MACHINE LEARNING ALGORITHMS USING STOCK MARKET DATASET–A COMPARATIVE STUDY

SAMUEL BINOY VARKEY[1], BELFIN R.V[2], G.ROBINSON PAUL[3]

[1,2] Department of CSE, Karunya Institute of Technology and Sciences, Coimbatore, India
[3] Department of ECE, Karunya Institute of Technology and Sciences, Coimbatore, India
[1]samuelvarkey.work@gmail.com, [2]researchbelfin@gmail.com, [3]robinson.paul@gmail.com

ABSTRACT:

Machine Learning is an important aspect that has had a drastic impact on our daily lives. From healthcare to social media, machine learning has had a game-changing impact across several industries. With the help of ML, Amazon and Netflix decides what specific products or shows to recommend for the particular user. One very important feature of ML is its ability to perform prediction and forecasting and this is extensively used in the Finance sector. The Stock Market is one of the most unpredictable aspects of our daily lives. Financial analysts and Investment Bankers have the task of performing predictions in such an ever-changing environment. Therefore, the use of ML in this domain is needed to allow these professionals to make better decisions with regards to people and companies portfolios. Stock market prediction has attracted people regardless of their affiliation with the finance sector. This is due to fact that decision making of business leaders heavily depends on the behaviour of this metric. Furthermore, due to its uncertain patters, predicting it to the highest accuracy possible is extremely vital. In this paper, we will conduct stock market prediction using various forms of regression and neural networks. The regression techniques are Random Forest Regression, XGBoost, Linear Regression, Support Vector Regressor (SVR) and Artificial Neural Networks(ANN). We will use 3 metrics to compare accuracies of the algorithms: RMSE (Root Mean Square Error), R2 score (R squared score) and MAE (Mean Absolute Error). In conclusion, we will be comparing the algorithms used and figure out the one giving best results for this dataset. We will also be comparing the algorithms with the benchmark algorithm i.e. SVM and verify the best performing one.

I. INTRODUCTION

Stock market prediction is an interesting topic for several researchers across various fields. Machine Learning is now used extensively for forecasting in the financial markets [1]. Popular algorithms, including neural networks and other regression techniques like Random Forest Regression, Linear Regression, XGBoost, SVM have been reported to be quite effective in tracing the stock market. In this paper, we will explore in-depth what each algorithm has to offer and in the end we will conclude with selecting the most efficient algorithm out of all.

- "Random Forest uses Ensemble Learning for classification and regression. The Random Forest is a baggage technique, not a boosting technique. The trees in the random forest are running in parallel[2]."
- "Linear regression is the analysis of the relation between a dependent variable and one or more independent variables[3]."
- XGBoost is a decision-tree based on gradient-boosting[4].
- Support vector machines are supervised learning algorithms used for regression and classification[5].
- Artificial neural networks (ANNs) are computational mechanisms modelled by biological neural networks that resemble animal brains[6].

II. LITERATURE

Although predicting the stock market is a close to impossible task due to the everyday changes it undergoes, it still remains extremely important. Predicting the stock market perfectly is impossible, however, with the help of ML, we can analyse with various algorithms the closest trend that will exist in the future. Figuring out the trends is extremely important for business leaders especially CFOs to make business decisions. A company's decision to acquire or invest in a firm all depends on the stock market that we predict and therefore, it is of utmost importance to give them the most accurate results. Financial analysts, investment bankers and investment advisors are all depending on Machine Learning and Data Science to analyse the risks affiliated with the stock market in order to provide the best results and to effectively manage their portfolios.

In the past, several algorithms has been used to conduct the prediction of stock markets. One of the research conducted uses SVR, SVR–ANN, SVR–RF and SVR–SVR fusion prediction models [7]. This study compares hybrid models with each other but does not measure the accuracy of each individual algorithm. Another study uses SVM exclusively, but they did not compare algorithms with each other [1]. Further studies include a hybrid machine learning system based on "Genetic Algorithm (GA) and Support Vector Machines (SVM) [8]", "ANN, Decision tress, KNN [9]", "Single Layer Perceptron (SLP), Multi-Layer Perceptron (MLP), Radial Basis Function (RBF) and Support Vector Machine (SVM) [10]", "Regression [11]", "an integrated model combining Principal Component Analysis (PCA) and Support Vector Machine (SVM) [12]", "Reinforcement learning [13]". With these references, we can notice a pattern generating indicating that SVM is very popular for stock market prediction and is used either on its own or as a hybrid with other algorithms. With the help of our study, we will be able to verify whether SVM deserves to be the most popular algorithm in stock market forecasting or should be replaced by another algorithm. In all the previous studies, the study comprises combining algorithms to create hybrid algorithms. Therefore, in this paper, we will compare algorithms individually without creating hybrid models to allow us to understand the best individually performing algorithms. In future studies, the best performing algorithms can be combined to create an even better performing hybrid algorithm.

