Quantum Computing Companies Stock Price Prediction

Xiaoyan Hua
Mingqian Chen
May 15, 2022

1. PROJECT OVERVIEW

Quantum physics has changed our lives. Quantum computing and quantum communication would impact many sectors, including biotechnology, chemistry industry, energy, finance, security, etc. The race is underway. Investments to the quantum research and technology from governments, research institutions, companies all around the world increasing rapidly, and this leads to a rapid growth of the quantum computing market. [1]

With the development of the quantum technology, we see a growth in the quantum computing market. The global Quantum Computing Market size was USD 392.5 million in 2020. The market is expected to grow from USD 486.1 million in 2021 to USD 3,180.9 million in 2028, at a CAGR of 30.8% [2]

Prediction of the quantum computing market size will in turn show the trend the global quantum technology competition. Thus we want to use machine learning technology to predict stock prices of leading quantum computing companies.

This project seeks to utilize Long-Short Term Memory (LSTM) model, to predict stock prices. We use Keras to build a LSTM model to predict stock prices with historical opening, closing prices and trading volume, and visualize and compare both the predicted price values over time. We also compared the result of linear regression with LSTM.

The performance of our model is measured by root-mean-square deviation and mean absolute percentage error, which are given by equations below:

$$RMSE = \sqrt{\sum_{i=1}^{n} \frac{(\hat{y}_i - y_i)^2}{n}} \tag{1.1}$$

$$MAPE = \frac{100\%}{n} \sum_{i=1}^{n} \left| \frac{(\hat{y}_i - y_i)^2}{y_i} \right| \tag{1.2}$$

2. Long Short-Term Memory (LSTM) Model

2.1. Model comparison

1. Linear Regression: Linear regression is utilized in business, science, behavioral and social sciences to identify possible relationships between variables, and to do relevant predictions and forecasting. Linear regression performs well on models whose parameters are linearly dependent, while models which are non-linearly related to their parameters are harder to determine.

   Linear regression with time series can be well applied as long as:

   a) The variables used for response prediction are not collinearly dependent.

   b) Both feature variables and the response variable are stationary.

   c) Errors are independent from each other, etc.

   The most important assumption in linear regression is no autocorrelation, which if not satisfied can cause bias ,inconsistent and inefficient problems [11].

2. Vanilla Recurrent Neural Network: Recurrent Neural Networks(RNNs) are a type of Neural Network where the output from previous step are fed as input to the current step. Linear regression don't remember the previous information, i.e. the inputs and outputs are independent of each other. In cases to predict the next word of a sentence, the previous words should be taken in to consideration. Here's where RNNs come into play. RNNs are analogous to human learning. We don't start thinking from scratch. Similarly, loops in RNNs allow them to use past information to compute the output. This procedure is done by the most important feature of RNN, the hidden layers.

3. Long Short Term Memory (LSTM): LSTM is a special model of RNN which allows information to persist, and increases the memory of RNNs. RNNs has gradient vanish/explode problems which limit them from remembering Long-term dependencies. LSTM was introduced to solve such problems that standard RNNs suffer from, and it will be discussed in the next section.

2.2. Why choose LSTM?

The goal of this project was to study time-series data and explore as many options as possible to accurately predict the Stock Price. While RNNs do perform short-term memory, it suffers from gradient vanish/explode. As time gap grows, they'll have a hard time connecting information from previous steps to later ones.

The vanishing gradient problem is caused during back propagation. RNN gets the first-order derivative of the loss function to search for the optimal values. Recursively, the first-order derivation process will make a number smaller and smaller. [3] When a gradient value becomes extremely small, it does not contribute too much learning.

This is where LSTM comes to play. LSTM is a type of RNN that capable of learning long-term dependencies, making it better in stock prices prediction.

2.3. Architecture Of LSTM

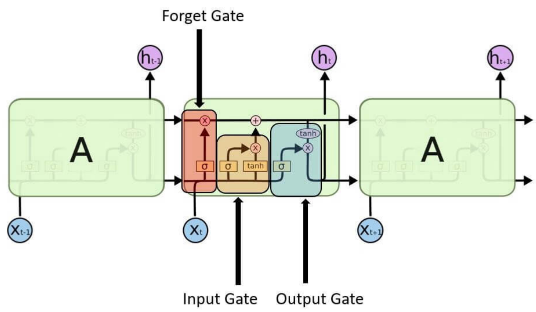

Figure 2.1: LSTM Architecture. [4]

Figure 2.1 illustrates the architecture of the LSTM. A common LSTM unit is composed of a cell, an input gate, an output gate and a forget gate [5].

1. Forget gate: Forget gate is used to decide what information to discard and what to keep from the cell state. This decision is made by a sigmoid layer. For each number in the cell state C_{t-1}, the forget gate takes h_{t-1} and x_t, which are output of the memory cell at $t-1$ and input at t, and outputs a number between 0 and 1 . A 1 represents keep the information, and a 0 represents discard the information.

