International Journal for Innovative Engineering and Management Research

A Peer Revieved Open Access International Journal

www.ijiemr.org

IJIEMR Transactions, online available on 06th Jan 2023. Link :http://www.ijiemr.org/downloads.php?vol=Volume-12&issue=ISSUE-1

DOI: 10.48047/IJIEMR/V12/ISSUE 01/35

Title **Prediction of Stock Market Using Trading Bot**

Volume 12, Issue 1, Pages: 399-411

Paper Authors

C.N. Sujatha, V. Padmavathi, G. Jahnavi Rao, M. Rakshitha Reddy , AVS Sai Sravani

International Journal for Innovative Engineering and Management Research
A Peer Revieved Open Access International Journal
www.ijiemr.org

Prediction of Stock Market Using Trading Bot

C.N. Sujatha[1], V. Padmavathi[2], G. Jahnavi Rao[3], M. Rakshitha Reddy [4], AVS Sai Sravani[5]

[1,3,4,5] Sreenidhi Institute of Science and Technology, Hyderabad, 501301,India
[2]Chaitanya Bharati Institute of Technology, Hyderabad, 500075, India
[1]cnsujatha@gmail.com
[2]chpadmareddy1@gmail.com
[3]jahnavi142000@gmail.com
[4]rakshithareddy999@gmail.com
[5]sravani.avss@gmail.com

Abstract
This paper proposes the best model suitable for trading suggestions using machine Learning. Machine Learning will help us to build an algorithm which will be able to buy, sell, hold and predict the movement of the stocks to give gain in asset value. Research has shown that financial news plays a major role to predict stock market movement of securities and financial instruments, so using sentiment analysis model and the machine learning together we are implementing this bot. Our approach is to apply supervised machine learning algorithms such as decision trees, random forests, regression analysis and ensemble learning. To see which of this model will be suitable for us to use in predicting the future movements of that stock. The main objective is to compare which model is very accurate at prediction and is capable of learning from the historical stock data which identifies trading opportunities in patterns seen on charts and in Price trends.

Keywords: LSTM, Random Forest Regressor, RNN model structure, Twitter Volume, XG Boost.

Introduction

The idea of stock market movement has forever been equivocal for investors in view of different compelling elements. This study intends to altogether diminish the gamble of pattern forecast with machine learning. Stock market is a medium of exchange where people buy, sell and issue equity shares of public corporations. The essence of stock market is that it is regulated by the government. For instance, in India, it is regulated by Securities and Exchange Board of India (SEBI). In stock market companies go public to get a good valuation, to raise funds so that it benefits the company to build up faster and expand. The price of the stock depends on the demand of the stock and at how much people are ready to buy the stock and the quantity. In short time cost and volume define the movements in the market for short term trades and will play a major factor in the bot. This trading algorithm endeavors use of speed as well as computational power of computers to be productive which only one of thousand investor tries to achieve. This bot improves the chances through better testing, learning, execution and technique configuration. This trading bot help to simplify the work of the trader and also make quick buck with a minimal effort. Such bots are very important for survival in the future financial market. Reports say that this trading bot market size will grow exponentially in the coming days. The goal of this paper is to further the revolution in the stock market of the coming tomorrow by providing an effective and efficient model to overcome most of the drawbacks faced due to manual trades.

For instance, trades can be executed at best price as possible, possibility to execute trades at ideal level, trades will be coordinated efficiently, effectively and immediately to keep up from high volume changes, reduce the exchange cost, simultaneously check with other market scenarios, reduce hazards of manual trading mistakes, it can also be back tested by the live and historical data to check if it is right time to trade or not, reduce the chances of psychological and emotional trades done by fear and greed by human traders. The objective of the work is to provide a trading bot which can give an accurate estimation of the future of any stock by processing the historical data. To develop a system which is accurate needs a lot of training and testing of the historical data and use Sentiment analysis referencing Twitter tweets about a particular stock and predict the movement of that stock.

Literature Survey

Mittal, Anshul, and Arpit Goel employed sentiment analysis and machine learning principles to study the link between public sentiment and market sentiment. They estimate stock market movements by using Twitter data to forecast popular sentiment, then combining that forecast with previous days' DJIA values [1].

