Stock Market Trend Prediction Based on LSTM and XGBoot

Jianqiao Liu, Wen-Chih Li
Computer Science – Graduate School
Washington State University
Pullman, WA, USA
jianqiao.liu@wsu.edu, wen-chih.li@wsu.edu

***Abstract*—In the stock market, trading is high frequency. It is hard to detect or predict the trend of the future stock market. Plenty of researchers and investors have done plenty of studies on stock market prediction, put forward many theories, and hypotheses try to find a relatively accurate predicting method. Therefore, we want to do some investment and try to find an appropriate method to predict the stock market. First, we will calculate the moving average of each company. Use this value to predict the future stock value. Then, we start to predict its value, we will have to take machine learning or deep learning methods into use. Nowadays, there are two popular methods which are the LSTM model and the XGBoost model for predicting stock value. In this paper, we are going to find a low-risk company, then, we are trying to find out which method has a high mimic to the future price of this company.**

Keywords—LSTM, XGBoost, Moving Average, Prediction.

I. Introduction

Stock market trend prediction is always one of the hotspots of research. However, the stock market provides both opportunities and risks for investors to make profits.

Stock price prediction refers to the prediction of trading operations at a certain time in the future. It is based on historical data sets and real data of the stock market, in accordance with a certain predictive model. The forecast plays an important and positive role in improving the efficiency of the trading market and giving full play to market signals. Accurate stock price forecasts can help investors adjust trading strategies in a timely manner, effectively avoid investment risks, and obtain higher returns.

Price predictions have long appeared in various trading markets. However, affected by many factors such as internal changes of the stock market and sudden impact of the external market, the prediction results of some existing stock price prediction models are not perfect.

In this paper, we grabbed the daily stock data of Apple, Google, Microsoft, and Amazon from Yahoo Finance for the past ten years. Then, we did data wrangling and calculated the moving average. We also analyzed the moving average of these four companies. Using the moving average of a stock price is a naïve approach to predict the stock trend. Furthermore, we used LSTM and XGBoost model to generate the predicted stock value and did some comparisons between the prediction results of these two models. At end, we found that comparing to XGBoost model, LSTM model has higher prediction accuracy. In conclusion, we need to further improve the algorithms and models, use the given historical data to extract valuable data information, and achieve more accurate stock price forecasts.

II. Problem Definition

With the advancement of technology, more and more open questions can be answered or achieved by giving the right methods. For example, for finance and computer science, predicting stock movements is an extremely difficult problem. Plenty of researchers and investors delve into this topic to find out a way to predict stock trading. If someone can come up with a method that can approximately predict the stock value, they will be able to become the next billionaire. In fact, the most popular method is to use machine learning, especially deep learning [1].

Now, with the dominance of electronic stock trading, it is possible to find and make a profit from the price difference in real-time. Machine learning has been applied in stock trading for years by companies. However, with the rising of deep learning, price forecasting models become more accurate, which creates more opportunities to gain higher profits. We consider using two methods, LSTM and XGBoost. These two methods are highly accurate or reasonable for us to predict the future stock value of each company.

III. Models & Algorithms

In this section, we will mainly discuss our methods, models, and algorithms. First, we will use data wrangling to organize our data to new data frames in order to do further research. The moving average of each company could also give us the trend of each company's stock price. Therefore, we visualize the results we mentioned above. After finding the moving average of each company, we will apply the Long Short-Term Memory (LSTM) model and the eXtreme Gradient Boosting (XGBoost) model for future price predicting.

A. Data Wrangling

When we first gain the historical data of each company from Yahoo finance, it is a messy dataset. So, we need to organize this dataset in order to easily call it in the future. Since globals() is a sloppy way of setting the DataFrame names, but it's simple. We also add a new column into our data frame which is the company's name. It would be convenient for us to call it when we are going to use it later on. For instance, if we want to use APPLE's historical data, we only need to call 'AAPL'.

B. Moving Average

In statistics, a moving average is a calculation used to analyze data points by creating a series of averages of different subsets of the full data set. In finance, a moving average (MA) is a stock indicator that is commonly used in technical analysis. The reason for calculating the moving average of a stock is to help smooth out the price data by creating a constantly updated average price [2].

By calculating the moving average, the impacts of random, short-term fluctuations on the price of a stock over a specified time frame are mitigated [2]. Moreover, we could use these results to predict the future stock trend.

