

STOCK PREDICTION AND SIMULATION OF TRADE USING SUPPORT VECTOR REGRESSION

Jay Borade¹, Ayush Khanvilkar², Rahul Kanvinde³, Akshay Narkhede⁴, Anuj Odedra⁵

¹Assistant Professor Information Technology at Fr CRCE, Mumbai University, Maharashtra, India

²Final year Information Technology student at Fr CRCE, Mumbai University, Maharashtra, India

³Final year Information Technology student at Fr CRCE, Mumbai University, Maharashtra, India

⁴Final year Information Technology student at Fr CRCE, Mumbai University, Maharashtra, India

⁵Final year Information Technology student at Fr CRCE, Mumbai University, Maharashtra, India

Abstract

Up's and Down's in share market are always unpredictable. Commercial banks offer their customers market predictions based on the sentiment and market news of the given day. These predictions are relevant for a short period of time. Commercial banks cater to a large demography hence they have to limit their prediction services. Investment banks have used predictive models which use past market data to predict stock prices and market indexes. Common people cannot afford the services provided by the investment bank. Candlestick is widely used in the trading community for analysis. But candlestick chart looks different for various time frames and they make it difficult to manage risks. The proposed system aims at helping traders make sound financial decisions. It simulates trading thereby helping new users understand the application. It will be of assistance to beginners so that they can learn how to trade without losing any capital. The system uses machine learning techniques and also lets the user view sentiment about the stock in real time. Both mathematical predictions and sentiments are used as parameters for making a financial decision. The proposed system is able to achieve prediction accuracy of up to 95%[1].

Keywords: Machine Learning, scikit-learn, share market, stock prediction, support vector regression

1. INTRODUCTION

From the past decade machine learning techniques are applied in wide range of domains like Communication, Finance and Electronics. Algorithms used for developing Recommendation systems, Predictive models and Natural Language Processing tools are implemented using machine learning. Stock Prediction using machine learning approaches makes it possible to process data at a large scale with greater speed. Using these techniques, models can be built that are trained on past market data to predict future stock prices. Therefore Predictive models can predict stock prices for longer period of time. Data consisting of stock prices of more than 10 years can train the model to predict stock prices for upcoming 30 to 35 days. The performance of the stock over a small period of time can mainly be determined by the sentiment or the public opinion about that stock in the trading community. Both the sentiment and the mathematical prediction can be used to estimate the performance of the stock over short term and long term respectively[2].

1.1 Environmental Setup

1) Data Source: In order to undertake the experiments and evaluate the results from the experiments, NASDAQ index was selected as our research domain. The Apple stock (AAPL) was selected to analyze the results and make

predictions. The historical data from the year 2004 till date was selected as the dataset. 80% of data was used for training and the remaining for testing. The original dataset contains 6 attributes: Date, Open, High, Close, Adj Close and Volume. The goal was to predict the Closing price of Apple stock (AAPL) for next 35 days[3].

```
Date,Open,High,Low,Close,Adj Close,Volume
18-08-04,2.179286,2.275,2.177857,2.267143,1.534757,91163800
19-08-04,2.250714,2.275714,2.168571,2.193571,1.484952,97230000
20-08-04,2.193571,2.213571,2.177857,2.2,1.489305,79195200
23-08-04,2.204286,2.233572,2.185714,2.22,1.502844,63665000
```

Fig 1: The above figure shows the format of historical data selected for Apple stock(AAPL).

2) Technology Stack: The python library scikit-learn is used for model selection, matplotlib for data visualization, Django web framework is used for developing the web application, NumPya python library is used to handle large collection of dataset since data from the year 2004 till present is used and another python library beautiful soup is used for scrapping financial news website and displaying relevant news related to stock market.

2. SYSTEM OVERVIEW

2.1 Client Side Perspective

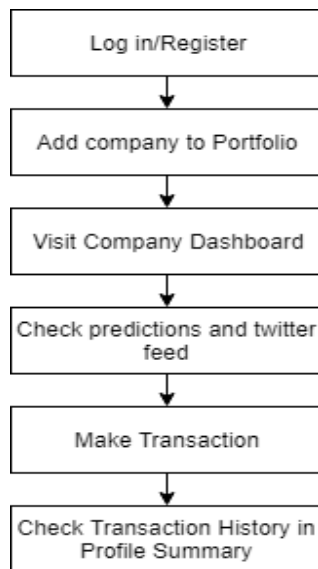


Fig 2: A flowchart of Client Side Perspective

The user has to register on the web app or log in if he/she a registered user.