Nguyen, Thien Hai, Kiyoaki Shirai, and Julien Velcin developed a model that uses social media sentiment to forecast stock price movement. Unlike most conventional methods, which take into consideration the overall mood or sentiment, the authors have constructed the model to take into account the sentiments of the company's specific themes, which are then put into the stock prediction model [2]. Pagolu, Venkata Sasank, Kamal Nayan Reddy, Ganapati Panda, and Babita Majhi used two alternative textual representations, Word2vec and N-gram, to analyze public mood in tweets. They analyzed the link between stock market movements of a firm and sentiments in tweets using sentiment analysis and supervised machine learning concepts on tweets retrieved from Twitter. They have applied sentiment analysis and supervised machine learning principles to the tweets extracted from Twitter analyzed the correlation between stock market movements of a company and sentiments in tweets [3].

Graham Kendall [4] first accompanied utilization of development system in machine learning in the year 2001, examining the improvement of a methodology that involved transformative procedure as a predictive tool. The methodology was easy to execute yet created results that contrasted well and the brain network forecasts. Hans-Georg Beyer [8] presented Evolution Strategies beginning from a study of history making sense of why the procedures are understood the way they are. The hypothetical issues, the fundamental ES algorithms and design principles along with future branches of ES research was introduced. Yong Hu in 2015 gave applications on utilizations of evolutionary methods and explored chosen papers based on purchase and hold procedure to analyze their models. Yong Hu uncovered the holes in ES strategies in stock exchanging [10].

Luke Rose [9] in 2018 introduced near examination on the new Machine Learning approaches utilized in anticipating the course and costs of chosen stocks for a specific time frame range, taking into account short, medium, and long-term investments. Lv, Dongdong [6] in 2019 made a detailed examination on the stock day trading strategy artificially assessing different machine learning algorithms and noticing the day trading performance of stocks under exchange cost and no exchange cost. This data set gives an enormous pool of data to the bot which will at last explore it through the stock exchanging process.

As of late, Nikhil Yadav [5] made sense of how twitter is viewed as a gigantic storehouse of general opinions and what it means for promoting procedures and

International Journal for Innovative Engineering and Management Research
A Peer Revieved Open Access International Journal
www.ijiemr.org

utilizing wistful assessment. Various methods utilized for characterizing an item were additionally stressed. Lufuno Ronald Marawala in 2010 figured about machine learning application in Stock Market and Trading. The primary thought process of this work was to utilize machine learning methods to demonstrate and predict the future cost of the securities exchange file. The outcomes show that the positioning of exhibitions support vector machines, neuro fuzzy frameworks, and neural networks is reliant upon the accuracy measure used [7].

Proposed Methodology

We are going to use machine learning models in order to get the accurate prediction of the stocks. Long Short-Term Memory Model (LSTM), Random Forest Regressor Model, XG Boost Model are the three models which we are going to use as shown in figure 1. The information in this paper comprises of the initial costs and shutting costs of the stocks in the New York Stock Trade NYSE (TSLA) removed from finance or yahoo finance, for TSLA our information covers the time of four years going from 1/02/2018 to 1/29/2022. To assemble the LSTM model we will utilize the LSTM RNN, this model purposes 80% of information for training and the other 20% of information for testing. For preparing we utilize mean squared mistake to streamline the model.

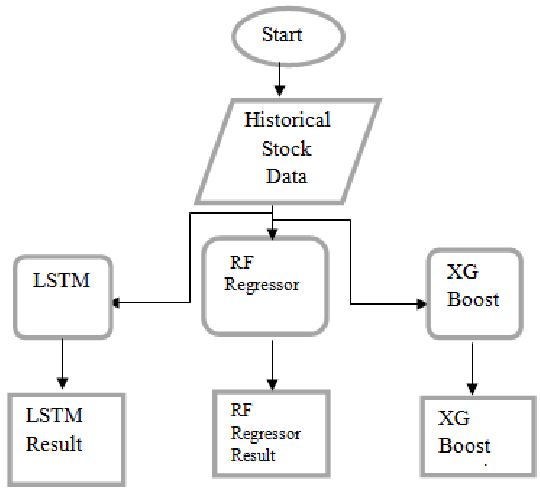


Fig.1. Architecture of the system

Dataset Generation

For our requirements, we made use of the dataset provided by Kaggle and yfinance. The dataset consists of tweet volume, polarity of the tweets, images and the dataset from yfinance consists of historical stock data. The datasets had to be merged according to the stock data available date and eliminate the rest of the excess data from the Kaggle dataset. The first issue with the twitter data was that it was not well sorted and had to be well sorted for use so that the accuracy doesn't get corrupted due to unwanted training and testing. In order to deal with this issue, excel sheets special features helped us to eliminate all unwanted data. Now, we had absolutely perfect data for processing. Now the dataset consists of date, open, high, low, close, adj close, volume, polarity, twitter volume.