III. METHODOLOGY

We will be using the following algorithms and we will compare them with the commonly used SVM and figure out the best algorithm. We will run the dataset on the algorithms, visualize the results and compare the results in the end in a tabular format.

3.1 Dataset

The dataset is an Exchange-Traded Fund (ETF) dataset from Kaggle [14]. An exchange-listed fund, much like stocks, is an investment portfolio listed on a stock market. An ETF retains properties such as stocks,bonds or commodities and typically works through an arbitration system structured to maintain it similar to its net asset value, while deviations may sometimes exist.[15]. The dataset has 7 labels, they are 'Date', 'Open',' High', 'Low', 'Close', 'Volume' and 'OpenInt'. Out of these 'Close' is our target variable and the rest are our features for the model. 'Open' is the price of the first trade for a stock for the day. This metric acts as a key to what trading activities will follow through the day. 'High' and 'Low' means the highest and lost price at a given time. 'Volume' is the number of stocks exchanged in the day. 'OpenInt' is the Open Interest and it is the number of pending contracts owned by participants at the end of the day. It increases by one when a new transaction occurs i.e. a new seller and a new buyer are initiating the transaction[16]. 'Close' is the last price at which a stock trades during a session.

3.2 Pre-processing

"Data Pre-processing is done using the MinMaxScaler. MinMaxScaler subtracts the minimum of the feature and then divides it by the range. The difference between the original maximum and the original minimum is the distance. MinMaxScaler retains the shape of the original distribution. It does not significantly change the information embedded in the original data [17]."

3.3 Algorithms

In the following segment, we will be having a detailed analysis of the algorithms being used.

3.3.1. Random Forest Regression

Bootstrap aggregation (bagging) is the technique used for training Random Forests. Assume a data set $X = x_1, ..., x_n$ with outputs $Y = y_1, ..., y_n$, repeatedly bagging (B times) chooses a sample by replacing the fitting trees to these samples [18]. For $b = 1, ..., B$:

1. Sample, after replacing, n training examples from the dataset; we will call them X_b, Y_b.
2. Perform a regression tree f_b on X_b, Y_b.

Post training, prediction of the unknown samples *x′* can be done by calculating the average of all the predictions from the individual regression trees on *x′* [18].

$$\boldsymbol{f = \frac{1}{B}\sum_{b=1}^{B} f_b(x')} \qquad (1)$$

By using standard deviation predicted from the trees, we can estimate the prediction's uncertainity[18].

$$\boldsymbol{\sigma = \sqrt{\frac{\sum_{b=1}^{B}(f_b(x')-f)^2}{B-1}}} \qquad (2)$$

3.3.2. Linear Regression

Linear regression is used to model the relation between two variables. The equation is the **slope formula**. The equation has the form Y= a + bX, where Y is the dependent variable, X is the independent variable, b is the slope of the line and a is the y-intercept [19].

$$\boldsymbol{a = \frac{(\sum y)\ \ (\sum x^2)-(\sum x)(\sum xy)}{n(\sum x^2)-(\sum x)^2}} \qquad (3)$$

$$\boldsymbol{a = \frac{n(\sum xy)-(\sum x)(\sum y)}{n(\sum x^2)-(\sum x)^2}} \qquad (4)$$

3.3.3. Support Vector Regression

SVR is used to find the points within the boundaries of the decision line. A hyperplane with a maximum number of points is the best fit line [20]. Assume 2 lines as being at a distance 'a', from the hyperplane. Let the equation of the hyperplane be as follows.

$$\boldsymbol{Y = wx + b}$$

Decision boundary becomes as follows for the above hyperplane:

$$\boldsymbol{a = wx + b} \qquad (5)$$

$$\boldsymbol{-a = wx + b} \qquad (6)$$

Any hyperplane that satisfies the SVR should satisfy:

$$\boldsymbol{-a < Y - wx + b < a} \qquad (7)$$

3.3.4. XG Boost

Contrary to bagging techniques, where trees are grown to their maximum extent, trees with fewer splits are used in boosting. Small trees are more interpretable and are not very deep. Validation techniques like k-fold cross validation can be used to optimally select parameters like the number of trees or iterations, the rate of gradient boosting learning, and the depth of the tree. **Overfitting can be caused by having a lot of trees[21].Therefore, choosing the stopping criteria is critical for boosting.** These are the steps of Boosting: -

