NOVEL APPROACH FOR ANALYZING INTRADAY STOCK MARKET BEHAVIOR USING STREAM DATA ANALYTICS

Vishal Shah[1]

[1]Department of Wildlife Resources, Virginia, USA.

ABSTRACT

Stock price prediction is an emerging domain of machine learning applications that largely depends on the continuous monitoring and processing of information. One of the open challenges in this domain is to find the best possible machine learning regression algorithm to accurately predict the closing price of any stock in real time using stream data. This research assesses the possibility of solving this challenge - whether a single ML algorithm can be superior for such a task. Further, an incremental learning approach is proposed to improve the performance of the stock market prediction system. Moreover, a prediction system has been implemented that gives buy/sell suggestions and predicts how much the stock price can vary in the future. The experiments are based on the historical data collected from Google Finance API and real-time data fetched from the yfinance python library. For analyzing data and building prediction models, the scikit-learn library has been used to analyze data and build predictions. This paper covers different algorithms that perform well on different stocks and reflects that adding an incremental learning approach boosts prediction accuracy.

Keywords: *Incremental Learning, Machine Learning, Stock Market, Stock Volatility, Stream Data Analysis, Trading Strategies.*

1. Introduction

Stream Data is continuously arriving data. With a targeted approach, valuable insights can be extracted from the analysis of streaming data, which can be useful in many domains like the stock market, Business growth, Geo-informatics, E-commerce, etc. Stock data is one such example as dynamic information is generated continually. Nowadays, almost all the financial sector, brokerages, and banks analyze stock scores to gain knowledge about the business. As a result of the pandemic, even the general public has started investing in stocks to sustain their incomes. As per a report [1] from CNBC, an American news channel, many people are investing in the stock market by intraday trading (also known as day trading) during the pandemic. Stock market prediction aims to determine the future movement of the stock value of a financial exchange. The stock market, being a financially volatile market, demands a very precise prediction of a future trend to boost profits for investors. Since these trends are not always uniform, advanced machine learning algorithms are required for accurately forecasting non-linear trends. In recent years, machine learning has greatly contributed to the field of stock market analysis and prediction [2]. To help clients assess risk and make a buying or selling decision, stock market analysis/prediction using machine learning techniques serves as a guide.

Intraday trading [1, 21] is the practice used by people to invest in the stock market that involves technical analysis and execution of some special trading strategies. It is different from the usual trading practice as

[1] https://www.cnbc.com/2020/09/21/many-people-turn-to-day trading-in-pandemic-few-will-be-a-winners.html

the stocks are purchased not to invest but, as the name suggests, to leverage profits from stock price changes on the same day. Trading stocks requires cumbersome groundwork of studying, researching, analyzing, and finding patterns [30, 31] to understand stock behavior but the challenging part of performing intraday trading is that it needs real-time technical analysis where accuracy in timing is imperative. Computations need to be performed at least every minute to give effective insights into intraday movements of prices and machine learning models used for predicting prediction, therefore inducing the need for stream data analytics and incremental learning.

Technical contributions made by this paper include (1) obtaining real-time data and preprocessing it; (2) building a regression model with incremental learning [32] capability for predicting closing price in real-time; (3) developing a scanner for continuously finding patterns in stock price to give buy/sell/hold suggestion; (4) providing risk assessment by calculating stock volatility [33]. The remainder of the papers is organized as follows: the 'Literature Survey' section reviews works related to the domain. The 'Proposed work' section covers the description of the proposed flow. The 'Experiments and Result Discussion' section elaborates on the experiments carried out and their respective results. Lastly, the paper ends with the 'Conclusion and Future Work' section.

2. Literature survey

Most of the works involving stock market analysis through machine learning show either of the two types of analysis. The first type of analysis uses regression algorithms on quantifiable data like opening, closing, high and low prices, and volume of shares traded, whereas the other one takes into account the nature of the market as affected by global economics and politics, natural and man-made disasters, market psychology, social media sentiments, etc. Behera et al. [2] have shown both such case studies: (1) Prediction of opening price based on data collected through Financial Websites and (2) Prediction of Stock price based on Twitter data. They attempted to develop scalable, fault-tolerant models for stock prediction from the real-time streaming data using four algorithms, namely Support Vector Machine: Regression (SVR), Decision Tree, Random Forest Regression, and Polynomial Regression [2]. Vijh et al. [4] collected historical data of five companies from Yahoo Finance [3] that contains information about the stock such as High, Low, Open, Close, Adjacent close, and Volume, and developed Artificial Neural Network and Random Forest Regression models to predict next day closing price [4, 29]. Selvin et al. [5] compared the performance of several Deep Learning algorithms in predicting stock prices of NSE-listed companies [5, 22]. Research conducted by Mei et al. [6] applied Random Forest to precisely predict New York electricity prices in real-time [6].

Gurav and Sidnal [19, 26, 28] discuss stock market-related technical indicators, mathematical models, and analysis of several types of machine learning algorithms and the 'when and how' of using them. They state that the majority of stockbrokers that deal with sellers, buyers, and trading in the stock market use technical, fundamental, or time series analysis to predict the stock market and argue that as these methods are based on past stock market trends, rather than actual stock price changes, they may not guarantee accurate stock market predictions. Henrique et al. [20, 27] use Support Vector Regression and measure its performance on various Brazilian, American, and Chinese stocks.