2.1.2 Add Company to Portfolio

Once the user has registered he/she can add companies to his portfolio. When the user clicks on the add button, the company is assigned to the user’s profile. The user can now visit the dashboard of the company.

2.1.1 Log in/Register

Company Name	Date	Close	Open	High	Low
↑ Microsoft Corp.	2018-04-13	\$93.0800	\$94.0500	\$94.1800	\$92.4400
↑ International Business Machines Corp.	2018-04-13	\$156.7100	\$158.6700	\$159.2200	\$155.9100
↓ McDonald's Corp.	2018-04-13	\$161.7300	\$162.1600	\$163.1600	\$161.2600
↑ Hewlett-Packard Co.	2018-04-13	\$21.7700	\$22.0400	\$22.1200	\$21.6450
↓ Walt Disney Co.	2018-04-13	\$100.3500	\$101.0000	\$101.5200	\$100.1600
↑ Johnson & Johnsons	2018-04-13	\$130.6200	\$131.0900	\$131.4200	\$129.6300
↓ AT&T Inc.	2018-04-13	\$35.1400	\$35.2100	\$35.3700	\$35.0200
↓ JPMorgan Chase & Co.	2018-04-13	\$110.3000	\$115.0200	\$115.1500	\$109.0500
↑ Wal-Mart Stores Inc.	2018-04-13	\$86.0200	\$85.8900	\$86.2700	\$85.1901
↑ General Motors Corp.	2018-04-13	\$38.7300	\$39.1200	\$39.3800	\$38.5650

*Close Prices, Open Prices, High Prices, Low Prices are in US dollars(\$).

Fig 3: List of companies the user can add to his portfolio.

2.1.3 Visit Company Dashboard

Once the company has been added to the user’s profile the user can visit the dashboard of the company from the profile summary page. The dashboard of the company displays the prediction graph of coming weeks, live price of that stock, predicted closing price for the next day and twitter feed of that company. The dashboard has a form so that the user can

buy stocks of that company by entering the amount of stocks user wants to buy at the live price displayed.