- F0 (initial model) predicts y (the target variable).
- The residuals are fit with a new model (h1).
- F1, the boosted version of F0, is formed by combining F0 and h1. The mean squared error of F1 will be lower than that of F0.[21]

$$\boldsymbol{F_1(x) < -F_0(x) + h_1(x)} \qquad (8)$$

We model the residuals of F1 to create a new model F2 to improve performace of F1 [21].

$$\boldsymbol{F_2(x) < -F_1(x) + h_2(x)} \qquad (9)$$

Until residuals have been minimized as much as possible, we continue this for 'm' iterations [21].

$$\boldsymbol{F_m(x) < -F_{m-1}(x) + h_m(x)} \qquad (10)$$

3.3.5. Artificial Neural Networks

ANNs are composed of artificial neurons which is similar to the concept of neurons. The network consists of connections, where one neuron's output is the input to another neuron. The relative importance of a connection is established by assigning a weight [6].

Let us assume n number of inputs x0, x1, x2, x3...x(n). These inputs are multiplied by a weight. The weights are represented as w0, w1, w2, w3…. w(n) **.**

The product of the weight and input is multiplied and the following product is given, which is later given to the activation function.

$$x_1w_1 + x_2w_2 + x_nw_n = \sum_{i=1}^{n} x_iw_i \qquad (11)$$

The activation function can be moved up or down with the help of the bais value(b)[22]. Adding the bias to the product of weights and inputs is optional and completely depends on the problem statement.

The activation function used in this problem is **Rectified Linear Units (ReLu).** ReLu is the common activation function ANN ranging from 0 to infinity[0,∞)[22].

$$\emptyset \sum(x_i.w_i) \qquad (12)$$

3.4. Accuracy metrics

In order to compare the algorithms, we will be using 3 metrics: RMSE, R2score and MAE

3.4.1. Root Mean Squared Error (RMSE)

RMSE measures the error between two data sets. It compares a predicted value and an actual value. The smaller the RMSE, the more accurate the predictions are[23]. The following equation is the formula for calculating RMSE, where P is predicted value, O is observed value and n is total number of observations.

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(P_i - O_i)^2}{n}} \qquad (13)$$

3.4.2. Rsquared (R2)

R-squared (R^2) is a measure that represents the variance for a dependent variable in comparison to the independent variable (variables) in a regression model [24].

$$R^2 = 1 - \frac{Unexplained\ Variation}{Total\ Variation} \qquad (14)$$

3.4.3. Mean Absolute Error (MAE)

"MAE measures the mean of the errors in the predictions, without considering their direction[25]."

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|x_i - y_i| \qquad (15)$$

3.5. Procedure

In this segment, we will be mapping out a flowchart of the processes to be conducted. Figure 1 visualizes the process involved in this comparision study.