Long Short Term Memory

LSTM is a specific type of RNN with a broad scope of purposes like document classification, time series analysis, voice and speech recognition. Opposite to feed forward networks, the forecasts (made by RNNs) are reliant upon earlier assessments. In exploratory works, RNNs are not applied comprehensively due to include of few lacks that result in impractical estimations. Without examination of a lot of details, LSTM tackles the issues by employing assigned gates for forgetting old information and learning new ones. LSTM layer is made off our neural network layers that collaborate in a particular technique. Long transient memory (LSTM) model builds the memory of the recurrent neural networks (RNN).

The RNN holds the momentary memory in which they permit the early deciding of the data that must be utilized in the ongoing neural network. For the quick assignments, the information decided early is utilized. They may not group all the prior resolved data for the neural networks. In repetitive, the LSTMS are the most broadly utilized in the brain organizations neural networks.

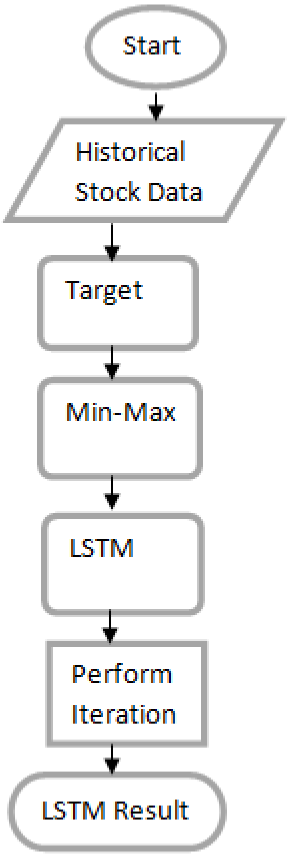


Fig. 2. LSTM Model

Figure 2 shows the LSTM model where a data frame should be created with the attributes adj close, ts_polarity, twitter_volume from the dataset. Create input feature vectors 'x' and target 'y'. Create a function to collect the column numbers for the features 'x' and the target 'y'. Divide the dataset into two segments- training and testing. For training 80% of the data is utilized and the remaining 20% data is used for testing. Scale both training data and test data using Min-Max Scaler. Import the Keras Modules in the LSMT RNN model structure with the help of

Sequential model, the LSTM layer and a Dropout layer. Define the LSTM RNN model structure and compile it using adam optimizer while taking mean square error as loss function. Train the model. Evaluate the model using testing data. Recover the original prices for the scaled version. Create a Data frame of real and predicted values.

The adequacy is utilized in the numerous groupings displaying of issues in utilizations of numerous areas like time-series, Natural language Processing, Video and geospatial. In any case, the main pressing concern with the RNN is the disappearing slope issue, it arises in light of the rehashed utilization of similar boundaries and once more, in the RNN block, at each step performed. To defeat this issue each various boundaries should be utilized at each time step.

Random Forest Regressor

The Random Forest is and quote ensemble learning and quote strategy comprising of the total of countless choice trees that brings about a decrease of fluctuation contrasted with the single choice trees. Random Forests are a collection of tree indicators with the goal of each tree relying on the advantages of an arbitrary vector that is analyzed independently also, with similar appropriation for all trees in the forest. Random Forest created by Leo Breiman is a gathering of un-pruned characterization or relapse trees produced using the irregular determination of tests of the preparation information. Irregular highlights are chosen in the enlistment cycle. Forecast is made by conglomerating the expectations of the gathering. The hypothesis botch consolidates quite far as the number of trees in the backwoods ends up being tremendous. The hypothesis bumble of a forest of tree classifiers depends upon the strength of the particular trees in the forest and the connection between them. Inner assessments screen mistake, strength, and relationship and these are used to show the response for growing the number of components used in splitting. Using an irregular assurance of features to part all center point yields mistake rates that differentiates well with XGBoost, and all are more remarkable as to commotion. Inside examinations are also used to quantify variable importance. These considerations are excessively proper to regression.