Incremental Learning is a machine learning paradigm that refers to learning from streaming data, which arrives over time, with limited memory resources and, ideally, without sacrificing model accuracy [7, 21]. It is often used in the setting of streaming data machine learning applications to keep models updated, such as the work done by Xu et al. [8, 23]. S Ruping [9] published a paper that serves as an approach for incremental learning with Support Vector Machines [9]. Incremental Learning is offered by many Python libraries, scikit-learn is one of them [10].

3. Proposed work

Today, various open-source APIs and libraries provide precise live stock prices, such as Alpha Vantage, Yahoo Finance, Zirra, Twelve Data, Google Finance, Financial Modeling Prep, etc. [11] The collection of data to carry out our work was obtained from the Google Finance API (for training the prediction models) and yfinance python library [12] (to fetch live streaming data). The proposed system consists of the following modules:

A. Closing Price Analysis

This module aims to build a system that predicts the closing price of a stock with as much accuracy as possible. The proposed algorithm for the same is the stochastic gradient descent algorithm [9] with the incremental learning method, available in the scikit-learn library [13], on top of it.
Input: Live stock data (Open, High, Low, Volume)
Output: Live prediction of the closing price

Method:

```
closing_price_analysis{
        while (the stock market is open)
                load historical data from google finance API
                scale the data
            fit the regression model
            save the model to github [14] repository using PyGithub library [15]
            A: fetch last minute's stock data with the help of
            yfinance library
            load the regression model
            scale the fetched data and give it as input to the model
            predict the closing price for the fetched data
            transform the predicted value back to original
            calculate the error between fetched and the transformed predicted value
            sleep for 60 seconds
            if the stock market is open
            go to A
    load the regression model using the PyGithub library
    fetch the last day's minute wise data using yfinance library
    apply the partial_fit method of scikit-learn library and fit newly fetched data to the model
    save the updated model to the GitHub repository using the PyGithub library

}
```

B. Stock Pattern Detection and Suggestion

In this module, a scanner has been developed that continuously looks for the following two intraday trading strategies among the stock prices: (a) Higher High and (b) Lower Low. These patterns have been finalized for several reasons. Firstly, it is a standard pattern used by almost all brokers in the market. Secondly, it is reliable in the majority of cases. The algorithm for the scanner was built entirely from scratch as, presently;

there, is no readily available method in Python to detect these patterns. For this task of detecting patterns, streaming 'High' and 'Low' value data of each stock was used with the help of the finance library.

Higher High [16] is a bullish continuation pattern of a minimum of 3 candles where the current candlestick breaks the 'high' of the previous candlestick, whereas Lower Low [16] is a bearish continuation pattern of a minimum of 3 candles where the current candlestick[2] breaks the 'low' of the previous candlestick. Once a higher high pattern is detected by the scanner, it continues to analyze the data for higher high and lower low patterns but with a slight change in the conditions. Now, for the higher high pattern to occur again, the first of the three consecutive candlesticks must break the 'high' of the third candle in the previous higher high pattern. In the same way, for the lower low pattern to occur again, the first of the three consecutive candlesticks must break the 'low' of the first candle in the previous higher high pattern. Once a lower low pattern is detected by the scanner, it continues to analyze the data for higher high and lower low patterns but with a slight change in the conditions. Now, for the higher high pattern to occur again, the first of the three consecutive candlesticks must break the 'high' of the first candle in the previous lower low pattern. In the same way, for the lower low pattern to occur again, the first of the three consecutive candlesticks must break the 'low' of the third candle in the previous lower low pattern.

Figure. 1 - Intraday Trading Strategies for (a) higher high and (b) lower low

C. Calculating Stock Volatility

In its simplest sense, volatility [17] can be described as an indicator of how much the stock price might vary. Volatility is a crucial factor when it comes to deciding which stock to invest in. Traders can use this information to estimate the potential deviation of the price from the average. Highly volatile stocks are those whose prices fluctuate wildly, hit new highs and lows, or move haphazardly. Stocks with low volatility tend to maintain a relatively stable price [20, 27]. Investing in an extremely volatile stock is inherently risky.

The calculations were made by using the population standard deviation of the closing price. The higher the worth, the more volatile the costs or the returns. In other words, a high value indicates prices are spread across a large spectrum. Mathematically, it is calculated as follows:

$$\sigma = \sqrt{\frac{\Sigma(x_i - \mu)^2}{N}} \qquad (1)$$

Where,

σ = population standard deviation

[2] A candlestick is a chart that displays the high, low, opening, and closing prices of a stock for a specific period. The wide part of the candlestick tells u opening price. Red indicates that the stock closed lower and green indicates that the stock closed higher.

$$N = \text{the size of the population}$$
$$x_i = \text{each value from the population}$$
$$\mu = \text{the population mean}$$

4. Experiments and result discussion

The efficiency and accuracy of the algorithms used in the experiments have been analyzed through the performance parameters such as Mean Absolute Error (MAE) [18, 25], Mean Square Error (MSE) [18, 25], Root Mean Square Error (RMSE) [18, 25], R-Squared (R2) [18, 25] and Accuracy.

