Project Report

Exploring Stock Price Prediction
with Past Data and Market Sentiment

CHI, Ting Hsuan(20819744), tchi@connect.ust.hk
Lin, Yen Po(20819782), ylindr@connect.ust.hk

May 5, 2024

1 Background Information

The stock market is known for being dynamic because multiple factors can affect the stock price. Factors like global economics, company's financial performance may reflect on the stock price. This also means that there's a lot of information and data we can find online for us to find patterns in. Therefore, the concept of algorithmic trading rises, which we use programs to automate the stock analysis and prediction. As a Computer Science student, we are too busy doing PAs. If we want to make some pocket money from the stock, we don't have the time to analyse the stock market by ourselves. So we want to explore how can we use AI model to do the stock analysis and prediction for us using past market data. Let the machine to help us discover the future trend of the stock market to make our life easier.

We found there are two ways to predict the stock price. One is using the past closing price, and another one is using the market sentiment derived from the tweets related to the company. In this project, we will explore how different time series model setting can affect the prediction output. And we will include the market sentiment into the past data as the input data and discuss on how the market sentiment can affect the training accuracy.

2 Related Application

Stock price prediction has been a popular topic in machine learning for a while, so there are plenty of existing applications related to stock analysis and stock market prediction. According to our research, there are multiple ways to implement a stock prediction model. In [1], the author compared multiple algorithms like supervised learning, unsupervised learning, ensemble algorithms, time series analysis and deep learning algorithms in stock predicting. We also saw some other ways to predict the stock price. Unlike us depend on the past stock closing price to train the model, they use news content to train the model to achieve price prediction with news. In this project we will work on both market sentiment and past data to poredict the stock price.

3 Hardware and Software Environment

This project is fully developed on **Google Colab Pro** with **T4 GPU** configuration. The used datasets are stored in the Google Drive and loaded during runtime.

4 Data Exploration and Preparation

In this project, we make use of two data sources: 1. A tweets dataset with tweets related to several companies. 2. The real market data from yfinance.

4.1 Data Overview

In the tweets dataset, there are total **64479** entries from **25** different companies. The tweets distribution is as shown in Figure: 1. In this project, we will use the top three companies with the most tweets in the dataset, which is Tesla(TSLA), Taiwan Semiconductor Manufacturing(TSM), and Apple(AAPL).

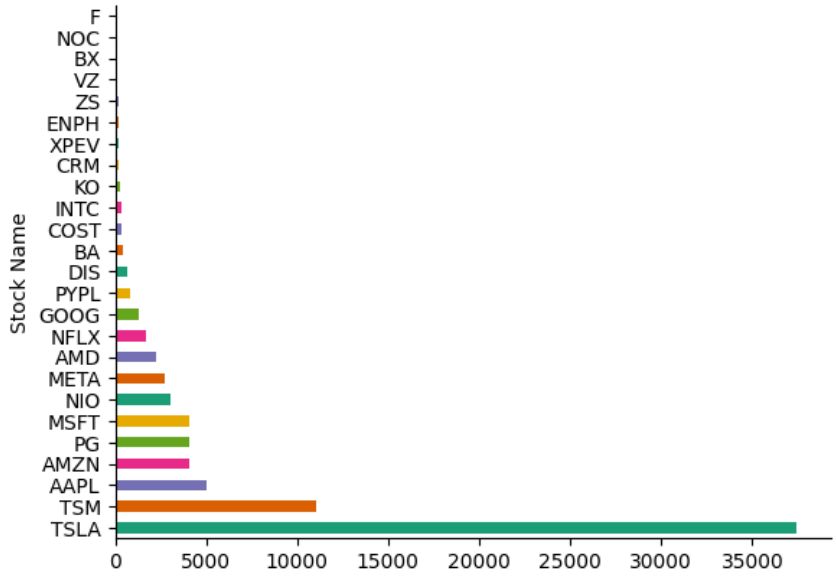


Figure 1: Distribution of the Tweets by company.

From yfinance, we download the data from 30 September 2021 to 30 September 2022 of the selected companies to match with the date range of our tweets dataset. In the downloaded data, we have the company's **Open price, Highest price, Lowest price, Closing price, Adjacent Closing price and Sold Volume** of each day. In this project, we will mainly focus on the closing price. The closing price of each company from 2021-9-30 to 2022-9-29 is shown in figure 2.