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \tag{2.1}$$

2. Input gate: Input gate consists of two parts, a sigmoid layer and a tanh layer, and is to decide what information to store in the cell. It takes the previous hidden state and current input to a sigmoid function, which transforming the values to the range of 0 to 1. 0 represents unimportant values we would like to ignore, and 1 represents important values we would like to keep.

   The tanh layer creates a vector of new candidate values, $\tilde{C}_t$, which stabilizing the training process by normalizing the values into the range [-1, +1]. The we update to the state by combining these two.

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \tag{2.2}$$

$$\tilde{C}_t = tanh(W_C \cdot [h_{t-1}, x_t] + b_C) \tag{2.3}$$

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \tag{2.4}$$

3. Output gate: The output is a filtered cell state. The sigmoid function decides which part of the cell state we would like to output. Then the cell state is send through the tanh layer which normalize values to the range [-1,1]. Multiplying it by the output of the sigmoid gate, we get the filtered information we want.

$$i_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \tag{2.5}$$

$$h_t = o_t * tanh(C_t) \tag{2.6}$$

3. Methodology

We employed linear regression, improved linear regression and LSTM in quantum computing companies stock prices.

3.1. Data Processing

We take the data file quantumstock1.csv from the Kaggle project "Quantum Computing Companies Stock Prices" [6]. We analysed stock prices of Accenture plc (ACN), IBM, Microsoft Corporation (MSFT), Quantum eMotion Corp (QNC.V), Intel Corporation (INTC), Baidu Inc (BIDU), Nokia Oyj (NOK), and Mitsubishi Electric Corporation (MIELY), and especially look into the stock price of ACN as an example.

Figure 3.1 shows the close prices of ACN, which has an increasing trend. A significant increase happens around 14000 day.

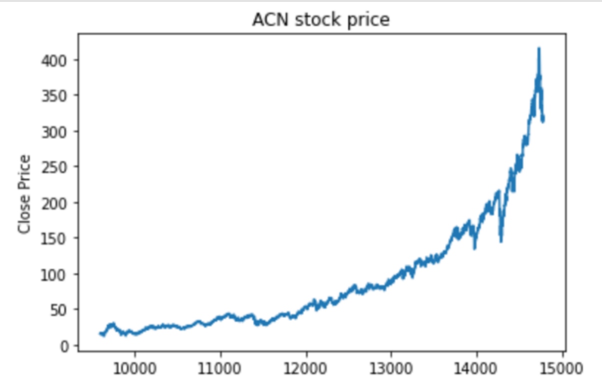

Figure 3.1: Quantum Companies Stock Prices Raw Data. [6]

	Date	open	high	low	close	adjclose	volume	ticker
9592	2001-07-19	15.10	15.29	15.00	15.17	11.223610	34994300.0	ACN
9593	2001-07-20	15.05	15.05	14.80	15.01	11.105230	9238500.0	ACN
9594	2001-07-23	15.00	15.01	14.55	15.00	11.097830	7501000.0	ACN
9595	2001-07-24	14.95	14.97	14.70	14.86	10.994254	3537300.0	ACN
9596	2001-07-25	14.70	14.95	14.65	14.95	11.060840	4208100.0	ACN

Figure 3.2: ACN Stock Prices [6]

The table shown if figure 3.2 contains the raw data of ACN from July 19 2001 to July 25 2001, which contains opening, closing prices, high, low values, adjclose and volume. We normalize the dataset with **MinMaxScaler** and split the dataset into training and testing sets with percentage of 90% and 10%.

	open	high	low	close	adjclose	volume
0	0.009139	0.007479	0.009195	0.008227	0.006056	0.389084
1	0.009015	0.006887	0.008698	0.007830	0.005764	0.101229
2	0.008892	0.006788	0.008077	0.007805	0.005746	0.081810
3	0.008768	0.006690	0.008450	0.007458	0.005491	0.037510
4	0.008151	0.006640	0.008325	0.007681	0.005655	0.045007

Figure 3.3: Normalize the dataset

3.2. Linear Regression

Stock price is time-series data as it is collected sequentially in time, and recorded in a specific time interval. Therefore, we apply linear regression to carry out the trend of ACN closing prices. In figure 3.4 and 3.5, we compared the trend of the closing prices with the actual data. Figure 3.4 clearly shows an upward trend, while figure 3.5, which is plotted only with test data, has a much flatter trend, and is unable to reflect the trend of the stock prices very well. Stock prices have seasonal fluctuation and irregular fluctuation which can not be captured by the trend, therefore the naive linear regression fails in forecasting stock prices. Also, as mentioned in section 2.1, in order to perform a successful prediction by linear regression, variables in the data should be independent from each other. Here is why time series data often causes a problem, as in stock prices forecast, the observed values are longitudinal in nature, thus will produce autocorrelation [7]. The performance of the model is evaluated by RMSE and MAPE, and equals to 0.3180 and 0.4124.