Figure 3 shows the RF Regressor model where a data frame should be created with the attributes adj close, ts_polarity, twitter volume from the dataset. Create input

International Journal for Innovative Engineering and Management Research
A Peer Revieved Open Access International Journal
www.ijiemr.org

feature vectors 'x' and target 'y'. Create a function to collect the column numbers for the features 'x' and the target 'y'.Divide the dataset into two segments- training and testing. For training 80% of the data is utilized and the remaining 20% data is used for testing. Scale both training data and test data using Min-MaxScaler. Create the Random Forest regressor instance and fit the model. Evaluate the model using testing data. Recover the original prices for the scaled version. Create a Data frame of real and predicted values.

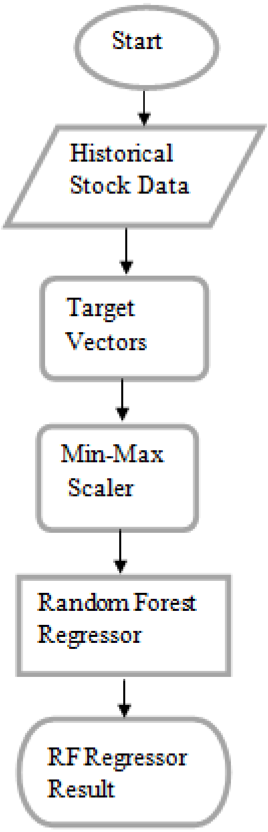


Fig. 3. RF Regressor Model

XG Boost

XGBoost is a circulated supporting tool kit that has been tuned for productivity, adaptability, and portability. It utilizes the Gradient Boosting structure to make machine algorithms. XGBoost is a parallel tree boosting algorithm (otherwise called GBDT, GBM) that settles different information science issues rapidly and precisely. A similar algorithm might handle issues with billions of models in a distributed environment (Hadoop, SGE, MPI). Gradient Boosted decision trees are completed in XGBoost. Decision trees are developed continuously in this system. In XGBoost, weights are extremely critical. Independent variables are given weights, which are given to the decision tree, which predicts results. The weight of components that the tree anticipated mistakenly is increased, and these new variables are assigned to the next decision tree. These various classifiers are then joined to make an extra solid and definite model. It can be used to handle issues including regression, classification, ranking, and user-defined prediction.

Figure 4 shows the XG Boost model where a data frame should be created with the attributes adj close, ts_polarity, twitter_volume from the dataset. Create input feature vectors 'x' and target 'y'. Create a function to collect the column

International Journal for Innovative Engineering and Management Research
A Peer Revieved Open Access International Journal
www.ijiemr.org

numbers for the features 'x' and the target 'y'. Split the data into training and test data. Divide the dataset into two segments- training and testing. For training 80% of the data is utilized and the remaining 20% data is used for testing. Scale both training data and test data using Min-Max Scaler. Create the XG Boost regressor instance and fit the model. Evaluate the model using testing data. Recover the original prices for the scaled version. Create a Data frame of real and predicted values.

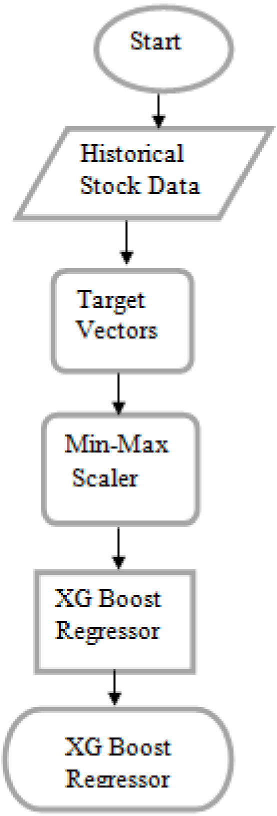
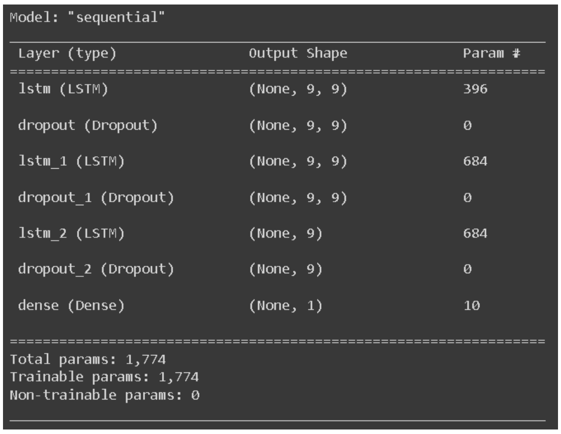