We use moving average to predict the future stock trend. In this way, the method is too naïve. Also, it has the drawback that even if these two companies have the same trend before, it would not always be the same in the future. We need to calculate the actual stock value of each day for each company.

C. Long Short-Term Memory (LSTM)

The LSTM model is a special recurrent neural network (RNN), which was developed to solve the problem of vanishing gradient in traditional RNN training. Unlike standard feedforward neural networks, LSTM has feedback connections that can learn long-term dependencies [5]. LSTM has three gates, namely input gate, forget gate and output gate. The update gate adds 19 pieces of information to the unit status. Forget the date to determine and delete information that is no longer needed by the model. The output gate determines the amount of information output to the next layer as an activation.

LSTM was introduced by Hochreiter & Schmidhuber (1997). By introducing Constant Error Carousel (CEC) units, LSTM deals with the vanishing gradient problem. The initial version of the LSTM block included cells, input, and output gates [6]. Compared with standard RNN, which has a simple structure with a single tank layer(figure 1).

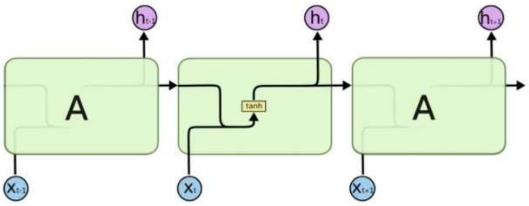


Fig. 1. Standard RNN repeating module with a single layer [7].

LSTM also has this chain structure, but the repeating module has a different structure. There is not only one neural network layer, but four, interacting in a very special way.

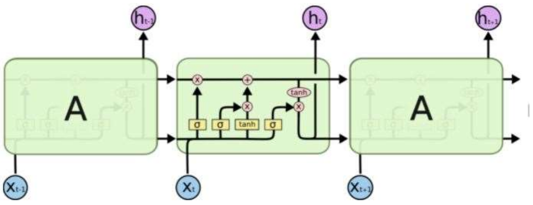
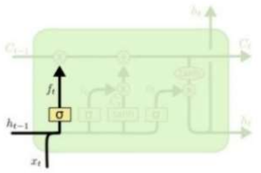


Fig. 2. The repeating module in an LSTM contains four interacting layers.

Let's walkthrough the LSTM model step by step. The first step in our LSTM is to decide what information we are going to throw away from the cell state. This decision was made by a sigmoid layer called the "forget gate layer." It looks at h_{t-1} and x_t, and outputs a number between 0 and 1 for each number in the cell state C_{t-1}. The 1 represents "completely keep this" while the 0 represents "completely get rid of this." [7]

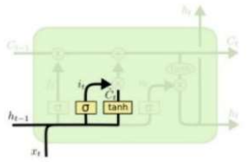
Let's go back to our language model example, which tries to predict the next word based on all the previous words. In such questions, the cell state may include the gender of the current subject so that the correct pronouns can be used. When we see a new subject, we want to forget the gender of the old subject.



$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] + b_f\right)$$

Fig. 3.

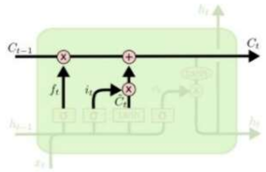
The second step is to decide what the new information we are going to get and store in the cell state.



$$i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] + b_i\right)$$
$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C)$$

Fig. 4. Second step

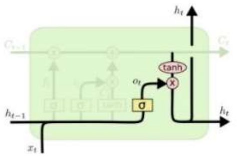
Update the old cell state into the new cell state.



$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$

Fig. 5. Upate step

Final step is that we are going to decide what we want to output.



$$o_t = \sigma\left(W_o \left[h_{t-1}, x_t\right] + b_o\right)$$
$$h_t = o_t * \tanh\left(C_t\right)$$

Fig. 6. Final step

D. eXtreme Gradient Boosting (XGBoost)

XGBoost uses a new regularization method on the traditional gradient booster (GBM) to significantly reduce complexity. In order to measure the performance of the model given a specific data set, XGBoost defines an objective function that considers the training loss $L(\theta)$ and regularized $\Omega(\theta)$ terms, where the latter penalizes the complexity of the model and prevents overfitting, Θ refers to the parameters that will be discovered during training [8] (Equation 7).