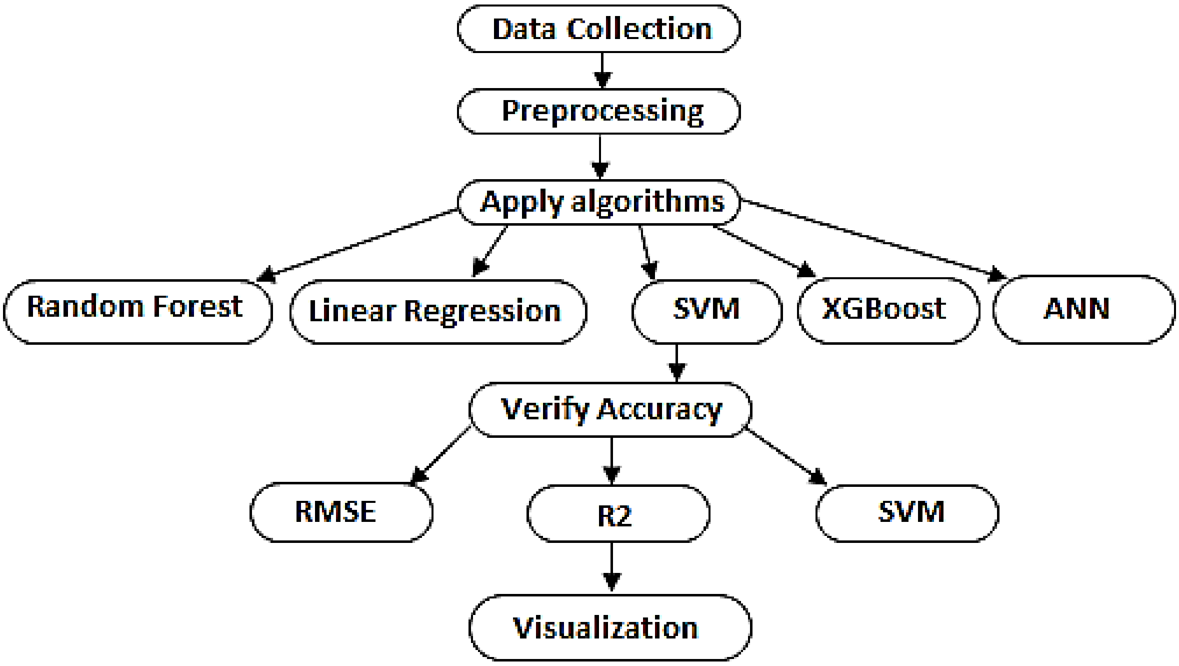


Fig 1. Procedure flowchart for the workflow of processes

IV. RESULTS AND DISCUSSION

For stock market prediction, the most commonly used algorithm is SVM. Therefore, we will train and test our dataset with the SVM regression algorithm and compare the results of the other algorithms with this.

4.1 Results

The metrics we use to compare the algorithms will be RMSE (Root Mean Squared Error), R2 (R squared score) and MAE (Mean Absolute Error).

- **SVM Regression or SVR: -** As mentioned earlier, SVM is one of the most commonly used regression techniques for Stock Market Prediction. The following are the results of executing the dataset on the SVM model.

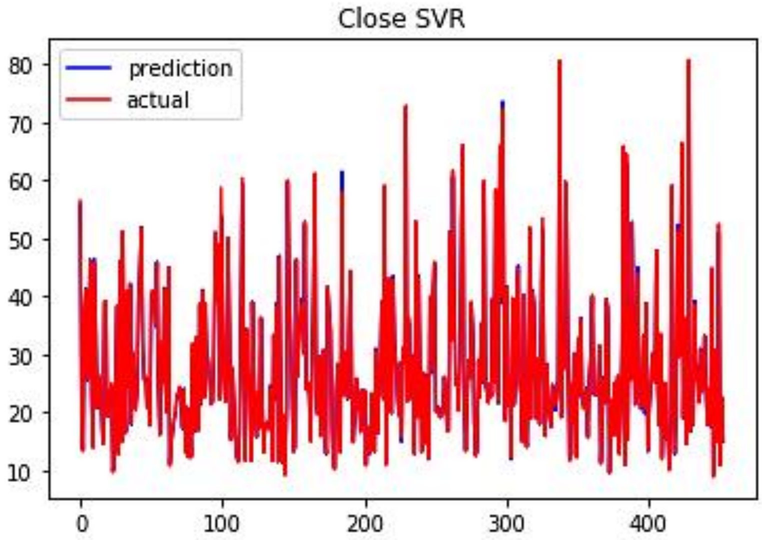


Fig 2. Prediction vs Actual projections using SVR

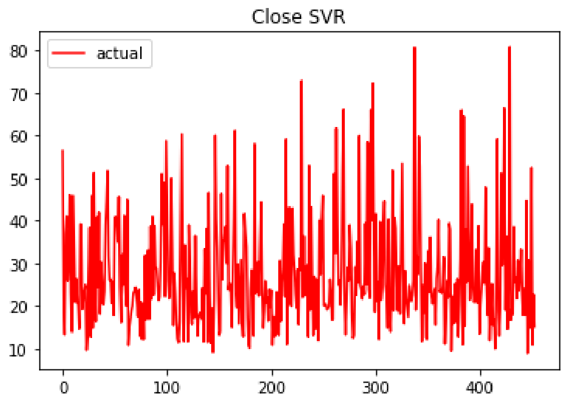


Fig 2.1. Actual projections using SVR

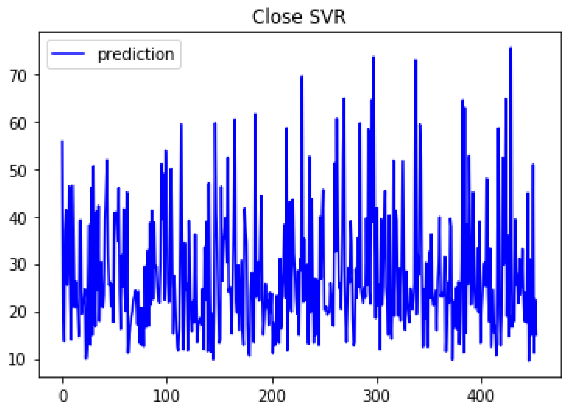


Fig 2.2. Predicted projections using SVR

RMSE: - 0.6940602338830809 (Closer to 0, higher the accuracy).

