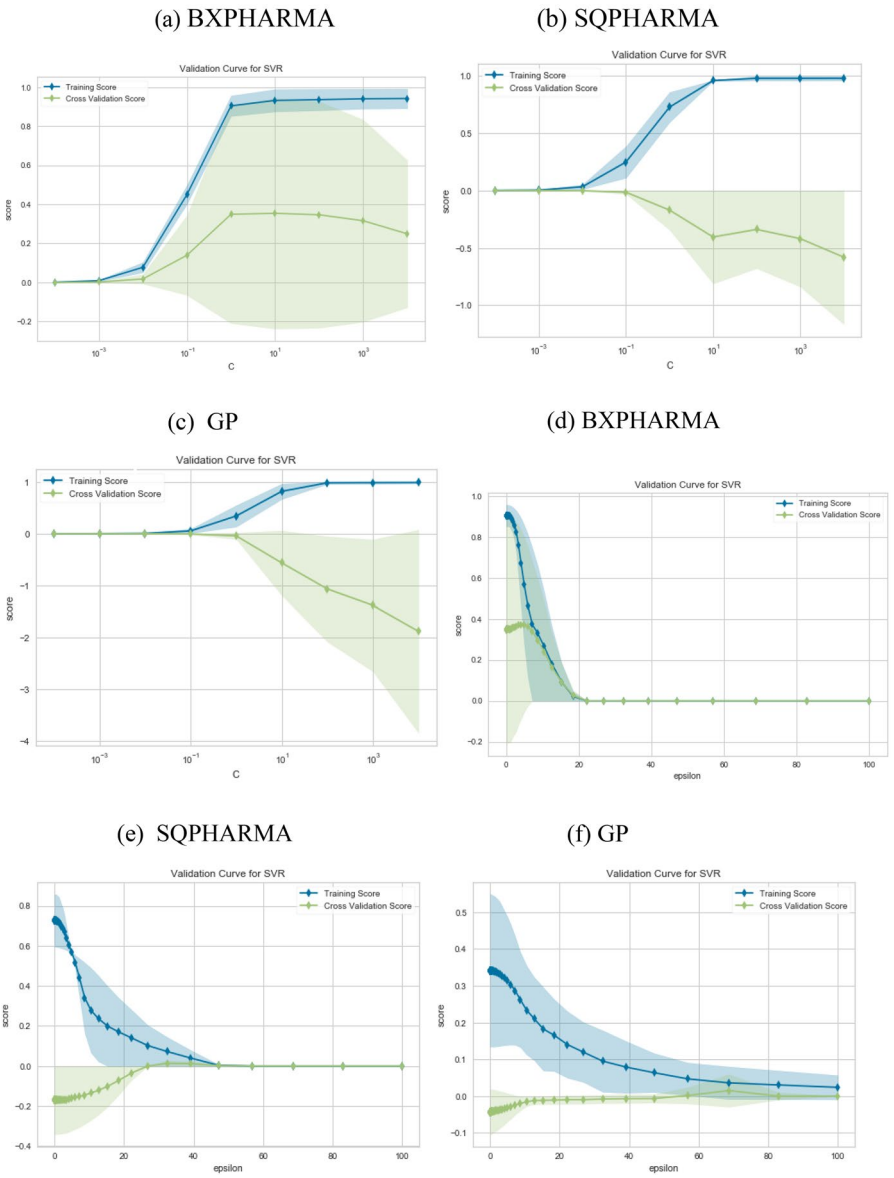

Fig. 5 Validation curves of C and ε

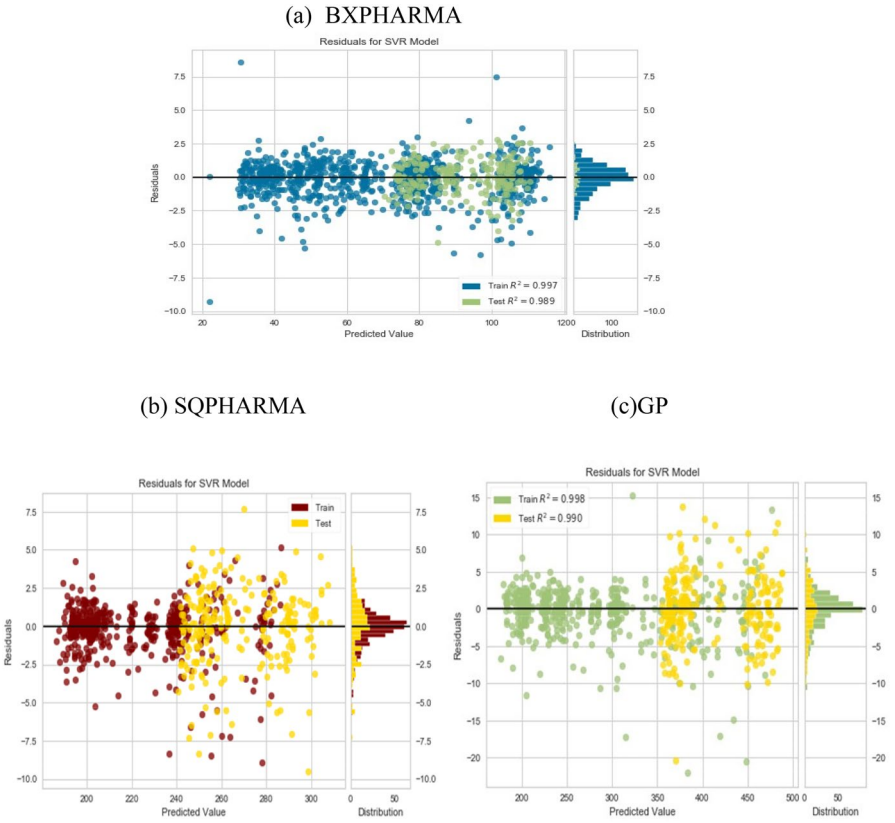

Fig. 6 Residual plots

Table 3 Sample values of hyper-parameters for linear SVR

Trading code	R^2	C	ε	Adjusted R^2	RMSE
BXPHARMA	10	0.0001	0.9639	0.9636	1.12
SQPHARMA	1	0.01	0.88	0.878	2.64
GP	10	0.001	0.825	0.824	2.98

Table 4 Performance Mmetrics of KNN

Trading code	R^2	Adjusted R^2	RMSE
BXPHARMA	0.9639	0.9636	1.38
SQPHARMA	0.82	0.815	6.582
GP	0.845	0.824	5.128

Table 5 Performance metrics of SVR

Trading code	R^2	Adjusted R^2	RMSE
BXPHARMA	0.9739	0.9636	1.239
SQPHARMA	0.88	0.878	2.04
GP	0.85	0.834	2.98

Fig. 7 Actual data and predicted data of the selected companies by KNN and SVR

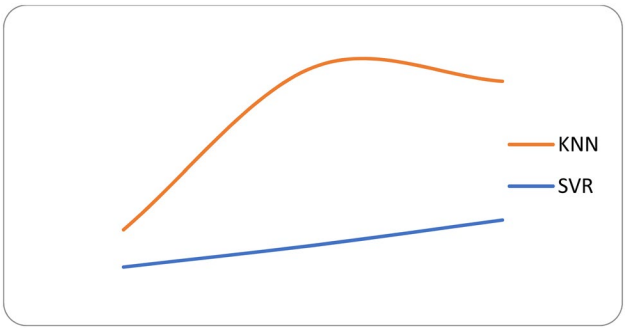

Fig. 8 Performance comparison of SVR and KNN in terms of RMSE

Conclusions

In this study, many machine learning algorithms have been used to predict the prices of DSE. Among them, the two most popular algorithms are KNN and SVR. We have made a comparative study between them. This study has used a regression approach instead of a classification approach to predict exact prices. K-nearest neighbor regression algorithm and linear epsilon intensive support vector regression algorithm have been used to predict the daily closing prices of selected shares of DSE. Cross-validation, along with the iteration process, has been done to determine the optimum hyper-parameters. We have made predictions after choosing the best model and applied it to test data. Our study shows that linear SVR has a smaller error than KNN and linear SVR has a higher and adjusted R^2 value in both test and train sets. So, linear SVR performed better, and it can be used to predict 1 day ahead of the closing price of the stock market, providing previous historical data. To sum up, as the stock market is a vital financial sector, a comparison between the time series models can help determine whether to buy a stock or sell it, and this crucial purpose can be served with the help of this comparative study of stock market predictions. In previous studies, we found that SVR is better but did not find whether it is linear or nonlinear nor did we compare with KNN. We found that linear SVR is better than KNN. This study has been done only on three selected companies and has further investigated the performance of SVR, and KNN should be tested for other time-series data. The study also suggests that the SVR algorithm is a more precise way of predicting the Dhaka Stock Exchange (DSE) than the KNN regression.

Author contributions SI (M.Sc.): conceptualization and writing original draft preparation; MFH (MPhil.): writing—reviewing and editing; MSS: data curation and methodology; PC*:* software visualization and investigation.

Funding This research received no specific grant from any funding agencies in the public, commercial, or not-for-profit sectors.