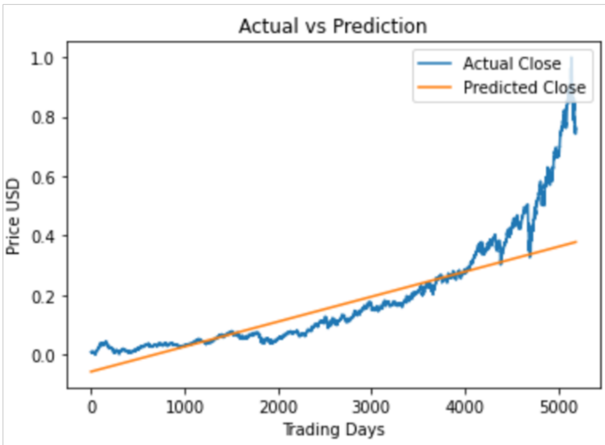

Figure 3.4: Linear Regression for Time Series: train set test set

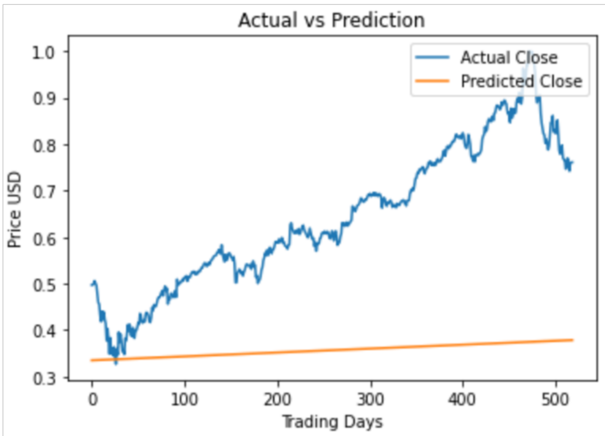

Figure 3.5: Linear Regression for Time Series: test set

In order to solve this problem, we carried out an improved linear regression algorithm by constructing Simple Moving Average (SMA). SMA, calculated by equation below, is the average price over a specified time period. We use a 30-day moving average as input to predict the close price. Figure 3.6 shows a better fitting trend. SMA has a better performance in fitting the trend than a naive linear regression model with RMSE and MAPE equals to 0.0401 and 0.0494. However, by comparing the actual data with the trend predicted by SMA, we see a delayed feedback, which fails SMA to forecast stock price. The reason is that SMA is a lagging indicator, it only look at historical data and tends to move behind the actual price. In case when the price is dropping and then comes back up, it may continue going higher after the price does[8].

$$SMA = \frac{1}{n}\sum_{i=1}^{n} P_i \tag{3.1}$$

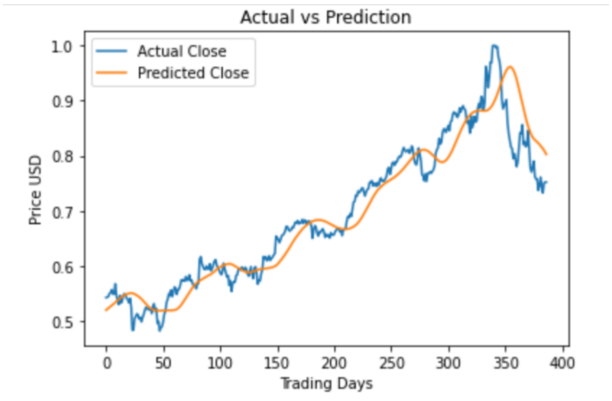

Figure 3.6: Linear Regression for SMA

3.3. Stock Price Forecast by Long Short-Term Memory Model

To solve the problems in linear regression model, we implement LSTM model. First, we dropped some variables, and only consider the opening, closing prices and volume as those are the most important parameters. If we get a higher closing price than the opening price by the end of day, we make profit, otherwise we lose money. The volume of share also matters, as a rising volume indicate a rising market, while decreasing stock price with a decreasing volume shows lack of interest, and may imply a potential loss in the future.

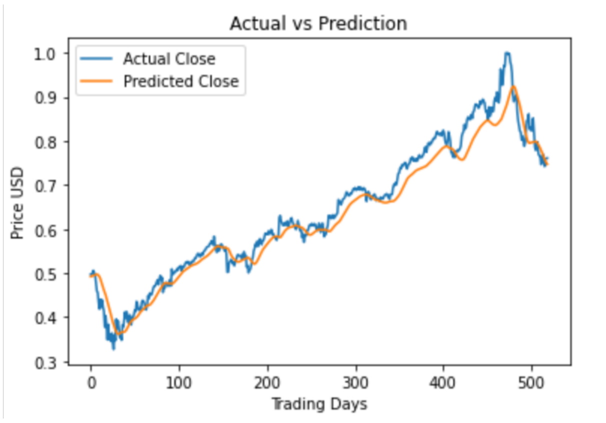

Figure 3.7: Basic LSTM Model

Figure 3.7 is the prediction done by the basic LSTM with one hidden layer. We pass the last 60 days of the data for training the model and output the predicted closing price. As we can see from the figure, predicting done by LSTM performs a better fitting than linear regression with higher accuracy (RMSE:0.0336 , MAPE:0.0406) but there is still space for improvement.