R2: - 0.9970951059721069 (Closer to 1, higher the accuracy).

MAE: - 0.3468786228164318 (Closer to 0, higher the accuracy).

- **ANN: -** The following are the results of executing the dataset on the ANN model.

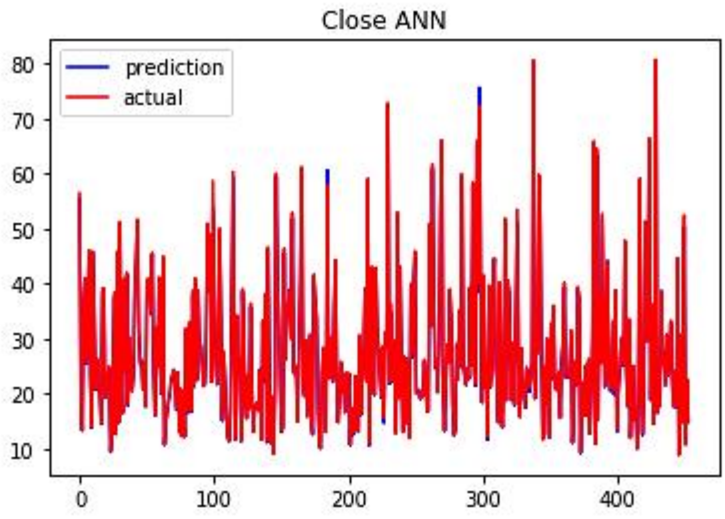


Fig 3. Prediction vs Actual projections using ANN

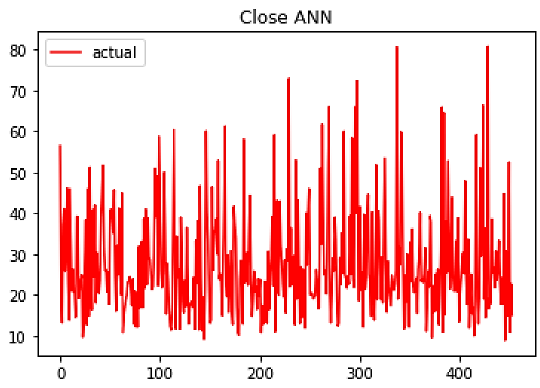


Fig 3.1. Actual projections using ANN

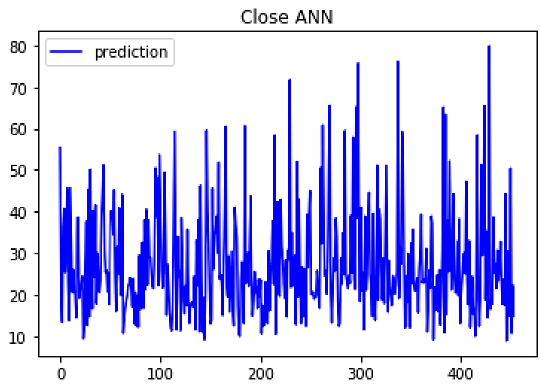


Fig 3.2. Predicted projections using ANN

RMSE: - 0.6156230649383455 (Closer to 0, higher the accuracy).

R2: - 0.9977145815204395 (Closer to 1, higher the accuracy).

MAE: - 0.39259326652206855 (Closer to 0, higher the accuracy).

- **XGB: -** The following are the results of executing the dataset on the XGBmodel.