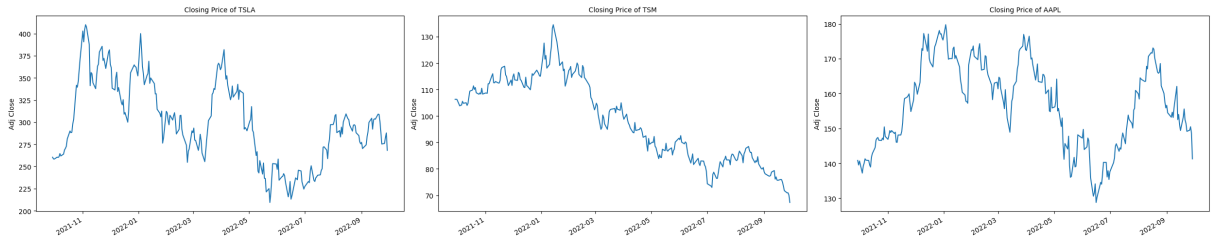


Figure 2: The closing price.

4.2 Plans for Data Preprocessing

We will perform sentiment analysis on the tweets to identify whether the tweet is positive or negative to the company's stock price. And to make use of the information extracted from yfinance, we will perform feature engineering to generate more technical indicators as input for the GAN model. Only closing prices of the yfinance data will be used in the LSTM model.

5 Data Preprocessing

5.1 Feature Engineering

From the data we extracted from yfinance, we perform feature engineering to generate more technical indicators to use as the input to our model. The added technical indicators are as follow:

- The **Moving Average** indicators: which is a common indicator when analyzing the trend of a stock, it would calculate the average of the price within a number of days. In this project, we adopted the Moving average within 7 days (MA7), 20 days (MA20) and the upper,lower bound of them.
- **Exponential Momentum Average(EMA)**: An alternative version of Moving Average. It gives higher weights to recent prices compared to those are long time ago.
- **Logmomentum**: The momentum indicate the gradient of the stock, showing a stock is "strong" or not. The Logmomentum then are calculated log space and with a horizontal line centered at zero for easier analysis

5.2 Long Short Term Memory (LSTM) Input Data

The LSTM model is only require the single time series data as input and make prediction based on that, Hence the data input we used on this model is the Close price series only.

However, the X,y data splitting is unique from the normal approach. The main idea of it is to use a period of days(ex:60 days) to predict an upcoming single day. For instance, we used the 1-60 day as the X data to predict the number 61 days and used 2-61 day to predict number 62 day and so on. Hence, the X,y proportion would be [length of training data : length of training data-the period of days(ex:60)]. Moreover, the train_test_split shows the split proportion with [train:test]=[19:1] causing by train_data_length equals to 0.95 of the total_data_length, and the test data take part in 0.05 of total_data_length

We normalize the input data between 0 and 1 to simplify the learning process. This process is implemented in the training stage of the model.

5.3 Generative Adversarial Network (GAN) Input Data

To reflect the situation of the market on social media, we include a tweets dataset that records tweets related to several stocks. The sentiment score of each tweet is obtained from the sentiment analysis with the pre-trained model Vader, which embeds the tweets into a score representing their emotions.

The dataframe that is ready to be treated as the input of GAN is shown as:

	Date	Open	High	Low	Close	Adj Close	Volume	MA7	MA20	EMA	MACD	20SD	upper_band	lower_band	logmomentum	sentiment_score
0	2021-10-28	149.820007	153.169998	149.720001	152.570007	150.434692	100077900	149.544288	145.351999	151.359926	-11.635046	3.815009	152.982017	137.721981	5.021048	0.025634
1	2021-10-29	147.220001	149.940002	146.410004	149.800003	147.703461	124953200	149.621432	145.709499	150.319977	-10.702460	3.882887	153.475272	137.943726	5.002603	0.205803
2	2021-11-01	148.990005	149.699997	147.800003	148.960007	146.875229	74588300	149.547147	146.200500	149.413330	-10.154965	3.620448	153.441395	138.959604	4.996942	0.245867
3	2021-11-02	148.660004	151.570007	148.649994	150.020004	147.920380	69122000	149.737148	146.646000	149.817780	-9.481344	3.507520	153.661040	139.630960	5.004081	0.237369
4	2021-11-03	150.389999	151.970001	149.820007	151.490005	149.369812	54511500	150.144291	147.120500	150.932597	-8.987294	3.487778	154.096056	140.144945	5.013897	0.036300

Figure 3: GAN input dataframe.