In our model, we perform three ways to improve the LSTM model:

1. Add hidden layer: adding stacked multi-layers is for extracting more abstract information. In this case, we increase the hidden layers from 1 into 3.
2. Add Dropout: adding dropout is a regular method to reduce overfitting and improving model performance. In this case, we add dropout of 0.2 at each layer of the LSTM.
3. Increase epochs: epochs is the times that the learning algorithm will work through the entire training dataset. In this case, we increase epochs from 10 to 100.

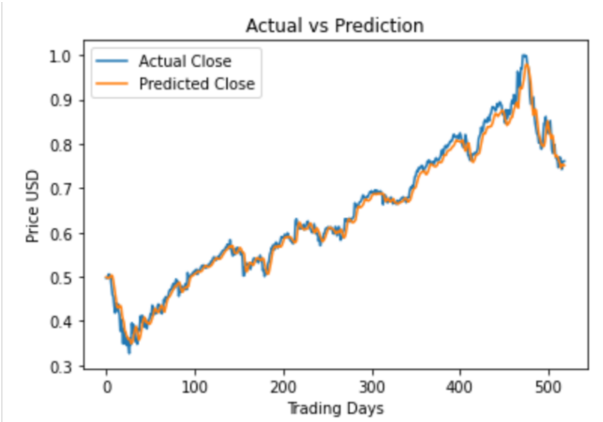

Figure 3.8: Improved LSTM Model

Figure 3.8 shows a significant improvement in out prediction with lowest RMSE (0.0185) and MAPE (0.0224) values.

3.4. Comparison of Linear Regression and LSTM

Table 3.1 records the accuracy of four models (linear regression, linear regression with SMA, basic LSTM and improved LSTM) by RMSE and MAPE. The improved LSTM has the best accuracy with the lowest RMSE(0.0185) and MAPE(0.0224) values, which is consistant with our figures.

Table 3.1: Accuracy for four models of ACN

	Linear Regression for Time Series	Linear Regression for SMA	Basic LSTM	Improved LSTM
RMSE	0.3180	0.0401	0.0336	0.0185
MAPE	0.4124	0.0494	0.0406	0.0224

We have also analyzed the stock prices of IBM, MSFT, QNC.V, INTC, BIDU ,NOK and MIELY (See appendix for details). Based on results of ACN, we choose linear regression with SMA and the improved LSTM for comparison. Table 3.2 records the accuracy for those two models of each company. In most cases, the model performance is consistent with our conclusion that the improved LSTM has the best performance. However, we noticed one reverse result of MSFT. In this case, linear regression seems perform better than LSTM. We checked the figures and found that both models fit very well, but LSTM model has an overall deviation at the peak. This will be our next research direction.

Table 3.2: Accuracy for models of 8 quantum computing companies

Company	Linear Regression		LSTM	
	RMSE	MAPE	RMSE	MAPE
ACN	0.0401	0.0494	0.0185	0.0224
IBM	0.0351	0.0419	0.0152	0.0174
MSFT	0.0285	0.0356	0.0343	0.0416
QNC.V	0.0474	0.1724	0.0260	0.0933
INTC	0.0470	0.0501	0.0214	0.0214
BIDU	0.0648	0.0762	0.0297	0.0381
NOK	0.0069	0.1070	0.0034	0.0438
MIELY	0.0377	0.0420	0.0181	0.0207

4. Conclusion

We have analyzed and compared the performance of four models in the prediction of quantum computing companies stock price. Each model has its advantages and disadvantages. Linear regression model is easy to implement but fails to capture nonlinear changes in price, and is unable to reflect other features. LSTM model solves the lagging variable problem of linear regression and gradient vanish problem of RNN, and performs a good forecast. But LSTM model also requires more resources and time to get trained. It is also affected by random weight initialization and is prone to overfit.