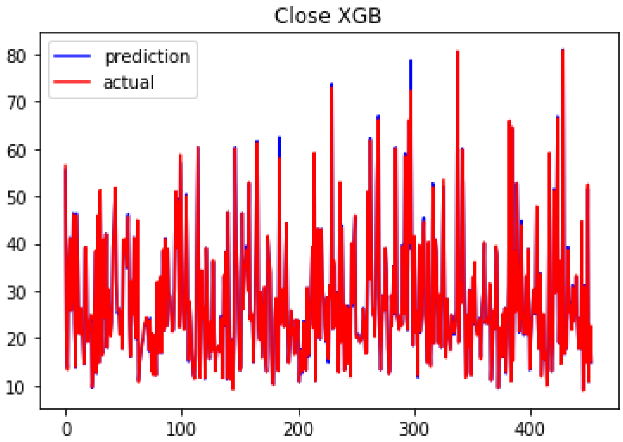


Fig 4. Prediction vs Actual projections using XGB

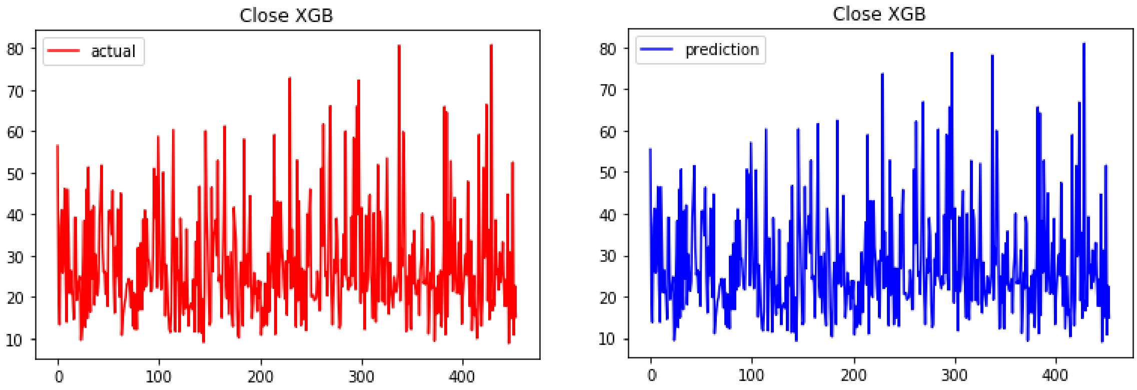


Fig 4.1. Actual projections using XGB **Fig 4.2. Predicted projections using XGB**

RMSE: - 0.5368128879209143 (Closer to 0, higher the accuracy).

R2 score: - 0.9982622718475536 (Closer to 1, higher the accuracy).

MAE: - 0.29015932293148755 (Closer to 0, higher the accuracy).

- **Random Forest Regression: -** The following are the results of executing the dataset on the Random Forest Regression model.

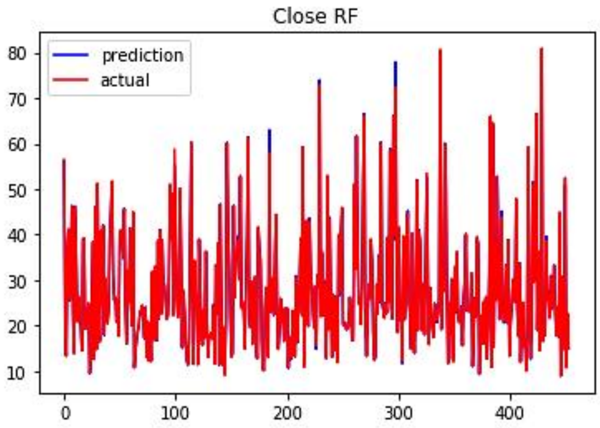
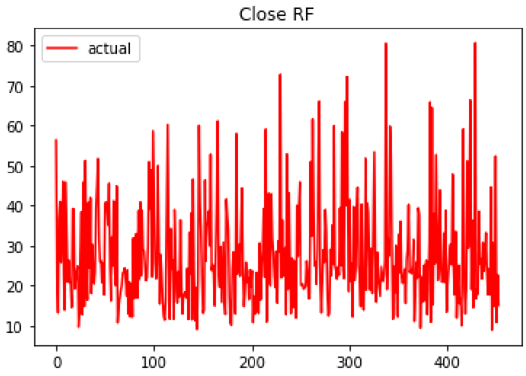


Fig 5. Prediction vs Actual projections using Random Forest



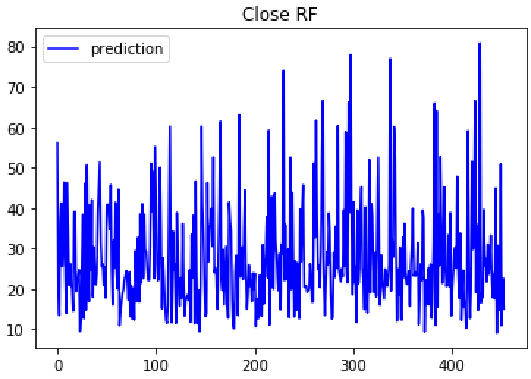


Fig 5.1. Actual projections using Random Forests **Fig 5.2. Predicted projections using Random Forest**

RMSE: - 0.504236375278368(Closer to 0, higher the accuracy).