Fig. 4. XG Boost Model

Results and Discussions

All the three algorithms have been run on a platform called google colab. For these algorithms we are using TESLA dataset. This dataset has four years of data going from 01/02/2018 to 01/29/2022. TESLA dataset can be generated from a platform called yfinance or yahoo finance. In colab, after giving the code upload the TESLA dataset file to the code and run it. Below are the outputs that are generated. Figure 5 shows the summary of the LSTM model.



Model: "sequential"

Layer (type)	Output Shape	Param #
lstm (LSTM)	(None, 9, 9)	396
dropout (Dropout)	(None, 9, 9)	0
lstm_1 (LSTM)	(None, 9, 9)	684
dropout_1 (Dropout)	(None, 9, 9)	0
lstm_2 (LSTM)	(None, 9)	684
dropout_2 (Dropout)	(None, 9)	0
dense (Dense)	(None, 1)	10

Total params: 1,774
Trainable params: 1,774
Non-trainable params: 0

Fig. 5. Summary of LSTM model

Fig 6 shows the real and predicted prices of the stocks of LSTM model at a particular date and time. Blue path indicates the real prices and the orange path indicatesthe predicted prices. The graph shows that the predicted prices are a bit similarto the real prices of the stocks. This depicts theoretically calculated mean

International Journal for Innovative Engineering and Management Research
A Peer Revieved Open Access International Journal
www.ijiemr.org

square error of LSTM model as 1056.26487. This model got the highest mean square error when compared to the other two models.

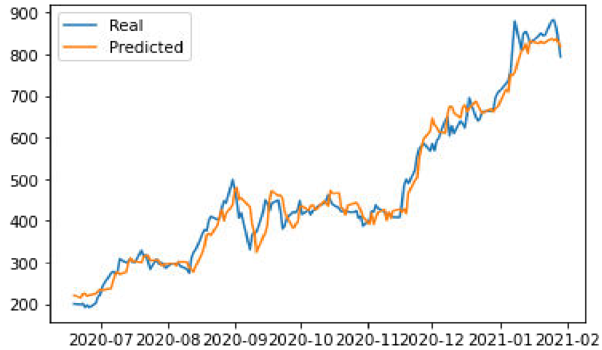


Fig. 6. Real vs Predicted values of LSTM Model

The Root Mean Squared Error and R-squared (coefficient of determination) of the testing target:

```
Root Mean Squared Error: 0.04668449702252395
R-squared : 0.9676307311088421
```

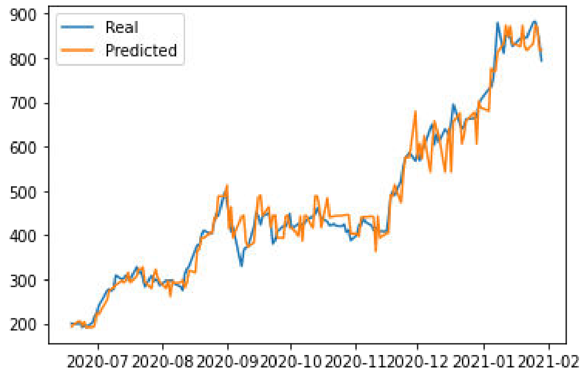


Fig. 7. Real vs Predicted values of RF Regressor Model

Random Forest Regressor

Fitting the model

```
RandomForestRegressor(bootstrap=False, max_depth=1000, n_estima
```

The Root Mean Squared Error and R-squared (coefficient of determination) of the testing target.

```
Root Mean Squared Error: 0.047023950888567916
R-squared : 0.9671582907115028
```

Figure 7 represents the real and predicted prices of the stocks of Random Forest Regressor model at a particular date and time. Blue path indicates the real pricesand the orange path indicates the predicted prices. The graph shows that the predicted prices are almost similar to the real prices of the stocks. This depicts theoretically calculated mean square error of Random Forest Regressor Model as 1041.07004534.