The model $\hat{y}^{(t-1)}$ obtained in the t-th round of training is a combination of k trees, that is, an addition strategy is applied during training, and a new tree that optimizes the system $f_t(x)$ is added to the model generated in the previous round at a time $\hat{y}^{(t-1)}$, where x is the input (Equation 8).

In order to determine the complexity of the tree $\Omega(f)$ [9], proposed a method to define it as an equation 9. The first term γT evaluates the number of leaves T, taking γ as a constant, and the second term calculates the L2 norm of the leaf score w_j.

In the equation 10 and the equation. 11. g_i and h_i are the first and second order of partial derivatives after Taylor expansion of the selected loss function, $I_j = \{i|q(x_i) = j\}$ is the index of the data point belonging to the jth leaf Group, $q(x)$ is the tree.

Finally, in the objective function, take the minimum and minimum independent variables of the quadratic function of a single variable w_j, treat $q(x)$ as a fixed value, λ is a small constant value, and the result is an equation 12 and the equation 13. The latter evaluates the quality of the tree structure, that is, the smaller the score, the better [9].

$$obj(\theta) = L(\theta) + \Omega(\theta) \tag{7}$$

$$\hat{y}^{(t)} = \sum_{k=1}^{t} f_k(x) = \hat{y}^{(t-1)} + f_t(x) \tag{8}$$

$$\Omega(f) = \gamma T + 1/2\lambda \sum_{j=1}^{T} w_j^2 \tag{9}$$

$$G_j = \sum_{i \in i_j} g_i \tag{10}$$

$$H_j = \sum_{i \in i_j} h_i \tag{11}$$

$$W_j^* = -\frac{G_j}{H_j+\lambda} \tag{12}$$

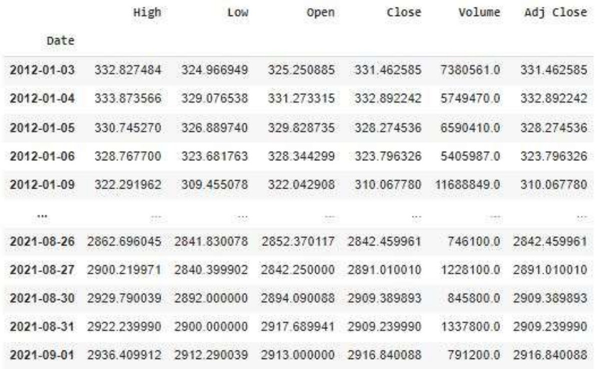
$$obj^* = -\frac{1}{2}\sum_{j=1}^{T}\frac{G_j^2}{H_j+\lambda} + \gamma T \tag{13}$$

IV. Implementation & Analysis

In this section, we will be present our implementation at Google Colab and VS code based on the methods mentioned in section III. There is more coding part could view through this link: https://github.com/Jianqiao-WSU/ML-Project.

A. Data

We use Pandas DataReader to grab the daily stock data of Apple, Google, Microsoft, and Amazon from Yahoo Finance for the past ten years.



	High	Low	Open	Close	Volume	Adj Close
Date						
2012-01-03	332.827484	324.966949	325.250885	331.462585	7380561.0	331.462585
2012-01-04	333.873566	329.076538	331.273315	332.892242	5749470.0	332.892242
2012-01-05	330.745270	326.889740	329.828735	328.274536	6590410.0	328.274536
2012-01-06	328.767700	323.681763	328.344299	323.796326	5405987.0	323.796326
2012-01-09	322.291962	309.455078	322.042908	310.067780	11688849.0	310.067780
...	...	...	...	...	...	...
2021-08-26	2862.696045	2841.830078	2852.370117	2842.459961	746100.0	2842.459961
2021-08-27	2900.219971	2840.399902	2842.250000	2891.010010	1228100.0	2891.010010
2021-08-30	2929.790039	2892.000000	2894.090088	2909.389893	845800.0	2909.389893
2021-08-31	2922.239990	2900.000000	2917.689941	2909.239990	1337800.0	2909.239990
2021-09-01	2936.409912	2912.290039	2913.000000	2916.840088	791200.0	2916.840088

Fig. 7. Stock data format.

As shown in the figure, the data consists of high price, low price, open price, close price, volume and adj close price. We mainly use close price and volume to do the implementation.

B. Moving Average

We calculate the moving average of each company and then check if there is any similar trends among each other. If there is a correlation, we could assume that the future prices of these two stocks are close to each other.

C. Long Short-Term Memory (LSTM)

First, we use MinMaxScaler from Sklearn to scale our datasets.