After that, the X,y data split comes into place, the X data is simply the whole dataframe we have described above, while the y data means the targeting columns, the one we would like to predict on, in this case, the prediction is the close price.

Furthermore, we have the proportion of [train_set:test_set] equals to [207:20],Lastly, the batch_size for the mini_batched is set to be 5. Since the **sigmoid** activation function is used in the fully connected layers, we normalize the input data between 0 and 1. This process is implemented in the training stage of the model.

6 Models

In this project we experiment on two different machine learning approaches, one is the most common Long Short Term Memory model with past data input, another one is a Generative Adversarial Network model with augmented past data input. We will compare the model structure, setting as well as the model performance.

6.1 Long Short Term Memory

The LSTM we used in this project consists of 4 LSTM layers, 4 dropout layers, and 2 dense layers. The dropout layers are introduced to prevent the model from overfitting. The dense layers are used to process to LSTM output and generate predictions. The detailed properties of the model are shown as:

Layer name	**Units/Dropout rate**	**Trainable parameters**
LSTM_1	256	264192
Dropout_1	0.2	-
LSTM_2	128	197120
Dropout_2	0.1	-
LSTM_3	64	49408
Dropout_3	0.2	-
LSTM_4	32	12416
Dropout_4	0.1	-
Dense_1	32	1056
Dense_2	1	33

Table 1: LSTM model structure

The LSTM model focuses on the time-series relationship between the data and makes a regression output with respect to it. Hence, the **Root Mean Square Error** is used to calculate the trianing loss.

6.2 Generative Adversarial Network

The Generative Adversarial Network(GAN) is constructed by a generator and a discriminator, the generator consumes the multi-indicator data and generates fake closing price data. On the contrary, the discriminator would try to discriminate the fake closing price data from the real one. The generator is learning the time dependent pattern in the input data. Under this concern, we construct a generator model with 5 layers of LSTM structure and 4 layers of fully connected layers. The parameters are listed below:

Layer name	Units	Trainable parameters
LSTM_1	1024	4259840
LSTM_2	512	3147776
LSTM_3	256	787456
LSTM_4	128	197120
LSTM_5	64	49408
Dense_1	32	2080
Dense_2	16	528
Dense_3	8	136
Dense_4	1	9

Table 2: Generator model structure

The discriminator needs to learn the local patterns to cross check the true and fake input. With this assumption, the discriminator is constructed by 5 layers of 1D convolutional layers, 2 leaky ReLU layers to regulate the model output, and lastly 3 fully connected layers to generate the output. The detailed parameters are listed below:

Layer name	Units	Trainable parameters	Kernel size
Conv1d_1	8	32	3
Conv1d_2	16	400	3
Conv1d_3	32	1568	3
Conv1d_4	64	6208	3
Conv1d_5	128	8320	1
Dense_1	32	2080	-
Dense_2	16	528	-
Dense_3	8	136	-
Dense_4	1	9	-

Table 3: Discriminator model structure

Unlike the LSTM model using the root mean square error, GAN's loss function requires more consideration, firstly, on the generator side, it generates the fake data and comparing to the real data. If we range the similarity from 0 to 1 identifying the probability that the input could be real or fake, theoretically, a real data probability should be close to 1 and the fake one should be close to 0. Hence. the generator would try to make the data generated be identified close to 1. Moreover, since the generator is performing a regression problem, the sigmoid function is used instead of softmax. The loss function then would be Binary Cross Entropy at the end.

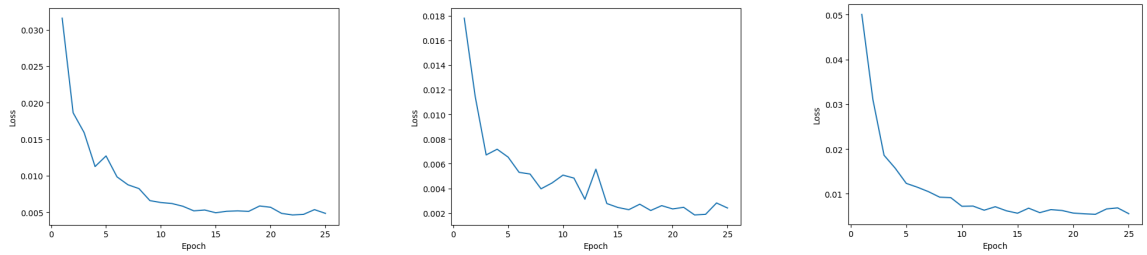
As described, the discriminator is trying to determine whether the input is real or fake, it needs to compare the Binary Cross Entropy between the 0(fake output) and the Binary Cross Entropy between

1(real output). By summing them together, the discriminator then could recognize the final score to identify how well the generator is performing.