MAE is calculated using eq. 2.

$$MAE = \frac{1}{N}\sum_{i=1}^{N} |y_i - \hat{y}| \qquad (2)$$

where 'y_i' = original closing price, '$\hat{y}$' = predicted closing price and 'N' = total window size.

MSE is calculated using eq. 3.

$$MSE = \frac{1}{N}\sum_{i=1}^{N} (y_i - \hat{y})^2 \qquad (3)$$

where 'y_i'= original closing price, '$\hat{y}$' = predicted closing price, and 'N' = the total window size.

RMSE is calculated using eq. 4.

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(y_i - \hat{y})^2}{n}} \qquad (4)$$

where 'y_i' = original closing price, '$\hat{y}$' = predicted closing price and 'N' = total window size.

R2 score is calculated using eq. 5.

$$R^2 = 1 - \frac{\Sigma(y_i - \hat{y})^2}{\Sigma(y_i - \underline{y})^2} \qquad (5)$$

where 'y_i' = original closing price, '$\hat{y}$' = the predicted closing price, '$\underline{y}$' = the mean of closing prices, and 'N' = the total window size.

Accuracy is calculated using eq. 6.

$$Accuracy = (1 - \frac{MAE}{\underline{y}}) \times 100 \qquad (6)$$

GoogleFinance API fetches day-wise data and is hence used for training the model. The yfinance library, on the other hand, can fetch data in various periods like minute-wise, hour-wise, etc., and has been used for testing the model. Approximately 10 years of stock data, which amounts to 2800 data points, is considered for training initial models of every stock. Testing data was collected after the market hours and summed up to 375 records. Our dataset, as shown in Table I, is based on six important attributes: Timestamp, Open, High, Low, Close, and Volume. Each attribute plays an important role in recording the stock price movements and trading activities throughout the day. The timestamp provides the temporal context, the price metrics being Open, High, Low, and Close, which capture the stock price range and movement patterns the Volume attribute shall indicate the trading intensity to offer insights about market participation and liquidity levels.

TABLE 1 DATASET DESCRIPTION

Attributes in the dataset	**Description**
Timestamp	The date and time when the price was recorded
Open	The opening price of the stock
High	The highest price level reachedduring that day
Low	The lowest price level reached during that day
Close	The closing price of the stock
Volume	Number of stocks traded during that day

a) To analyze closing prices, 14 regression algorithms have been implemented on the data of six stocks (Apple, Google, Microsoft, Tesla, IBM, Accenture). As shown in Table II, these algorithms include a diverse range of approaches from Random Forest and Decision Tree Regression to more sophisticated methods like Gaussian Process Regression and Linear SVR. The errors were obtained after applying these algorithms on minute-wise test data (obtained from the yfinance library) of the six stocks. These errors were noted down in four Google sheets to perform analysis. All the sheets contain the same evaluation criteria (MAE, MSE, RMSE, R2, Accuracy) but differ in the type of data used to calculate errors (scaled & unscaled) and parameters of the ML algorithms (default & tuned) thus resulting in 4 sheets.

The implemented algorithms [9] used from the scikit-learn library are as follows:

TABLE 2 LIST OF REGRESSION ALGORITHMS

Random Forest Regression	Decision Tree Regression
PLS Regression	Nearest Neighbours Regression
AdaBoost Regression	ARD Regression
Gradient Boosted Regression	Stochastic Gradient Regression

Random Forest Regression	Decision Tree Regression
Bayesian Ridge Regression	Gaussian Process Regression
Linear Regression	Support Vector Regression
MLP Regression	Linear SVR

Figures 2-7 show the RMSE values to represent the performance of each of the 14 algorithms on each of the six stocks.