Second, we create the training data set and create the scaled training data set.

Third, split the data into x_train and y_train data sets

Fourth, build the LSTM model, then, compile and train the model using the TensorFlow backend.

Finally, we plot our results. We use red line to denote the real value and orange line to denote the predicted value.

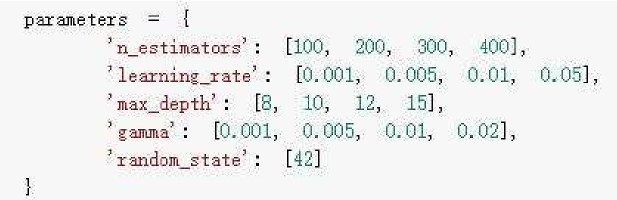
We use MinMaxScaler to scale our datasets to make training become faster. After implementing the LSTM model, we want to check whether the results have higher accuracy or the stock trend is similar to the reality.

D. eXtreme Gradient Boosting (XGBoost)

First, we split the stock data into three subsets: training data set (70% of total dataset), validation data set (15% of total dataset) and test data set (15% of total dataset). Then, we calculated split indices and create three separate frames (train_df, valid_df, test_df).

Second, drop all unnecessary columns and split them into features and labels.

Third, build the XGBoost model and pick the parameters. We will choose the best parameter from these testing parameters (which are shown in the following figure) for our model and calculate the best validation score.



```
parameters  =  {
        'n_estimators':  [100,  200,  300,  400],
        'learning_rate':  [0.001,  0.005,  0.01,  0.05],
        'max_depth':  [8,  10,  12,  15],
        'gamma':  [0.001,  0.005,  0.01,  0.02],
        'random_state':  [42]
}
```

Fig. 8. Testing parameters.

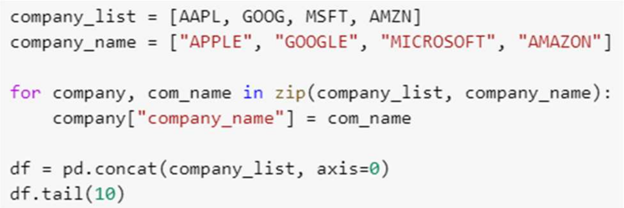
Finally, after finding the best parameters and best validation score, we start to plot the results. We use the blue line to denote real value, and the orange line to denote the predicted value.

We hypothesize that if the accuracy of the results of XGBoost model is higher than LSTM model, XGBoost model might be a better choice to predict the stock value.

V. Results & Discussion

In this section, we will present our results based on the implementation mentioned in the last section. The discussion or observation will go along with the results.

A. Data Wrangling

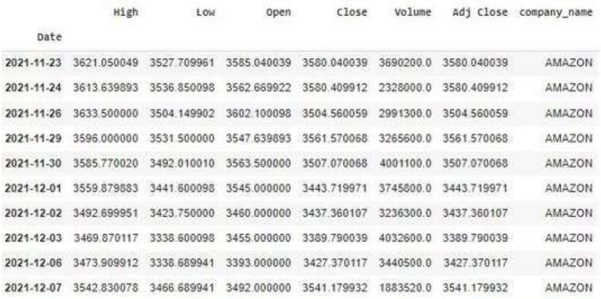


```
company_list = [AAPL, GOOG, MSFT, AMZN]
company_name = ["APPLE", "GOOGLE", "MICROSOFT", "AMAZON"]

for company, com_name in zip(company_list, company_name):
    company["company_name"] = com_name

df = pd.concat(company_list, axis=0)
df.tail(10)
```

Fig. 9. Data wrangling implementation.



	High	Low	Open	Close	Volume	Adj Close	company_name
Date							
2021-11-23	3621.050049	3527.709961	3585.040039	3580.040039	3690200.0	3580.040039	AMAZON
2021-11-24	3613.639893	3536.850098	3562.669922	3580.409912	2328000.0	3580.409912	AMAZON
2021-11-26	3633.500000	3504.149902	3602.100098	3504.560059	2991300.0	3504.560059	AMAZON
2021-11-29	3596.000000	3531.500000	3547.639893	3561.570068	3265600.0	3561.570068	AMAZON
2021-11-30	3585.770020	3492.010010	3563.500000	3507.070068	4001100.0	3507.070068	AMAZON
2021-12-01	3559.879883	3441.600098	3545.000000	3443.719971	3745800.0	3443.719971	AMAZON
2021-12-02	3492.699951	3423.750000	3460.000000	3437.360107	3236300.0	3437.360107	AMAZON
2021-12-03	3469.870117	3338.600098	3455.000000	3389.790039	4032600.0	3389.790039	AMAZON
2021-12-06	3473.909912	3338.689941	3393.000000	3427.370117	3440500.0	3427.370117	AMAZON
2021-12-07	3542.830078	3466.689941	3492.000000	3541.179932	1883520.0	3541.179932	AMAZON

Fig. 10. Data wrangling results.