7 Performance Analysis

7.1 Long Short Term Memory

We have chosen three stocks to analyze their performance, which are TSM, AAPL, TSLA, and generate predictions for 12 days. The LSTM model is set to train for 25 epochs with the training loss shown below:



(a) TSLA (b) TSM (c) AAPL

Figure 4: LSTM training loss

The result of evaluating these stocks on the test set is shown below, the green line are the prediction and the orange lines are the ground truth:

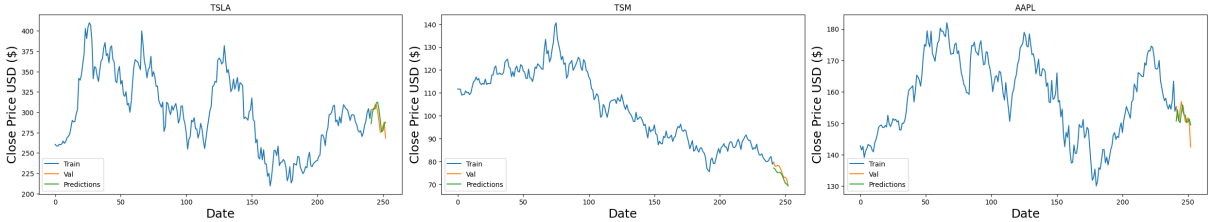


Figure 5: LSTM result

It shows that the predictions made by the LSTM model have high similarities to the actual stock data, obtaining low root mean square errors during the model training process.

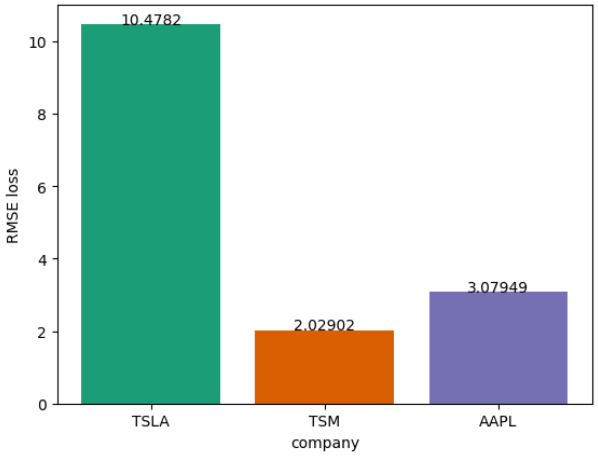
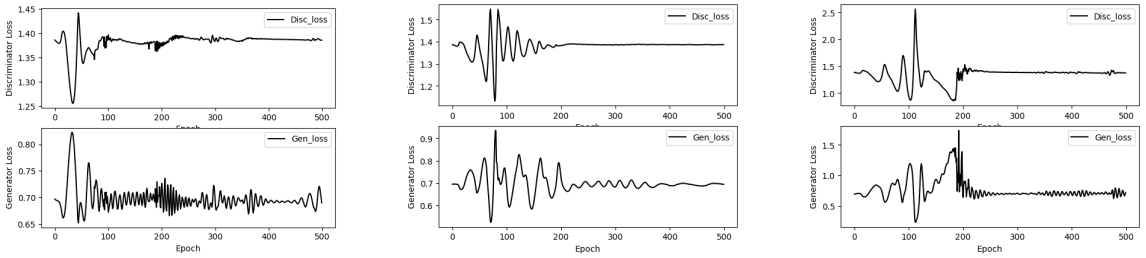


Figure 6: LSTM RMSE result

From the statistics above, we found out that the rmse of TSLA is significantly big since the range of the TSLA data is bigger than the rest two. The price range of TSLA ranges from 200 to 400, while the rmse calculates the distance between the prediction and the ground truth values, we assume this is the reason that even the prediction performance is decent compared to the other two stocks but the rmse is still way higher than the other two with smaller data range.

7.2 Generative Adversarial Network

In the GAN model, we train the model for 500 epochs on each company and keep track of the cross entropy loss of the generator and the discriminator. The loss along the training process is shown in Figure 7. The loss tends to have more fluctuation in the beginning and converge to a value near the end.