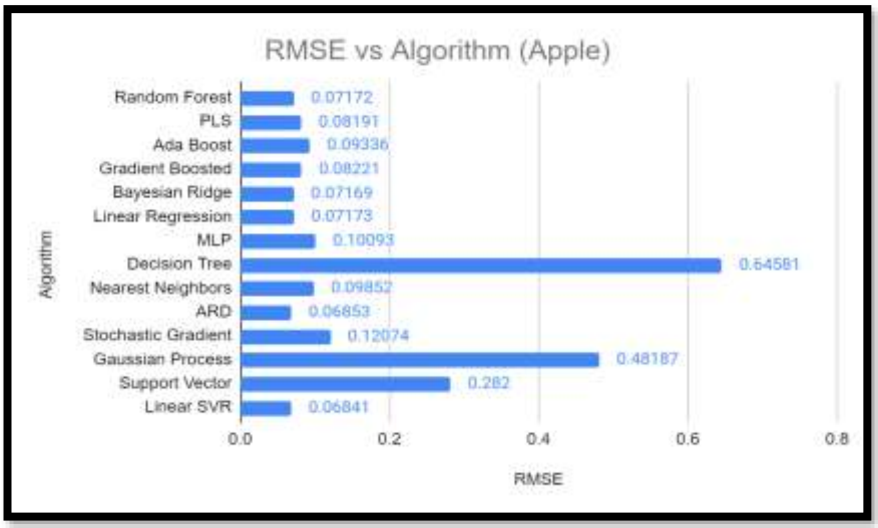

Figure. 2 - RMSE of algorithms on 'Apple' Data

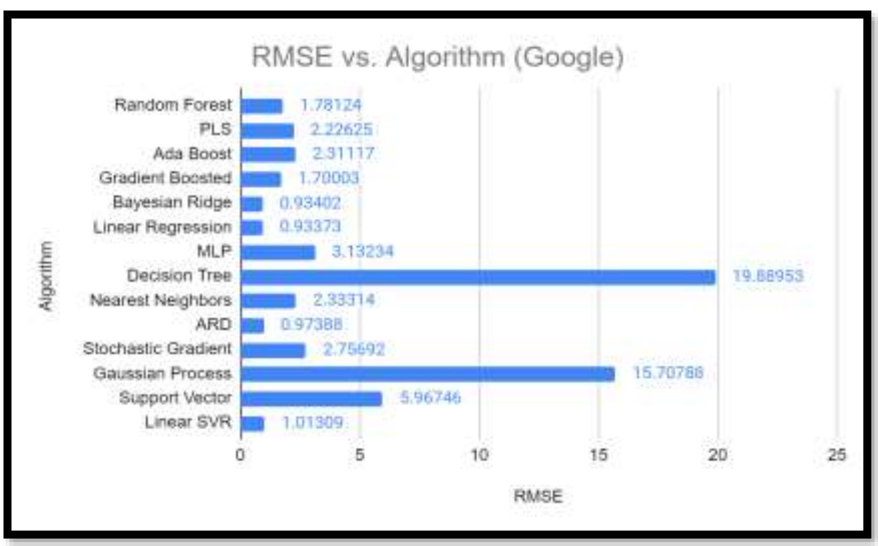

Figure. 3 - RMSE of algorithms on 'Google' Data

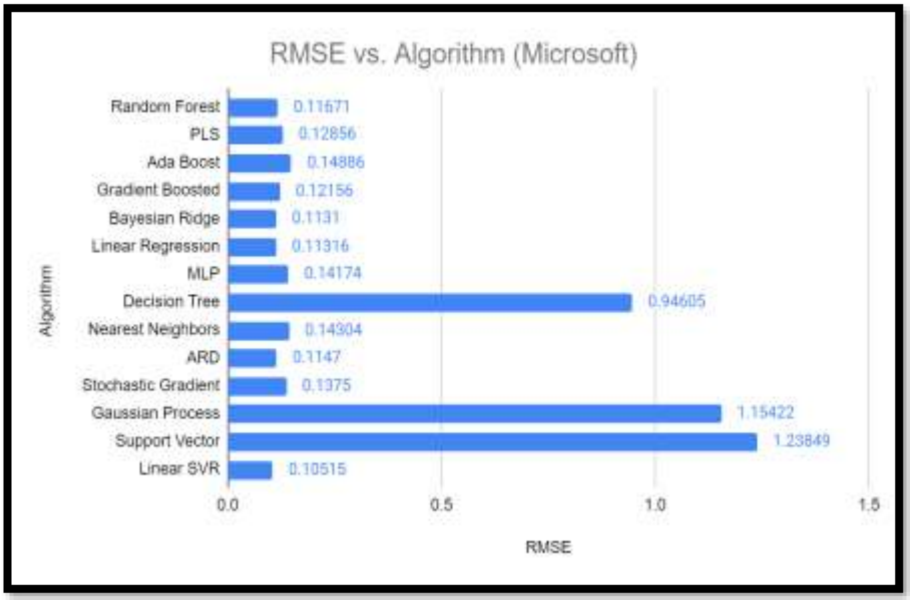

Figure. 4 - RMSE of algorithms on 'Microsoft' Data

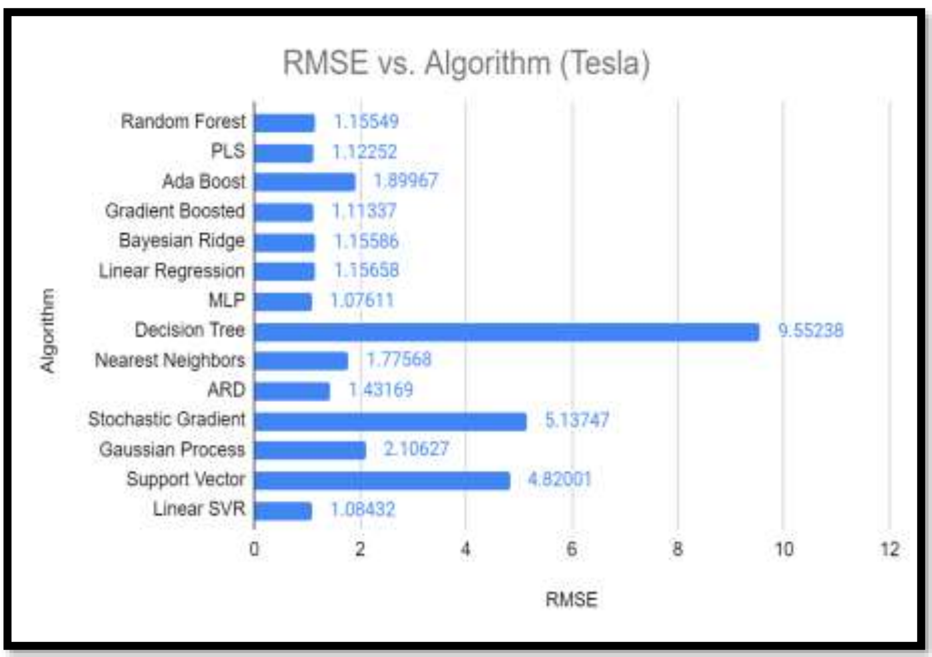

Figure. 5 - RMSE of algorithms on 'Tesla' Data

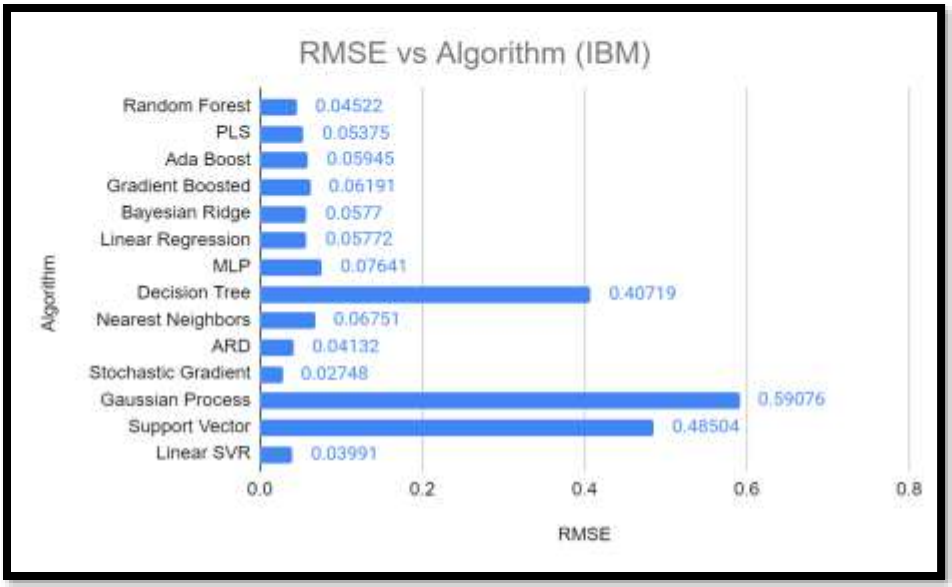

Figure. 6 RMSE of algorithms on 'IBM' Data

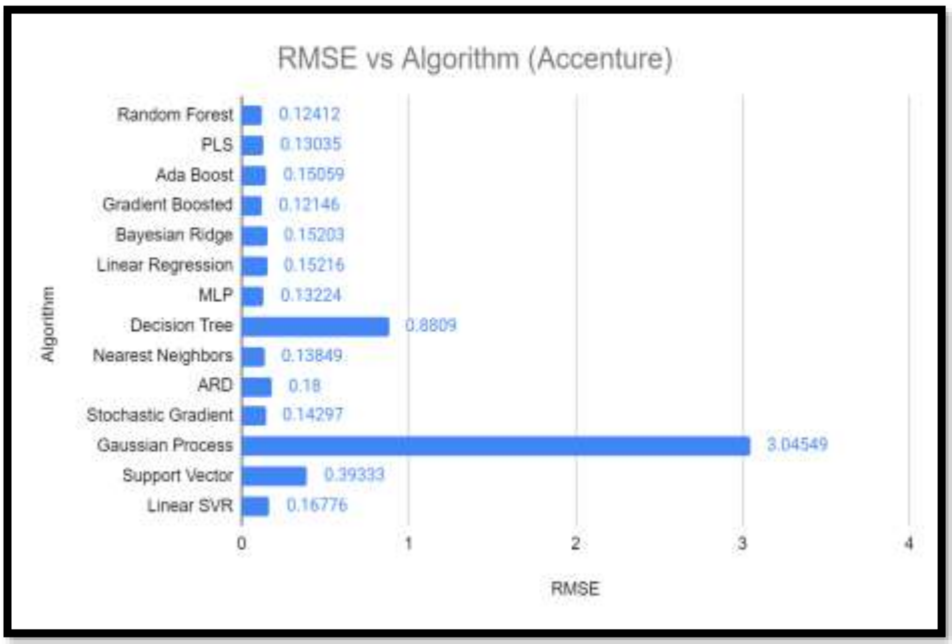

Figure. 7 - RMSE of algorithms on 'Accenture' Data

Table III shows the most efficient algorithm for predicting the closing price for each stock. Because the pattern of price change in each stock is unique, different algorithms perform differently for each stock.

TABLE III CONCLUSION OF SHEET ERROR ANALYSIS

Stock	Best Algorithm	MAE	MSE	RMSE	R2	Accuracy
Apple	ARD	0.0549	0.00469	0.06853	0.99223	99.9534
Google	Linear	0.7333	0.87185	0.93373	0.99886	99.9646
Microsoft	Linear SVR	0.8394	0.01105	0.10515	0.99242	99.9636
Tesla	MLP	0.8562	1.15799	1.07610	0.99412	99.8536
IBM	Linear SVR	0.0304	0.00159	0.03991	0.99725	99.9757
Accenture	Gradient Boosted	0.0897	0.01475	0.12146	0.99119	99.9645