As shown in the graphs above, we can see that we grabbed the data and added the company name into the data frame.

B. *Moving Average*

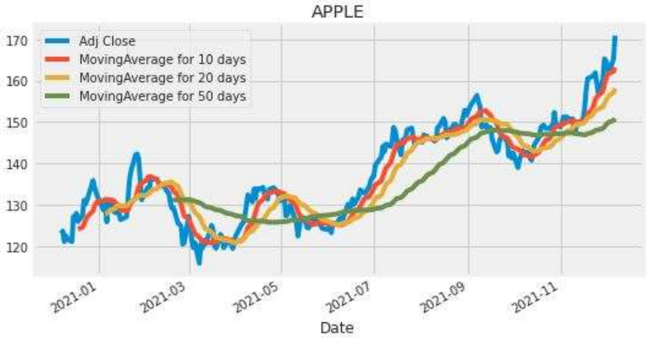


Fig. 11. Moving average of Apple.

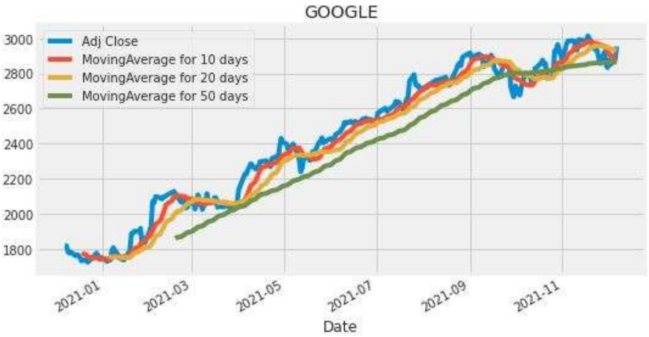


Fig. 12. Moving average of Google.

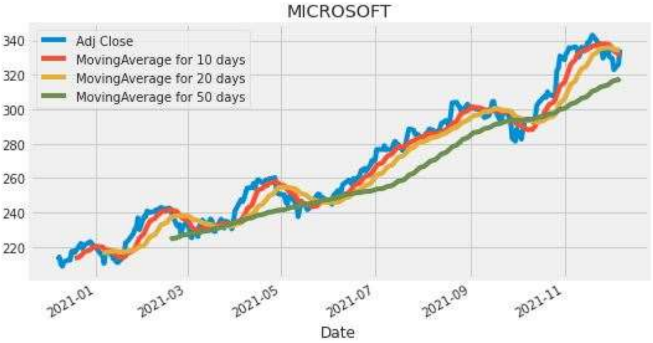


Fig. 13. Moving average of Microsoft.

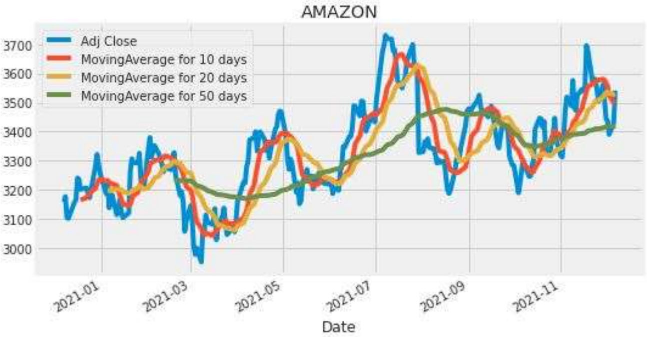


Fig. 14. Moving average of Amazon.

As the figures show, we can easily see that the moving average trends of Google and Microsoft are quite similar, which means they have a strong correlation. If we want to predict their future prices or assess their investment risks, we should bundle them and observe them together. On the other hand, the moving averages of Apple and Amazon are unstable, which tells us that they could have high risk, but we still need more evidence to prove this.