A. Appendix: Results of Other Quantum Computing Companies

IBM	RMSE	MAPE
Linear Regression	0.0351	0.0419
LSTM	0.0152	0.0174

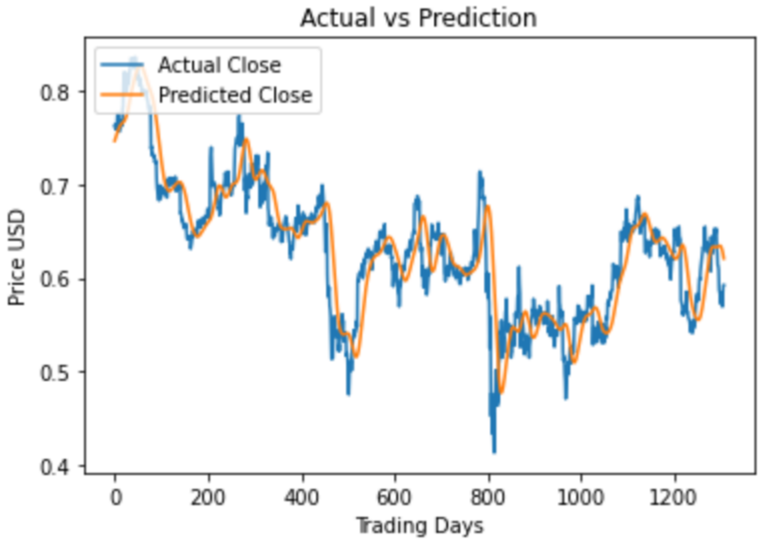

Figure A.1: Linear Regression model of IBM

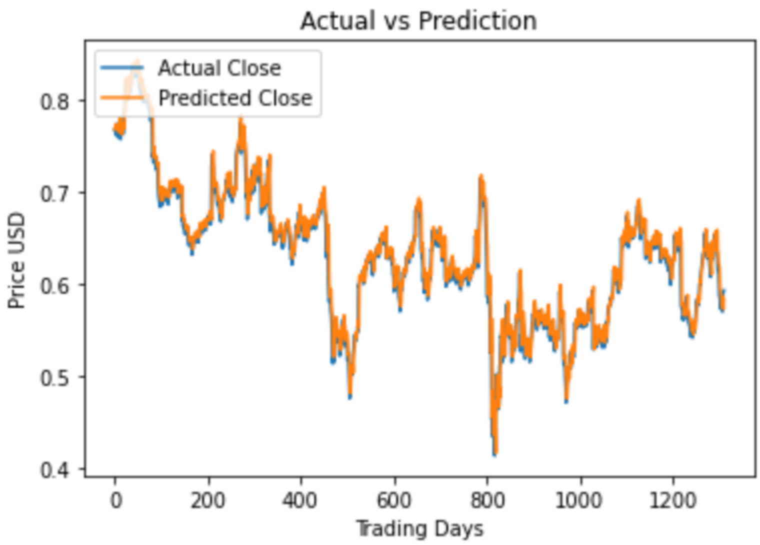

Figure A.2: LSTM model of IBM

MSFT	RMSE	MAPE
Linear Regression	0.0285	0.0356
LSTM	0.0343	0.0416