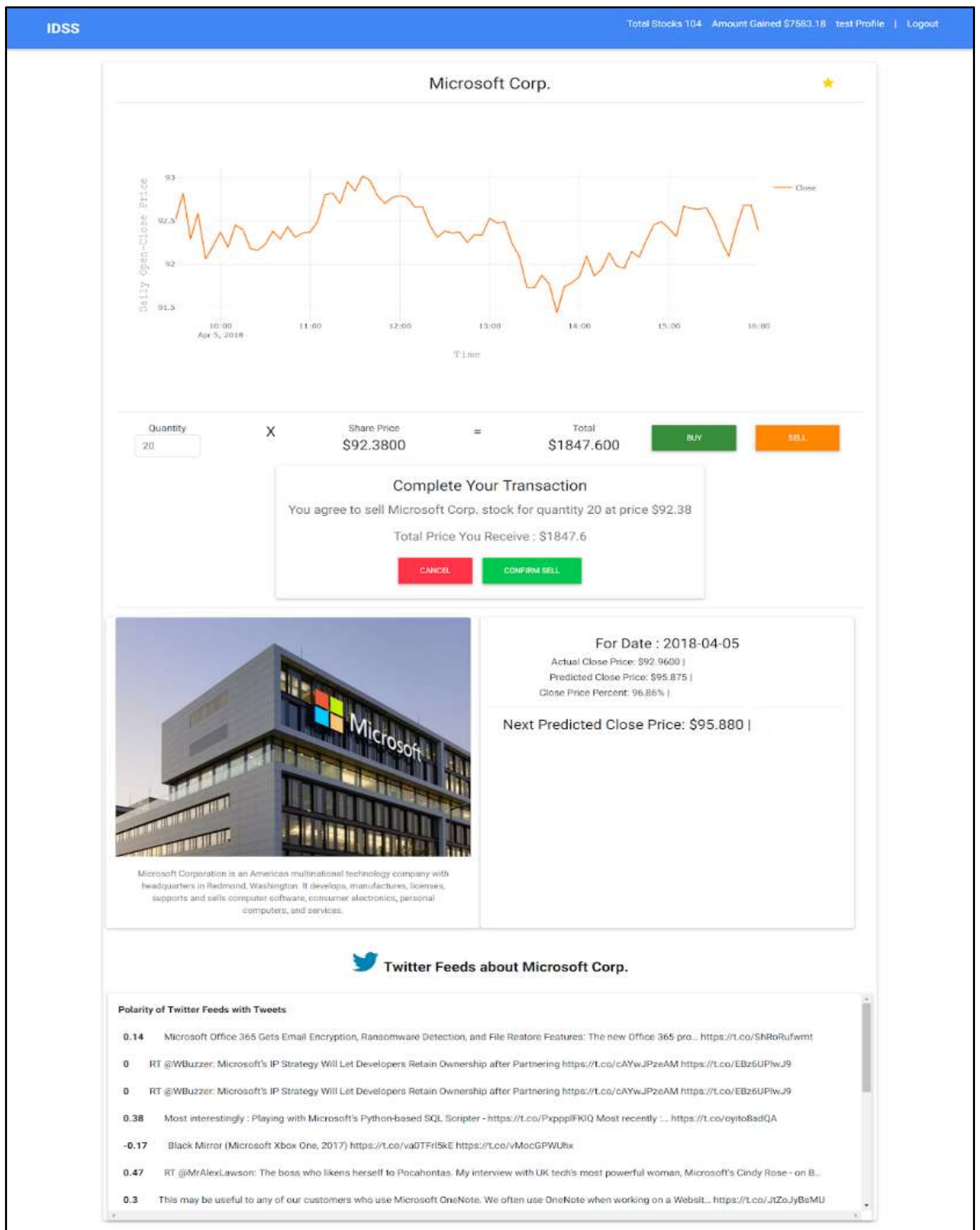


Fig 4: The dashboard displaying the prediction graph, user transaction and twitter feeds for specific company

2.1.4 Check Predictions and Twitter Feed

The user can use the graph of the prediction and twitter feed as indicators for making a transaction. If the indicator are positive the user can buy more stocks of the given company if not then he can sell the stocks.

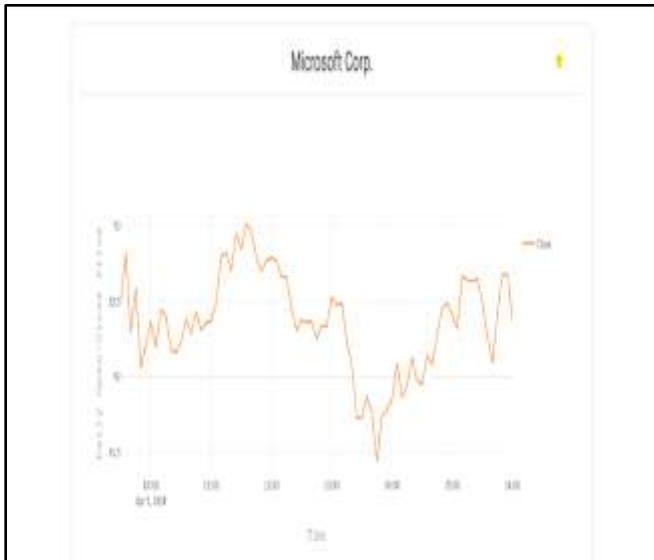


Fig 5: The predicted graph for the company Microsoft Corp.



Fig 6: The twitter feeds for the company Microsoft Corp.

2.1.5 Make Transaction

The transaction can be at the price displayed on the dashboard once the user enters the amount of stocks he/she would like to buy/sell. The user gets a confirmation after the transaction is processed. The data about the transaction is reflected in the transaction history.



Fig 7: The dashboard displaying ongoing user's transaction.

2.1.6 Check Transaction History in Profile Summary

The profile summary reflects statistics considering the recently processed transaction. The user can further view Transaction History by clicking on the check history button.

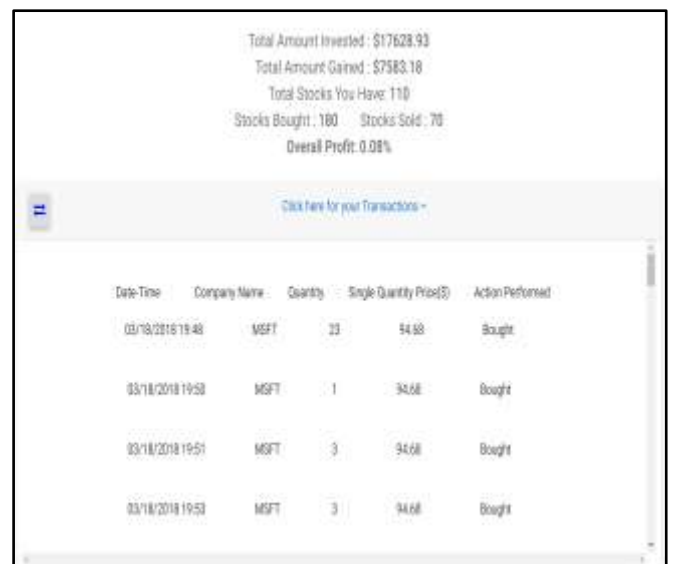


Fig 8: The transaction history of the user currently logged in.