C. *Long short-term memory (LSTM)*

We use red line to denote the real value and orange line to denote the predicted value.

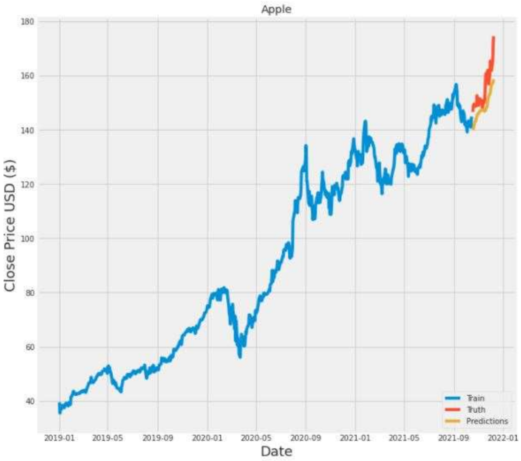


Fig. 15. Apple prediction based on LSTM model.

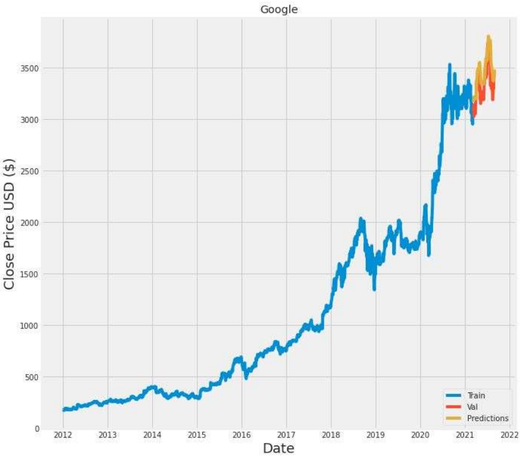


Fig. 16. Google prediction based on LSTM model.

The two figures show the comparison results of the real data and the predicted data generated by using the LSTM model to train the Apple and Google historical stock data. However, the accuracy of the two is slightly different.

Since we only use 3 years of Apple’s stock data for training, the accuracy was lower than we expected. In contrast, we use the Google’s stock data from 2019 to the present, so the prediction graph is ideal. Then, we plot other companies based on the conclusion we got above.

To conclude, for the LSTM model, the more training data we fed to the model, the more accuracy we will receive.

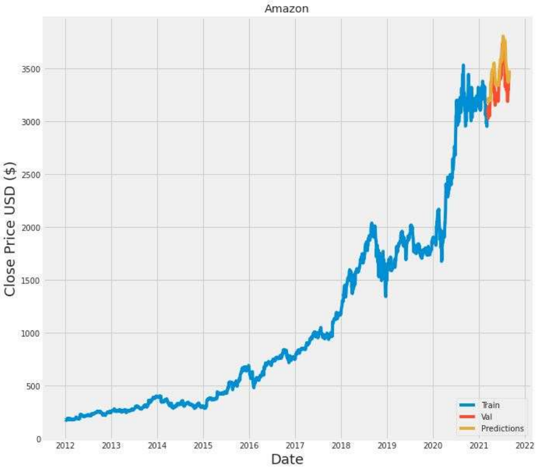


Fig. 17. Amazon prediction based on LSTM model.

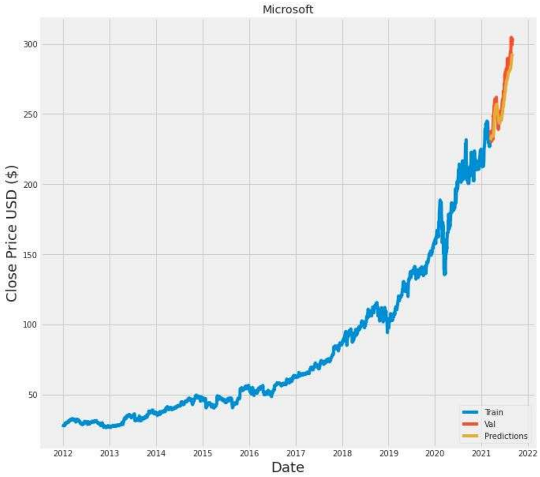


Fig. 18. Microsoft prediction based on LSTM model.