R2 score: - 0.9981828899476122 (Closer to 1, higher the accuracy).

MAE: - 0.23058035165562643(Closer to 0, higher the accuracy).

- **Linear Regression: -** The following are the results of executing the dataset on the Linear Regression model.

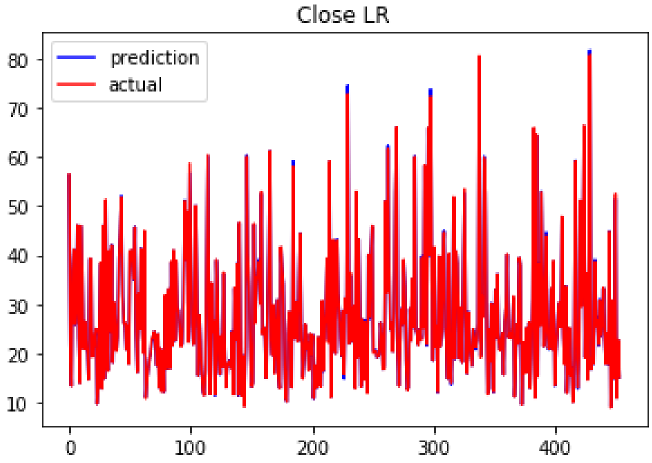


Fig 6. Prediction vs Actual projections using Linear Regression

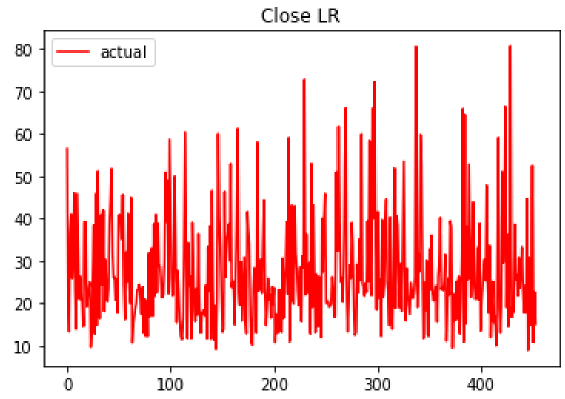


Fig 6.1. Actual projections using Linear Regression

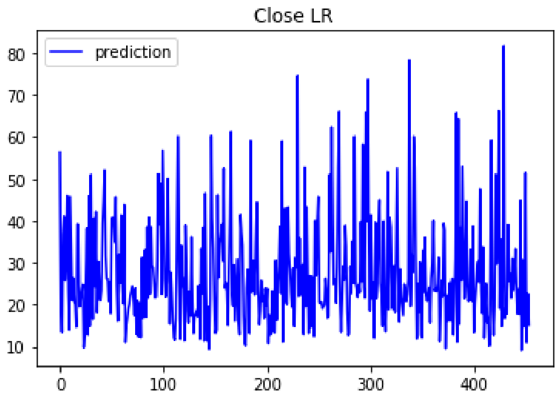


Fig 6.2. Predicted projections using Linear Regression

RMSE: - 0.3011277635594553 (Closer to 0, higher the accuracy).

R2 score: - 0.9994531885919454 (Closer to 1, higher the accuracy).

MAE: - 0.17995621080353494 (Closer to 0, higher the accuracy).

4.2 Comparison and Discussion: - The following table compares all the algorithms used with the SVM algorithm.

Table I. Comparison of algorithms

Algorithm↓ Metric→	RMSE	R2	MAE
SVM	0.6940	0.9970	0.3468
ANN	0.6156	0.9977	0.3925
Random Forest	0.5042	0.9984	0.2305
XGB	0.5368	0.9982	0.2901
Linear Regression	0.3011	0.9994	0.1799

With the help of the above table, we can notice that in all 3 metrics, Linear Regression is showing the most accurate results. However, to verify the accuracy, we will use only RMSE as the R2 values have negligible difference between them and MAE only measures the magnitude difference between the points, ignoring the direction [25].