2.2 Server Side Perspective

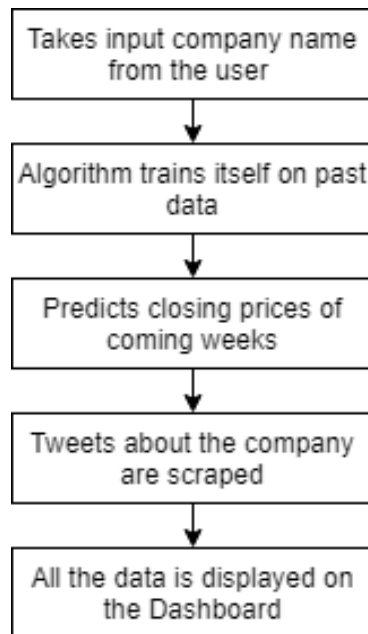


Fig 9: A flowchart of Server Side Perspective.

2.2.1 Take Input Company Name from User

The system takes input from the user as to which company the registered user is interested in. The user can visit the dashboard of the company after adding it to its profile.

2.2.2 Algorithm Trains itself on Past Data

The system trains itself on the past data of the company identifies patterns to predict prices for coming weeks. The system runs Support Vector Regression on the company's data. The past data constitutes of opening price, closing price, highest price, lowest price of the day for the last decade.

2.2.3 Predicts Closing Prices of Coming Weeks

Based on the analysis of the past data of the company the algorithm predicts closing prices of the stock for coming weeks. The array of the closing prices is visualized on a graph. The graph helps the user understand the trend stock will follow.

2.2.4 Tweets about the Company

The twitter feed is significant to estimate the sentiment about the stock in the trading community. To analyse the sentiment the system queries the name of the stock on twitter using Tweepy library in python. Tweepy returns bunch of tweets in real time for the user to see. The sentiment behind the tweet can be analysed using TextBlob. The library returns polarity of the tweet. If the polarity of is more than 0 then the sentiment is positive else the the sentiment is negative. The polarity is shown along with the tweet.

2.2.5 Data Displayed on Dashboard

All the data is shown on the dashboard. The dashboard is designed based on the guidelines of Material Design. The web pages are developed in MDBootstrap. The UI is simple and easy on the eye. The dashboard has prediction graph at the top followed by dialog box to make a transaction and information of the company like live price and predicted closing price for the next day. At the end the twitter feed of the company is displayed.

3. ALGORITHM

Our system uses support vector regression technique to predict future stock price for specific company. The goal is to use SVR model to predict the stock price of next 30 days considering historical data from the year 2004 till date. This dataset consists of Date, Open, High, Low, Close and Volume of each stock. It will help us predict the stock price movement for the next 30 days. We conducted a survey on stock prediction using various machine learning models and found that the best results were obtained with Support Vector Regression. Our proposed System is able to achieve prediction accuracy of upto 95%.

3.1 Support Vector Machine

Support Vector Machines are the best binary classifiers. It creates a hyperplane which separates the data points on both sides of the decision boundary to correctly classify the dataset. The equation of the hyper plane is given as:

$$\beta_0 + \beta_1 X_1 + \dots + \beta_n X_n = \beta_0 + \sum_{i=1}^n \beta_i X_i = 0$$

The data points are known as the support vectors and the hyperplane is called as the Support Vector Classifier (SVC). We have used the python's scikit-learn implementation of the SVM. SVM has two major applications, classification and regression. The main characteristics of Support Vector Regression (SVR) is that it tries to minimize the generalized error bound so as to achieve generalized performance[4]. It consists of 3 kernel functions which are given below:

Linear Kernel: $K(X, Y) = X^T Y$
 Polynomial Kernel: $K(X, Y) = (\gamma \cdot X^T Y + r)^d, \gamma > 0$
 Radial Basis Function (RBF): $K(X, Y) = \exp(-\|X - Y\|^2 / 2\sigma^2)$
 Here r, γ, σ are kernel parameters.