Data availability statement The data that support the findings of this study are openly available on the Dhaka Stock Exchange (DSE) website http://www.dsebd.org.

Declarations

Conflict of interest The authors have no affiliations with or involvement in any organization or entity with any financial interest (such as honoraria; educational grants; participation in speakers' bureaus; membership, employment, consultancies, stock ownership, or other equity interest; and expert testimony or patent-licensing arrangements), or non-financial interest (such as personal or professional relationships, affiliations, knowledge or beliefs) in the subject matter or materials discussed in this manuscript. All authors have no competing interests.

Ethics approval The authors certify that this research is totally based on the data that were collected from the website and it is freely available on the Internet. Therefore, ethical approval was not obtained for this research. Authors are responsible themselves.

References

Abecasis SM, Lapenta ES, Pedreira CE (1999) Performance metrics for financial time series forecasting. J Comput Intell Finance 7(4):5–23

Alkhatib K, Najadat H, Hmeidi I, Ali MK (2013). Stock price prediction using K-Nearest Neighbor (kNN) Algorithm. https://www.semanticscholar.org/paper/Stock-Price-Prediction-Using-K-Nearest-Neighbor-Alkhatib-Najadat/1507329f5382a1550430657ebae7f70507f63410. Accessed 11 Dec 2019

Anwar M, Rahman S (2019) Forecasting stock market prices using advanced tools of machine learning. Doctoral dissertation, Brac University

Cao LJ, Tay FEH (2003) Support vector machine with adaptive parameters in financial time series forecasting. IEEE Trans Neural Netw 14(6):1506–1518

Chen KY, Ho CH (2005) An improved support vector regression modeling for Taiwan Stock Exchange market weighted index forecasting. In: 2005 International conference on neural networks and brain, vol 3. IEEE, pp 15–1638

Chen WH, Shih JY, Wu S (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

Chen CC, Chen CH, Liu TY (2020) Investment performance of machine learning: analysis of S&P 500 index. Int J Econ Financ Issues 10(1):59–66

Cortes C, Vapnik V (1995) Support-vector networks. Mach Learn. https://doi.org/10.1007/BF00994018

Mbeledogu NN (2012) Stock feature extraction using principal component analysis. In: 2012 International conference on computer technology and science (ICCTS 2012), [online] 47. https://www.researchgate.net/publication/318958826_Stock_Feature_Extraction_Using_Principal_Component_Analysis. Accessed 5 Oct 2019

Meesad P, Rasel RI (2013) Predicting stock market price using support vector regression. In: 2013 International conference on informatics, electronics and vision (ICIEV). IEEE, pp 1–6

Sai K, Lakshminarayanan J, Mccrae (2019) A Comparative study of SVM and LSTM deep learning algorithms for stock market prediction. http://ceur-ws.org/Vol-2563/aics_41.pdf. Accessed 5 Dec 2019

Theodoridis S (2020) SVM Learning. [online] svmlearning.com—2020 product ranking algorithm powered by AI. https://svmlearning.com/parameters/. Accessed 9 Oct 2020

Tsai CF, Wang SP (2009) Stock price forecasting by hybrid machine learning techniques. In: Proceedings of the international multiconference of engineers and computer scientists, vol 1(755), p 60

Vainionpää I, Davidsson S (2014) http://www.diva-portal.org/smash/record.jsf?pid=diva2%3A771141. [online] DIVA. http://www.divaportal.org/smash/record.jsf?pid=diva2%3A771141&dswid=-412. Accessed 14 Dec 2020

Wang Y (2014) Stock price direction prediction by directly using prices data: an empirical study on the KOSPI and HSI. Int J Bus Intell Data Min 9(2):145. https://arxiv.org/pdf/1309.7119. Accessed 20 Apr 2019

www.saedsayad.com. (n.d.). KNN Regression. https://www.saedsayad.com/k_nearest_neighbors_reg.htm

Authors and Affiliations

Sharmin Islam[1] · Md. Shakil Sikder[1] · Md. Farhad Hossain[2] · Partha Chakraborty[3]

Md. Shakil Sikder
shakil.stat.bsmrstu@gmail.com

Md. Farhad Hossain
farhad390ju@gmail.com

Partha Chakraborty
partha.chak@cou.ac.bd

[1] Department of Statistics, Bangabandhu Sheikh Mujibur Rahman Science and Technology University, Gopalganj 8100, Bangladesh

[2] Department of Statistics, Comilla University, Cumilla, Bangladesh

[3] Department of Computer Science and Engineering, Comilla University, Cumilla, Bangladesh