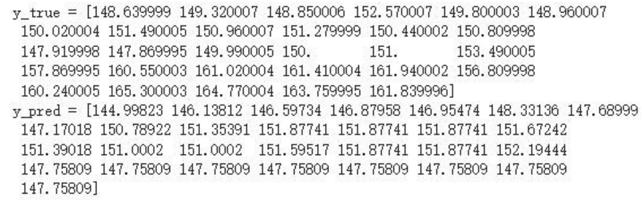
D. eXtreme Gradient Boosting (XGBoost)

After doing the doing the experiment, we get the following best parameters and best validation score:

TABLE I. BEST PARAMETERS AND BEST VALIDATION SCORE

	Gamma	*Learning Rate*	*Max Depth*	*N Estimators*	*Random State*
Best Paramater	0.001	0.05	10	100	42
Best validation score	-0.6227028102045647				

With these parameters, we predict the value of Apple price with XGBoost model and get the following results:



```
y_true = [148.639999 149.320007 148.850006 152.570007 149.800003 148.960007
 150.020004 151.490005 150.960007 151.279999 150.440002 150.809998
 147.919998 147.869995 149.990005 150.        151.        153.490005
 157.869995 160.550003 161.020004 161.410004 161.940002 156.809998
 160.240005 165.300003 164.770004 163.759995 161.839996]
y_pred = [144.99823 146.13812 146.59734 146.87958 146.95474 148.33136 147.68999
 147.17018 150.78922 151.35391 151.87741 151.87741 151.87741 151.67242
 151.39018 151.0002  151.0002  151.59517 151.87741 151.87741 152.19444
 147.75809 147.75809 147.75809 147.75809 147.75809 147.75809 147.75809
 147.75809]
```

Fig. 19. Predictions with XGBoost model.

Here is the comparison result of the predicted and the real value. The blue line denotes the real value, and the orange line denotes our predicted value. As the figure shown, we can see that it did not predict well.

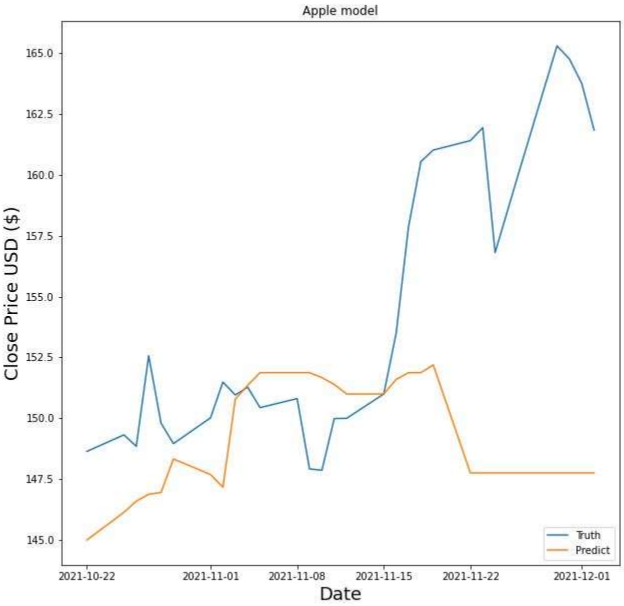


Fig. 20. The plot of predicted and real value.

Take a closer look at the plot, we can find out that it does have a similar trend at the beginning. However, when more data is provided, the predicted line becomes more stable and there is no similar trend compared to the line of real value. This is a problem that needs to be figured out.

Due to the way in which tree-based models divide the input space of any given problem, these algorithms are largely unable to infer target values that exceed the training data limit when making predictions. This is usually not a big problem in classification tasks, but it is definitely a limitation in regression tasks that involve predicting continuous output [10].

As the author mentioned above, the XGBoost model has limitations to predict continuous values. If the training data only contains values between 0 and 100. Tree-based regression models will have difficulty predicting values outside this range. Furthermore, the author gives an example of the S&P 500. If we look at the trend of a popular stock market index like the S&P 500 over the past 50 years, we will find that the price of the index has experienced highs and lows, but will eventually rise over time. In fact, according to historical data, the average annual rate of return of the S&P 500 Index is about 10%, which means that prices increase by about 10% every year on average. Just try to use XGBoost to predict the price of the S&P 500 Index, and you will find that it can predict price drops, but fail to capture the overall upward trend in the data. To be fair, predicting stock market prices is an extremely difficult problem, even machine learning cannot solve it, but the point is that XGBoost cannot predict price increases beyond the range of training data [10].