XG Boost

Fitting the model:

```
XGBRFRegressor(n_estimators=1000, objective='reg:squarederror')
```

Squared Error and R-squared (coefficient of determination) of the testing target:

```
Root Mean Squared Error: 0.041744171382483916
R-squared : 0.974119109715336
```

Figure 8 shows the real and predicted prices of the stocks of XG Boost model at a particular date and time. Blue path indicates the real prices and the orange path indicates the predicted prices. The graph shows that the predicted prices are almost similar to the real prices of the

International Journal for Innovative Engineering and Management Research
A Peer Revieved Open Access International Journal
www.ijiemr.org

stocks. This depicts theoretically calculated mean squareerror of XG Boost model as 832.388828.

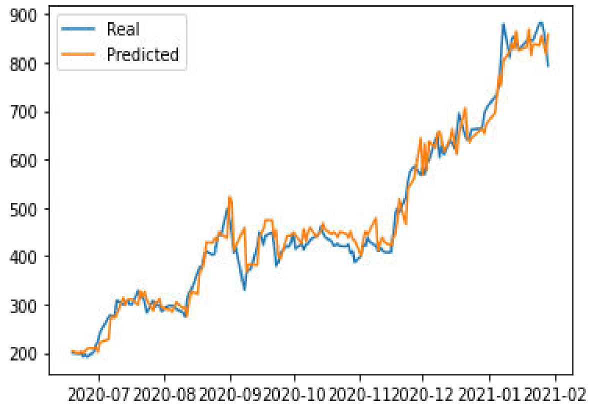


Fig. 8. Real vs Predicted values of XG Boost Model

LSTM Model (Reliance Dataset)

The above models using Tesla data set which has information that covers the time of four years going from 1/02/2018 to 1/29/2022. LSTM has got the highest mean square error. So, now we are going to use another dataset just to check whether we can get low mean square error for LSTM or not.

The dataset we are going to use is reliance data set which has information that covers a time of 25 years going from 01/01/1996 to 02/09/2021.This data set is generated from a medium called yfinance and Kaggle. In this model we are going to predict the stock prices for the next 30 days also. Figure 9 shows the total summary of LSTM model using reliance dataset.

```
Model: "sequential_1"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 lstm_3 (LSTM)               (None, 100, 50)           10400

 lstm_4 (LSTM)               (None, 100, 50)           20200

 lstm_5 (LSTM)               (None, 50)                20200

 dense_1 (Dense)             (None, 1)                 51

=================================================================
Total params: 50,851
Trainable params: 50,851
Non-trainable params: 0
_________________________________________________________________
```

Fig. 9. Summary of LSTM Model

Figure 10 shows the graph of stock prices for the next 30 days. The stock prices of the next 30 days are being predicted for LSTM model using reliance dataset.

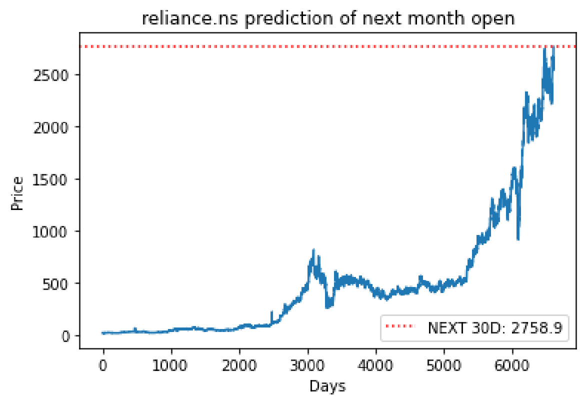


Fig. 10. Next 30 days Prediction

Squared Error and R-squared (coefficient of determination) of the testing target.

```
Root Mean Squared Error:
0.020765191669240905
R-squared:0.9900825893527314
```

International Journal for Innovative Engineering and Management Research
A Peer Revieved Open Access International Journal
www.ijiemr.org

This depicts theoretically calculated mean square error of LSTM model as 0.000431193

Figure 11 represents the real and predicted prices of the stocks of LSTM model at a particular date and time. Blue path indicates the real prices and the orange path indicates the predicted prices. The graph shows that the predicted prices are almost similar to the real prices of the stocks. This LSTM model has got the least error when compared to the other models using TESLA dataset.