The above conclusions were made by analyzing which algorithm has the majority evaluation measures with the most favorable values. Taking the example of Apple's data, in New_Scaled and New_Transformed sheets, ARD Regression consistently outperformed the other algorithms by having the lowest MAE, MSE, and RMSE errors and the highest R2 score and Accuracy. Even though in New_Tuning_Scaled and New_Tuning_Tranformed sheets Gaussian Process Regression outperformed, the difference between its errors and ARD Regression's errors is insignificant, while on the other hand, the difference between its errors and ARD Regression's errors in New_Scaled and New_Transformed sheets is more significant. Therefore, ARD Regression has been selected as the best algorithm for Apple stock data. Similar logic has also been applied to select the best algorithms for other stocks.

b) Another experiment was conducted to observe the effect of implementing incremental learning on a machine learning algorithm. An implementation of SGD+incremental learning was created using scikit-learn methods and PyGithub library, its performance was noted with all stocks at the same time as observing the results of the other algorithms (without incremental learning) and after almost a month of SGD Regression training incrementally, the errors were noted again. Even though SGD is not the best algorithm, incremental learning will improve its performance in the long term, exceeding the efficiency of the algorithms (without incremental learning) in Table III. The results from our second experiment, which was conducted to prove the same point, are mentioned in Table IV and Table V.

TABLE IV PERFORMANCE MEASURES OF SGD+INCREMENTAL LEARNING (NOTED AT THE SAME TIME AS TABLE III)

Stock	MAE	MSE	RMSE	R2	Accuracy
Apple	0.0960	0.0119	0.1092	0.9805	99.92783
Google	0.7912	1.0611	1.0301	0.9945	99.96533
Microsoft	0.1016	0.0161	0.1270	0.9959	99.96077
Tesla	1.1494	1.7973	1.3406	0.9769	99.84336
IBM	0.1033	0.0143	0.1198	0.9771	99.92729
Accenture	0.0680	0.0087	0.0937	0.9933	99.97664

TABLE V PERFORMANCE MEASURES OF SGD+INCREMENTAL LEARNING (NOTED AFTER ALMOST A MONTH)

Stock	MAE	MSE	RMSE	R2	Accuracy
Apple	0.0425	0.0033	0.0579	0.9655	99.96571
Google	0.5917	0.5678	0.7535	0.9931	99.9747
Microsoft	0.0949	0.0147	0.1213	0.9675	99.96068
Tesla	0.4673	0.4184	0.6468	0.9664	99.91619
IBM	0.0526	0.0046	0.0679	0.9807	99.96300
Accenture	0.0461	0.0037	0.0610	0.9622	99.98375

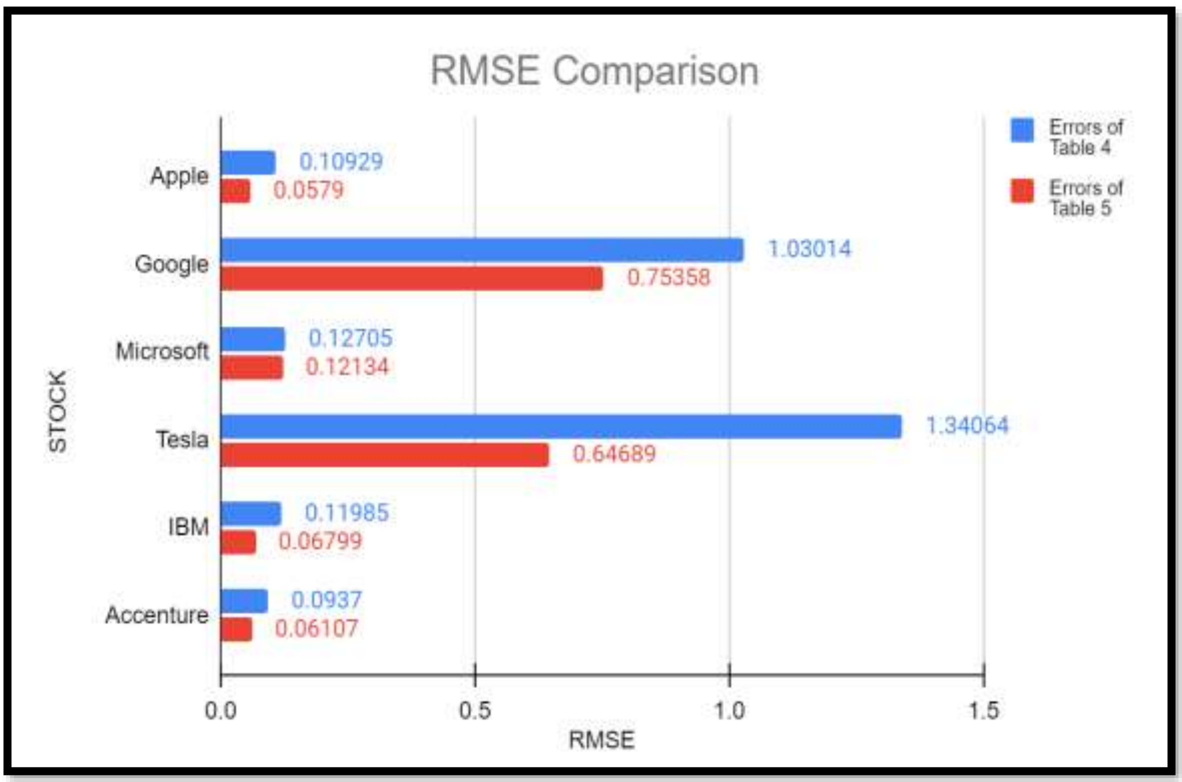

Fig. 8 - Performance improvement over one month

5. System demonstration

This chapter shows the user interface developed for the Stock Pattern Detection & Suggestion and Stock Volatility modules of our work and explains its characteristics.