(a) TSLA (b) TSM (c) AAPL

Figure 7: Generator and discriminator loss

Then we input a month long data points to the generator to generate the prediction of the market trend of each stock, the generated predictions by the generator are shown below

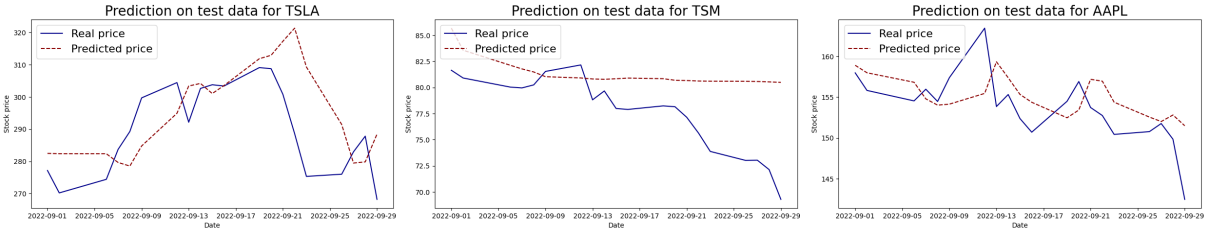


Figure 8: GAN result

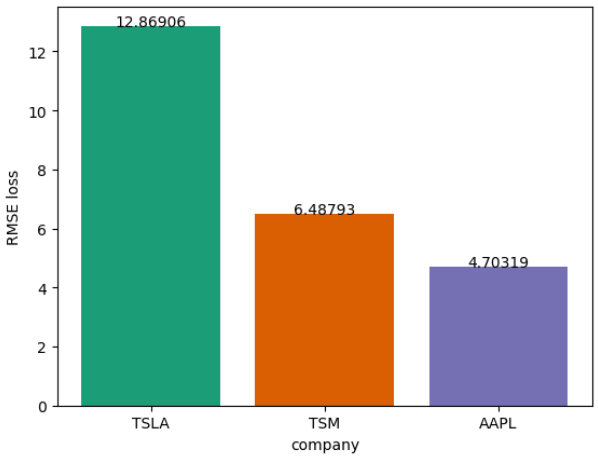


Figure 9: GAN RMSE result

It is observed that the generator can mimic the real data quite well under the limited data. We can see that the predicted output is able to follow the price trend. We found that the prediction of TSM is not as good as the others. We assume this is because of our input period selection. In the TSM input data, there aren't many ups and downs compared to the other two stocks. We suggest that this is the reason why the prediction is more flat. Moreover, in the predicted month, the price drop is larger than the price movement throughout the year. We assume this is why the prediction is higher than the ground truth.

8 Conclusion

It is seen that both LSTM model and GAN have shown the ability to analyze and make predictions on the stock closing prices. The model predicts the trend of the stock price, even though it's not a one-to-one accurate prediction. While it would still show its useful application when analyzing the stock market. The LSTM model seems to perform better than the GAN model, and this result should be normal according to [2]. To achieve better performance, there are different improvements we can do after this project:

- LSTM models usually requires large amount of data to train. In this project only one year of data is used, we can further improve the model performance by increasing the training period.
- GAN performs better when there are more input features. We may discover more indicators that the stock price depends on to use as the input in the future. Also our GAN model is not suitable for less popular stocks since the tweets information for those stocks may not be enough.

9 Appendix

9.1 Labor Distribution

The workload shared between the two of us is around 50% each. We've done the topic research, code implementation, report, and presentation video together. The collaborating process was smooth and both of us contributed a great amount of time to the project.

9.2 Topic Inspiration

Our topic is inspired by the following sources:

- Sentiment Analysis on Stock tweets:
  https://www.kaggle.com/code/anushadixit1901/sentiment-analysis-on-stock-tweets
- Stock Market Analysis + Prediction using LSTM:
  https://www.kaggle.com/code/faressayah/stock-market-analysis-prediction-using-lstm
- Stock Prediction GAN+Twitter Sentiment Analysis:
  https://www.kaggle.com/code/equinxx/stock-prediction-gan-twitter-sentiment-analysis

References

[1] G. Sonkavde, "Forecasting stock market prices using machine learning and deep learning models: A systematic review, performance analysis and discussion of implications," *International Journal of Financial Studies*, vol. 11, no. 94, 2023.

[2] G.-M. Chatziloizos, D. Gunopulos, and K. Konstantinou, "Deep Learning for Stock Market Prediction Using Sentiment and Technical Analysis," *SN Comput. Sci.*, vol. 5, pp. 1–19, June 2024.