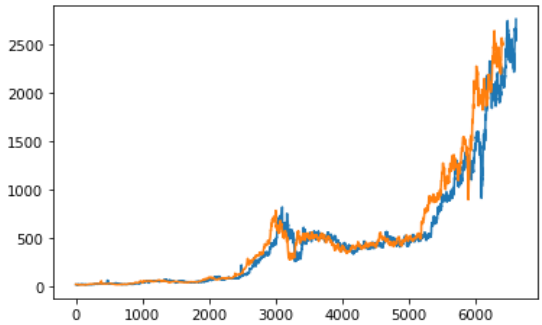


Fig. 11. Real and predicted values

Table 1 shows the comparison of mean square errors for the three algorithms using Tesla dataset. This shows that LSTM model has got the highest mean square error, then comes the Random Forest Regressor model and the least error is for XG Boost model.

Table 2 shows the comparison of mean square errors for LSTM model using Tesla dataset and Reliance dataset. This shows that the mean square error for LSTM model using Reliance dataset is the least when compared to Tesla. It also has the least mean square error when compared to the other models.

Table 1. Comparison of algorithms

S. No.	Model	Mean Square
1	LSTM	1056.26
2	Random forest Regressor	1041.07
3	XG Boost	832.38

Table 2. Comparison for Mean Square Error for LSTM

S. No.	Model	Data Set	Mean Square Error
1	LSTM	Reliance	0.00043
2	LSTM	Tesla	1056.26

Conclusion

The objective of this paper is to predict the movement of the stock market by using machine learning. Trading Bot not just gives Security, Cost, and Speed but on the other hand is a progressive innovation for the future monetary business sectors and economy. Trading Bot makes it more straightforward for both new merchants as well as laid out ones in getting productive results with limited exertion, time and loss. The mix of Financial Knowledge with Machine Learning is an interest of future Trading and improves both Performance and Revenue. In this paper we conclude that XG Boost Regressor is more efficient than Long Short-Term Memory and Random Forest Regressor as per the results shown above. LSTM model using reliance dataset has the least mean square error when compared to Tesla. Italso has the least mean square error when compared to the other models.

References

Mittal, Anshul, and Arpit Goel. "Stock prediction using twitter sentiment analysis." Standford University, CS229(2011http://cs229.Stanford.edu/proj2011/GoelMittalStockMarketPredictionUsingTwitterSentimentAnalysis. pdf) 15 (2012): 2352.

Nguyen, Thien Hai, Kiyoaki Shirai, and Julien Velcin, Sentiment analysis on social media for stock movement prediction, Expert Systems with Applications 42, no. 24 (2015): 9603-9611.

Pagolu, Venkata Sasank, Kamal Nayan Reddy, Ganapati Panda, and Babita Majhi. "Sentiment analysis of Twitter data for predicting stock market movements." In 2016 international conference on signal processing, communication, power and embedded system, pp. 1345-1350. IEEE, 2016.

Kendall, Graham, Jane M. Binner, and Alicia M. Gazely. Evolutionary strategies: A new macro economic policy tool IFAC Proceedings Volumes 34,20 (2001): 377381.

Yadav, Nikhil, et al. "Twitter Sentiment Analysis Using Machine Learning for Product Evaluation, 2020 International Conference on Inventive Computation Technologies (ICICT). IEEE,

Lv, Dongdong, et al. An empirical study of machine learning algorithms for stock daily trading strategy, Mathematical Problems in Engineering, 2019 (2019).

Bali, Ashish and Madan, Archit and Upadhyay, Aayush and Sah, Piyush and Nehra, Vibha, Development of Trading Bot for Stock Prediction Using Evolution Strategy, EasyChair, 2021.

Beyer, Hans-Georg and Schwefel, Hans-Paul, Evolution strategies--a comprehensive introduction, Natural computing, 1,1,3-52, Springer, 2002.

Rose, Luke and Gartner, Thomas, Automated Stock Market Trading using Machine Learning, IEEE, 2018.

Hu, Yong and Liu, Kang and Zhang, Xiangzhou and Su, Lijun and

Ngai, EWT and Liu, Mei, Application of evolutionary computation for rule discovery in stock algorithmic trading: A literature review, Applied Soft Computing, 36, 534-551, Elsevier, 2015.