Mathematically it can be described as follows:

The training set is given as $T \{(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)\}$, Where R . Let us consider that the training data set is based on a distribution of the selected $x_i \in R^n, y_i \in R$ independent and identically distributed sample points. The final real function $\tilde{f}(x)$, so as $y = \tilde{f}(x)$ to infer the value of a corresponding input $x_i \in R^n$, and make the expected risk on the training set $y \in R$. And on the training set makes the expected risk

$$R(f) = \int c(x, y, \bar{f}) dP(x, y)$$

Steps:

1. The training dataset is given as $T = \{(x, y), i1, 2, , l\}$
2. Select the appropriate kernel function and set the positive parameters of e and C .
3. Construct the optimization problem and find the optimal solution which is given as

$$\bar{a}^{(*)} = (\bar{a}_1, \bar{a}_1^*, \bar{h}, \bar{a}_1, \bar{a}_1^*)^T$$

4. Construct the function regression estimates:

$$\bar{f}(x) = \sum_{i=1}^l (\bar{a}_i^* - \bar{a}_i) K(x, x_i) + \bar{b}$$

Which \bar{b} calculated by the following formula in the range of $(0, C)$ choice \bar{a}_j or \bar{a}_k^* , if elected to that \bar{a}_j , the

$$\bar{b} = y_j - \sum_{i=1}^l (\bar{a}_i^* - \bar{a}_i) K(x, x_i) + \varepsilon$$

If the election to that \bar{a}^* , the [5]

$$\bar{b} = y_j - \sum_{i=1}^l (\bar{a}_i^* - \bar{a}_i) (x, x_i) -$$

3.2 Prediction Graph

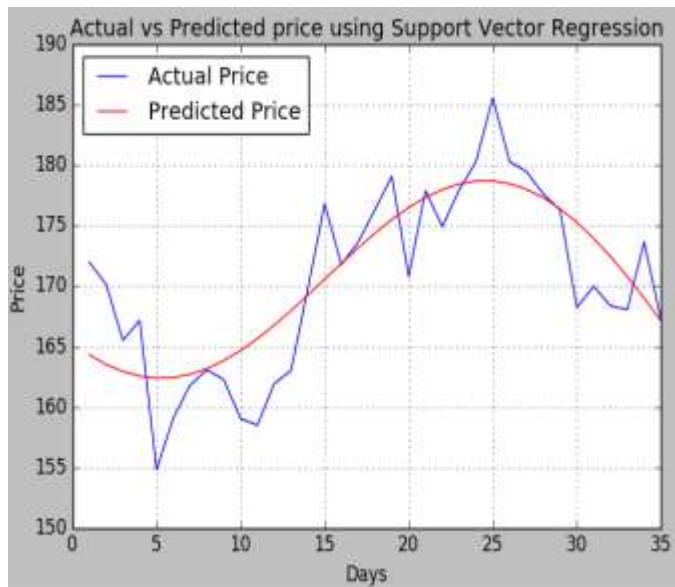


Fig 10: The graph above shows the comparison between the actual value and predicted value for AAPL stock. The above graph is for the time span of 35 days for both actual and predicted price. For the most part, the graph of actual values and predicted values are overlapping. After 20 days the predicted and actual value is 178 and after 35 days the predicted and actual value is 171. In comparison to other algorithms support vector regression produces accurate results.

4. RESULTS & CONCLUSION

Table 1: Actual Versus Predicted Values

Days	Support Vector Regression		
	Actual	Predicted	Accuracy %
5	154.8	162.7	94.6%
10	159.0	164.6	96.4%
15	176.8	170.2	96.2%
20	170.8	177.7	96.1%
25	185.5	178.6	96.28%
30	168.2	175.2	95.83%

The above table gives the actual versus predicted values for the Apple stock (AAPL) for the upcoming 30 days. The results from Table I shows that Support Vector Regression is giving approximately 95 % of accuracy.

REFERENCES

- [1] George Tzani, IoannisKatakis, IoannisPartalas, IoannisVlahavas, "Modern Applications of Machine Learning", *Aristotle University of Thessalonik*
- [2] S. Angra and S. Ahuja, "Machine learning and its applications: A review", *Big Data Analytics and Computational Intelligence (ICBDAC), 2017 International Conference on 23-25 March 2017.*
- [3] Jay Borade, AyushKhanvilkar, Rahul Kanvinde, AkshayNarkhede, AnujOdedra, "Survey on Stock Prediction Using Machine Learning Approaches.", *IJRASET Volume 6, Issue IV, April 2018.*
- [4] Saahil Madge, "Predicting Stock Price Direction using Support Vector Machines", *Princeton University.*
- [5] Chun Cai, Qinghua Ma, and ShuqiangLv, "Research on Support Vector Regression in the Stock Market Forecasting", *Advances in Intelligent and Soft Computing Volume 148.*