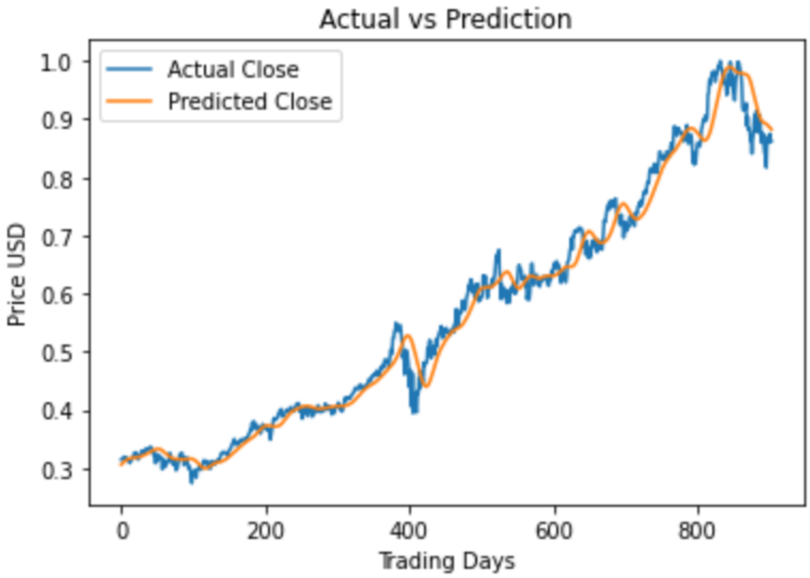

Figure A.3: Linear Regression model of MSFT

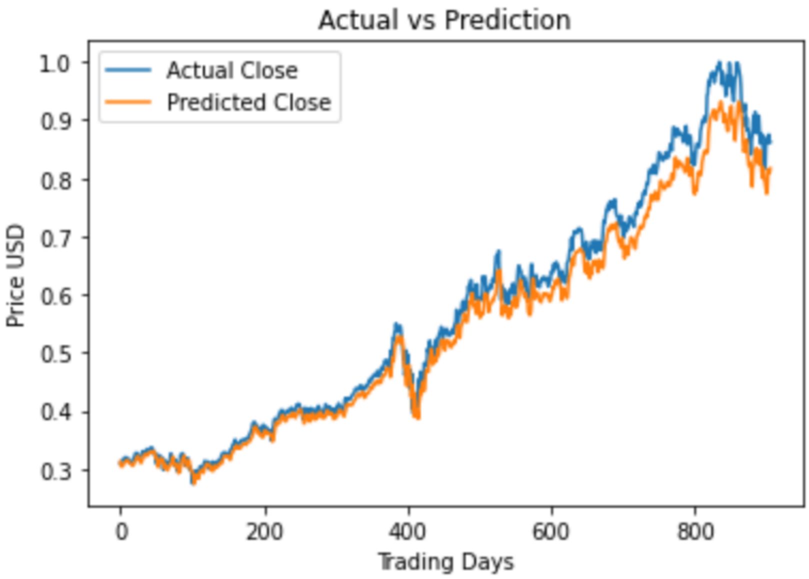

Figure A.4: LSTM model of MSFT

QNC.V	RMSE	MAPE
Linear Regression	0.0474	0.1724
LSTM	0.0260	0.0923

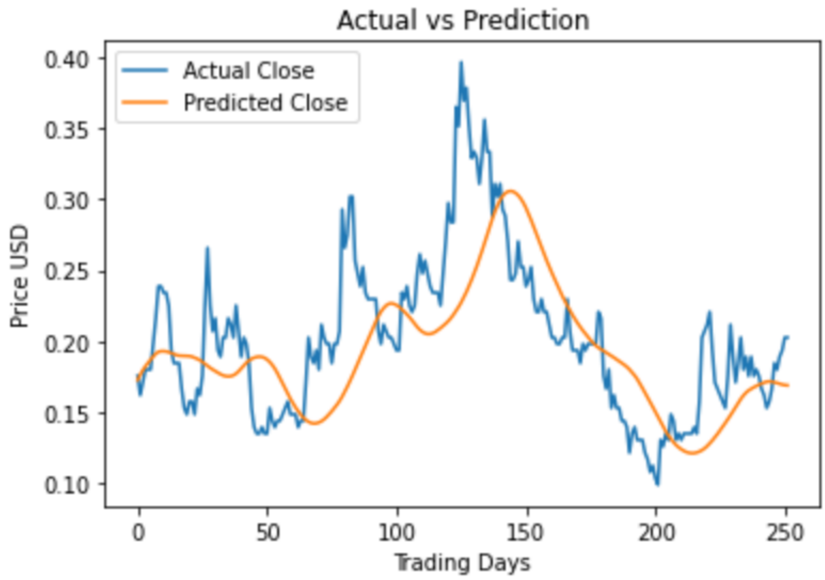

Figure A.5: Linear Regression model of QNC.V

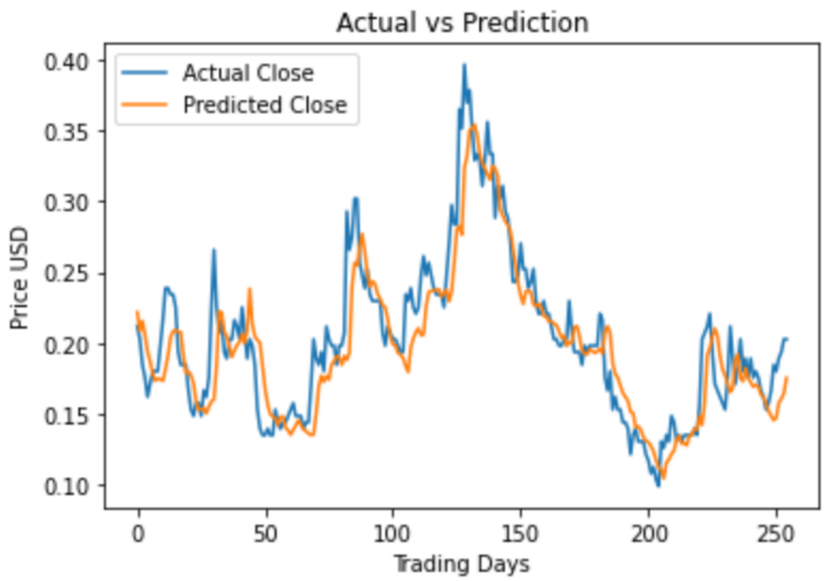

Figure A.6: LSTM model of QNC.V