Using only RMSE, we can notice that Linear Regression outperforms all the other algorithms and most notably, outperforms SVM by 56.61%. We can also note that SVM is the most underperforming among all the used algorithms. SVM is outperformed by ANN (by 11.30%), Random Forest (by 27.35%) and XGB (by 22.65%).

V. CONCLUSION

From the conducted study we can conclude the following: -

- Linear Regression is the best performing algorithm for conducting Stock Market Prediction for the ETF dataset.
- Linear Regression outperforms SVM by 56.61%.
- Among the algorithms tested, SVM showed the least accuracy in comparison to the other algorithms.
- Random Forest and Linear Regression shows the highest accuracy in this prediction and its hybrid can be created in future studies to improve the accuracy.

REFERENCES

1. E. Rosenzweig, “Successful user experience: Strategies and roadmaps,” *Success. User Exp. Strateg. Roadmaps*, pp. 1–344, 2015, doi: 10.1016/c2013-0-19353-1.
2. Chakure, “Random Forest Regression Types of Ensemble Learning :,” 2020. .
3. “Linear regression,” 2020. .
4. V. Morde, “XGBoost Algorithm : Long May She Reign ! What is XGBoost ?,” 2020. .
5. “Support vector machine,” 2020. .
6. “Artificial neural network,” 2020. .
7. Stock Market Analysis using Data Visualization, Bharat Gupta, Shefali Bhardwaj, K. Govinda, Rajkumar. R, International Journal of Recent Technology and Engineering (IJRTE) ISSN: 2277-3878, Volume-8 Issue-6, March 2020
8. Low cost audio based intelligent guidance system for visually impaired people,Vattumilli Komal Venugopal, Alampally Naveen, Rajkumar.R, K. Govinda, Jolly Masih. International Journal of Psychosocial Rehabilitation, ISSN: 1475-7192. DOI: 10.37200/IJPR/V24I3/PR200809, Pages: 515-520
9. Rajkumar Rajasekaran, Govinda k, Ashrith Reddy, Uday Sai Reddy, Yashwanth Reddy: *Visual Analysis of Temperature Time Series and Rainfall Using Big Data*. DOI:10.36872/LEPI/V50I3/201023
10. Rajasekaran Rajkumar, Jolly Masih, K.Govinda: *An analysis of mobile pass-codes in case of criminal investigations through social network data*. International Journal of Computers and Applications 09/2019;, DOI:10.1080/1206212X.2019.1662169
11. J. Patel, S. Shah, P. Thakkar, and K. Kotecha, “Expert Systems with Applications Predicting stock market index using fusion of machine learning techniques,” *Expert Syst. Appl.*, vol. 42, no. 4, pp. 2162–2172, 2015, doi: 10.1016/j.eswa.2014.10.031.
12. R. Choudhry and K. Garg, “A Hybrid Machine Learning System for Stock Market Forecasting,” no. July, 2014.
13. B. Qian and K. Rasheed, “Stock market prediction with multiple classifiers,” *Appl. Intell.*, vol. 26, no. 1, pp. 25–33, Feb. 2007, doi: 10.1007/s10489-006-0001-7.
14. M. Usmani, S. H. Adil, K. Raza, and S. S. A. Ali, “Stock market prediction using machine learning techniques,” *2016 3rd Int. Conf. Comput. Inf. Sci. ICCOINS 2016 - Proc.*, no. May 2018, pp. 322–327, 2016, doi: 10.1109/ICCOINS.2016.7783235.
15. A. Sharma, D. Bhuriya, and U. Singh, “Survey of stock market prediction using machine learning approach,” in *Proceedings of the International Conference on Electronics, Communication and Aerospace Technology, ICECA 2017*, 2017, doi: 10.1109/ICECA.2017.8212715.
16. Y. Wang and I. Choi, “Market Index and Stock Price Direction Prediction using Machine Learning Techniques: An empirical study on the KOSPI and HSI,” *arXiv Prepr. arXiv1309.7119*, 2013.
17. J. W. Lee, “Stock price prediction using reinforcement learning,” in *IEEE International Symposium on Industrial Electronics*, 2001, doi: 10.1109/isie.2001.931880.
18. J. Hale, “Scale , Standardize , or Normalize with Scikit-Learn What do These Terms Mean ? Why Scale , Standardize , or Normalize ?,” 2020.
19. “Statistics How To Why use Linear Relationships ? What is Simple Linear Regression ? How to Find a Linear Regression Equation : Overview,” 2020.