VI. Related work

According to the article "Forecast of LSTM-XGBoost in Stock Price Based on Bayesian Optimization" [11], it clearly shows that using the combination of the LSTM model and the XGBoost model has a huge improvement over using only one of them. Furthermore, this article mentions a surprising discovery that the author uses Bayesian calculation to optimize the parameters that proves to be difficult to find the best solution in the XGBoost model.

In this paper, a hybrid model (LSTM-BO-XGBoost) based on the correlation analysis of LSTM and XGBoost enhanced by Bayesian optimization was proposed to solve the challenge of stock price prediction. They compared with the single LSTM network model, RNN network model, and the LSTM-BO-XGBoost hybrid model. The results show that the LSTM-BO-XGBoost model has higher performance, stability, and feasibility than the other models [11].

VII. Conclusion

The development of intelligent models to predict the stock price could help investors to become the next billionaire. This paper presents two well-known machine learning methods, the LSTM and the XGBoost to predict the stock price. To validate the performance of both methods, we use a data set containing the historical data of each company from 2012 till now.

The analysis of results shows that in predicting the future price of stocks, the LSTM model has a higher accuracy than the XGBoost model. We can clearly see the XGBoost model as the drawback of not being able to infer target values beyond the training data limit when making predictions.

For the future work, we are going to find more appropriate algorithms to predict the future stock price. However, as we mentioned in the related work, the article "Forecast of LSTM-XGBoost in Stock Price Based on Bayesian Optimization" [11], clearly cited that combining the LSTM model and the XGBoost model into a new model named LSTM-BO-XGBoost would have higher performance than any other methods. But, in this century, everything changes extremely faster than anyone can imagine. It might have more methods or models that are suitable for stock price prediction that will come up soon. The only thing we can do is to stay closer to the technology and learn any invention that happen.

VIII. Acknowledgment

We are grateful to our professor Janardhan Rao Doppa for providing the guidelines. We would also like to thank Fares Sayah from the kaggle.com who gave us the inspiration for this project.

References

[1] W. Haotian, "Trading Decision Making Based on Hybrid Neural Network," 2021 6th International Conference on Intelligent Computing and Signal Processing (ICSP), 2021, pp. 1247-1250, doi: 10.1109/ICSP51882.2021.9408683.

[2] Fernando, J. (2021, December 7). Moving Average (MA). Investopedia.Com. https://www.investopedia.com/terms/m/movingaverage.asp.

[3] seaborn: statistical data visualization — seaborn 0.11.2 documentation. (n.d.). Pydata.Org. Retrieved December 7, 2021, from https://seaborn.pydata.org/

[4] Hayes, A. (2021, December 7). Correlation. Investopedia.Com. https://www.investopedia.com/terms/c/correlation.asp

[5] Chen, L. (2020). Stock Price Prediction using Adaptive Time Series Forecasting and Machine Learning Algorithms. UCLA. ProQuest ID: Chen_ucla_0031N_18851. Merritt ID: ark:/13030/m5zs84k9. Retrieved from https://escholarship.org/uc/item/0zp9s76c

[6] Wikipedia contributors. (2021, November 17). Long short-term memory. Wikipedia, The Free Encyclopedia. https://en.wikipedia.org/w/index.php?title=Long_short-term_memory&oldid=1055779877

[7] Understanding LSTM Networks. (n.d.). Github.Io. Retrieved November 20, 2021, from http://colah.github.io/posts/2015-08-Understanding-LSTMs/

[8] Cerna, Selene et al. A comparison of LSTM and XGBoost for predicting firemen interventions. Advances in Intelligent Systems and Computing, v. 1160 AISC, p. 424-434. Available at: <http://hdl.handle.net/11449/198964>.

[9] Chen, T., Guestrin, C.: XGBoost: A scalable tree boosting system. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. pp. 785–794. KDD '16, ACM, New York, NY, USA (2016). https://doi.org/10.1145/2939672.2939785

[10] A. Mavuduru, "Why XGBoost can't solve all your problems," Towards Data Science, 10-Nov-2020. [Online]. Available: https://towardsdatascience.com/why-xgboost-cant-solve-all-your-problems-b5003a62d12a. [Accessed: 09-Dec-2021].

[11] T. Liwei, F. Li, S. Yu and G. Yuankai, "Forecast of lstm-xgboost in stock price based on bayesian optimization," *Intelligent Automation & Soft Computing*, vol. 29, no.3, pp. 855–86