A. Stock Pattern Detection

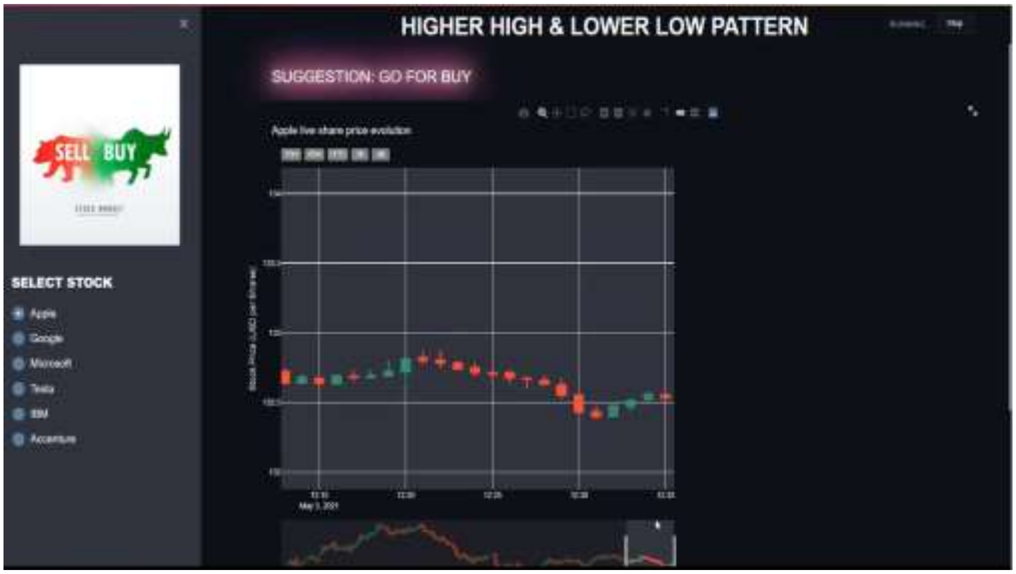

Fig. 9 - Stock Pattern Detection & Suggestion Page

To choose from a list of stocks, a navigation bar has been provided. Every stock has its radio button, which tells the user that only one stock can be selected at a time. The body of the page contains suggestions regarding buying/selling shares. This decision is based on the algorithm that was created. If a 'Higher High' pattern is detected, it means that the stock price is going to rise. Therefore, the suggestion given is to 'Buy' that stock. Similarly, if a 'Lower Low' pattern is detected, it means that the stock price is going to fall. Therefore, the suggestion given is to 'Sell' that stock. It also consists of a candlestick chart that gets updated in real-time. The user can view the chart for any time frame with the help of a range slider provided just below the graph. For immediate access, five-time periods are provided as a button right above the chart, namely, 15 minutes, 45 minutes, 1 hour, 3 hours, and the entire day.

B. Stock Volatility

Fig. 10 - Stock Volatility Page

The navigation column gives a general understanding of the concepts of volatility for users who are not very familiar with it. As soon as the user selects the number of months, day-wise data for that many months is fetched for each stock and then the function of population standard deviation is applied to the closing price data of each stock. This is how volatility is calculated for all the stocks for the given time frame. Simply put, if a user wants to calculate the volatility of stock for the past 4 months, then the population standard deviation will be calculated on the closing price of the past 4 months. This value will indicate the extent to which the price for said stock can fluctuate in the next 4 months. The stock name has been displayed alongside its volatility in a tabular form in ascending order and also a graph for better visualization.

6. Conclusion & future work

In this paper, different regression algorithms are explored to find the most suitable algorithm for the task of stock price prediction. Different experiments are conducted on well-known datasets. However, it is not possible to find the best algorithm due to the different characteristics of the datasets and algorithms. The work presented here proposes the incremental learning approach of Stochastic Gradient Descent for predicting the price of different stocks. The work further considers the impact of news articles and global events while doing prediction analysis and creating methods for incremental learning in other algorithms as well.

References

1. Loginov, A., Heywood, M., & Wilson, G. (2021). Stock selection heuristics for performing frequent intraday trading with genetic programming. Genetic Programming and Evolvable Machines, 22(1), 35-72.

2. Behera, R. K., Das, S., Rath, S. K., Misra, S., & Damasevicius, R. (2020). Comparative Study of Real Time Machine Learning Models for Stock Prediction through Streaming Data. Journal of Universal Computer Science, 26(9), 1128-1147.

3. Yahoo Finance - Business Finance Stock Market News, [Accessed on August 16, 2018]

4. Vijh, M., Chandola, D., Tikkiwal, V. A., & Kumar, A. (2020). Stock closing price prediction using machine learning techniques. Procedia Computer Science, 167, 599-606.