INTC	RMSE	MAPE
Linear Regression	0.0470	0.0501
LSTM	0.0214	0.0214

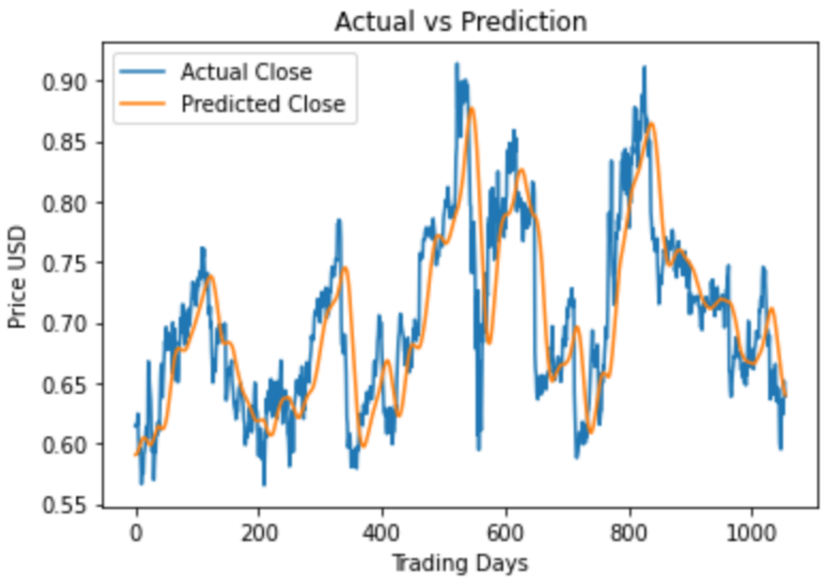

Figure A.7: Linear Regression model of INTC

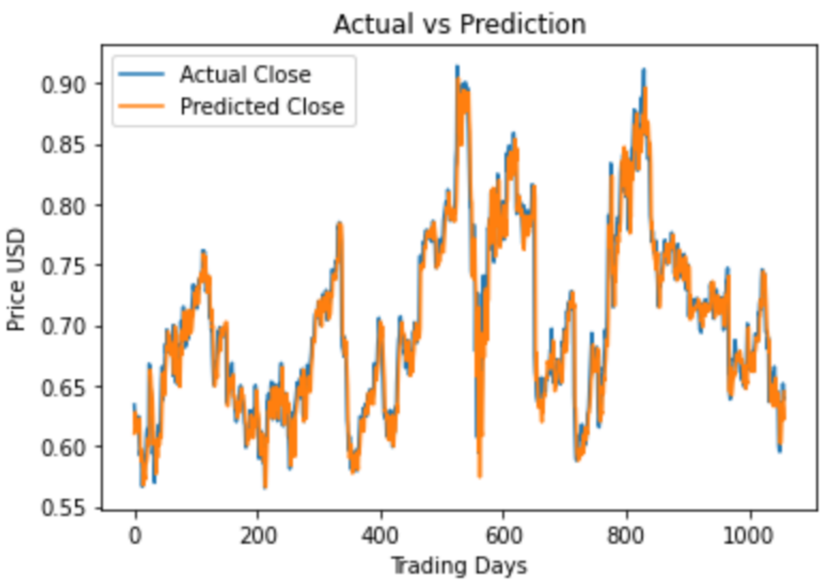

Figure A.8: LSTM model of INTC

BIDU	RMSE	MAPE
Linear Regression	0.0648	0.0762
LSTM	0.0297	0.0381

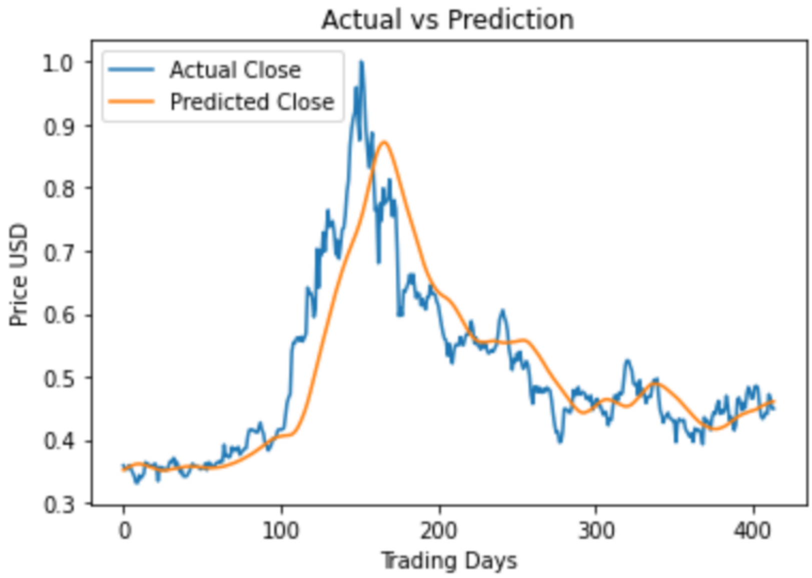