5. Selvin, S., Vinayakumar, R., Gopalakrishnan, E. A., Menon, V. K., & Soman, K. P. (2017, September). Stock price prediction using LSTM, RNN and CNN-sliding window model. In 2017 international conference on advances in computing, communications and informatics (icacci) (pp. 1643-1647). IEEE.

6. Mei, J., He, D., Harley, R., Habetler, T., & Qu, G. (2014, July). A random forest method for real-time price forecasting in New York electricity market. In 2014 IEEE PES General Meeting| Conference & Exposition (pp. 1-5). IEEE.

7. Gepperth, A., & Hammer, B. (2016). Incremental learning algorithms and applications. In European symposium on artificial neural networks (ESANN).

8. Xu, S., & Wang, J. (2016). A fast incremental extreme learning machine algorithm for data streams classification. Expert Systems with Applications, 65, 332-344.

9. Ruping, S. (2001, November). Incremental learning with support vector machines. In Proceedings 2001 IEEE International Conference on Data Mining (pp. 641-642). IEEE.

10. Géron, A. (2019). Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow: Concepts, tools, and techniques to build intelligent systems. O'Reilly Media.

11. RapidAPI Staff, 2021, April 16, Top 7 Best Stock Market APIs (for Developers) [2021], https://rapidapi.com/blog/best-stock-api/

12. Yan, Y. (2017). Python for Finance. Packt Publishing Ltd. (Page 59)

13. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... & Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. the Journal of machine Learning research, 12, 2825-2830.

14. McDonald, N., & Goggins, S. (2013). Performance and participation in open source software on github. In CHI'13 Extended Abstracts on Human Factors in Computing Systems (pp. 139-144).

15. Tomi‘bgt’Suovuo, J. H., Smed, J., & Leppänen, V. Mining Knowledge on Technical Debt Propagation.

16. Higher-highs and Higher-lows vs Lower-highs and Lower-lows, https://blog.bettertrader.co/technical-analysis/higher-highs-and-higher-lows-vs-lower-highs-and-lower-lows/

17. Jia, F., & Yang, B. (2021). Forecasting Volatility of Stock Index: Deep Learning Model with Likelihood-Based Loss Function. Complexity, 2021.

18. Willmott, C. J. (1982). Some comments on the evaluation of model performance. Bulletin of the American Meteorological Society, 63(11), 1309-1313.

19. Gurav, U., & Sidnal, N. (2018). Predict Stock Market Behavior: Role of Machine Learning Algorithms. Advances in Intelligent Systems and Computing, 383–394. doi:10.1007/978-981-10-7245-1_38.

20. Henrique, B. M., Sobreiro, V. A., & Kimura, H. (2018). Stock price prediction using support vector regression on daily and up to the minute prices. In The Journal of finance and data science, 4(3), 183-201.

21. Loginov, A. (2020). On Increasing the Scope of Genetic Programming Trading Agents.

22. Hiransha, M., Gopalakrishnan, E. A., Menon, V. K., & Soman, K. P. (2018). NSE stock market prediction using deep-learning models. Procedia computer science, 132, 1351-1362.

23. Lawal, I. A. (2019). Incremental SVM learning. In Learning from data streams in evolving

environments (pp. 279-296). Springer, Cham.

24. Losing, V., Hammer, B., & Wersing, H. (2018). Incremental on-line learning: A review and comparison of state of the art algorithms. Neurocomputing, 275, 1261-1274.

25. Willmott, C. J., Ackleson, S. G., Davis, R. E., Feddema, J. J., Klink, K. M., Legates, D. R., ... & Rowe, C. M. (1985). Statistics for the evaluation and comparison of models. Journal of Geophysical Research: Oceans, 90(C5), 8995-9005.

26. Vadlamudi, S. (2017). Stock Market Prediction using Machine Learning: A Systematic Literature Review. American Journal of Trade and Policy, 4(3), 123-128.

27. Rustam, Z., & Kintandani, P. (2019). Application of Support Vector Regression in Indonesian Stock Price Prediction with Feature Selection Using Particle Swarm Optimisation. Modelling and Simulation in Engineering, 2019.

28. Dehghani, M., Ghasemzadeh, M., & Ansari-samani, H. (2019). Machine learning algorithms for time series in financial markets. Journal of Soft Computing and Information Technology, 8(3), 60-67.

29. Singh, A., Gupta, P., & Thakur, N. (2021). An Empirical Research and Comprehensive Analysis of Stock Market Prediction using Machine Learning and Deep Learning techniques. In IOP Conference Series: Materials Science and Engineering (Vol. 1022, No. 1, p. 012098). IOP Publishing.

30. Glas, S., Kiesel, R., Kolkmann, S., Kremer, M., Graf von Luckner, N., Ostmeier, L., ... & Weber, C. (2020). Intraday renewable electricity trading: Advanced modeling and numerical optimal control. Journal of Mathematics in Industry, 10, 1-17

31. Arumugam, D. (2023). Algorithmic trading: Intraday profitability and trading behavior. Economic Modelling, 128, 106521

32. Singh, T., Kalra, R., Mishra, S., Satakshi, & Kumar, M. (2023). An efficient real-time stock prediction exploiting incremental learning and deep learning. Evolving Systems, 14(6), 919-937

33. Ammar, I. B., & Hellara, S. (2022). High-frequency trading, stock volatility, and intraday crashes. The Quarterly Review of Economics and Finance, 84, 337-344