Figure A.9: Linear Regression model of BIDU

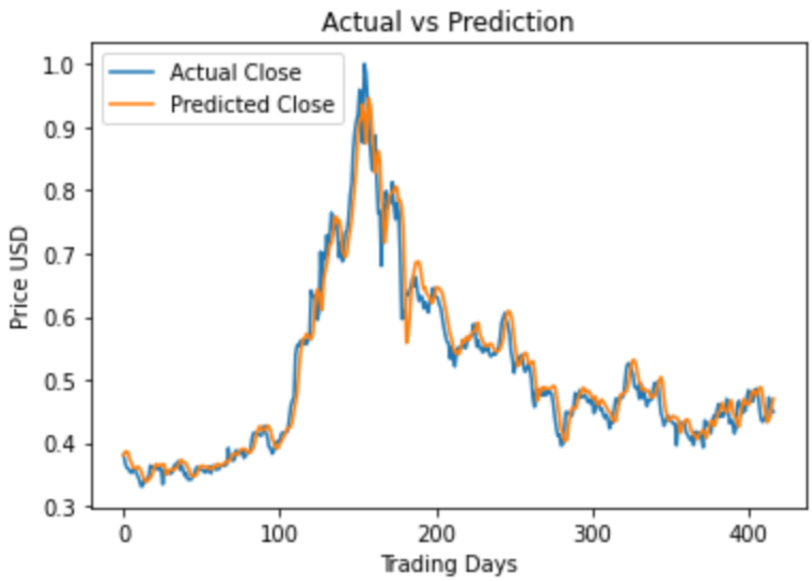

Figure A.10: LSTM model of BIDU

NOK	RMSE	MAPE
Linear Regression	0.0069	0.1070
LSTM	0.0034	0.0438

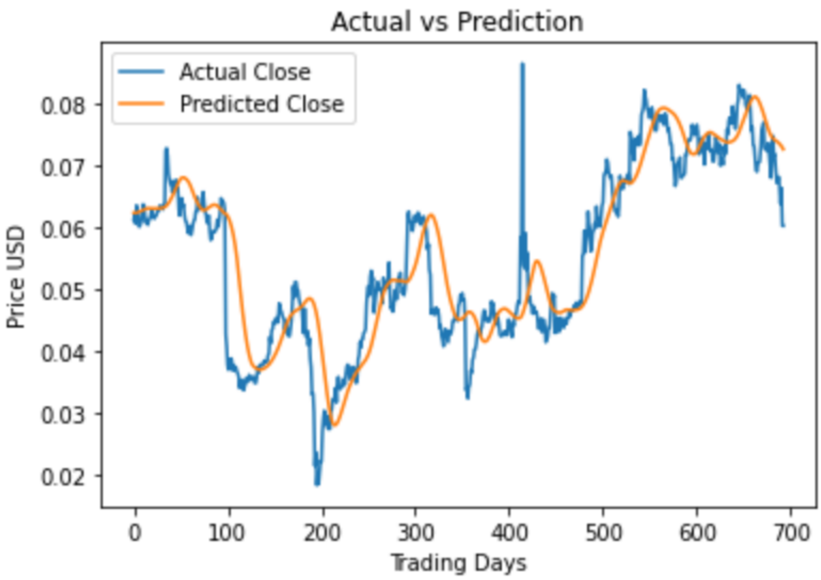

Figure A.11: Linear Regression model of NOK

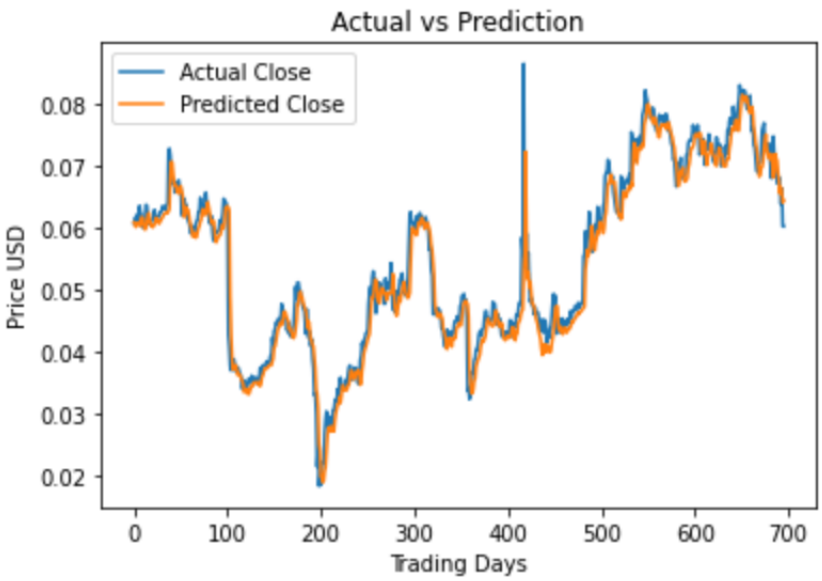

Figure A.12: LSTM model of NOK

MIELY	RMSE	MAPE
Linear Regression	0.0377	0.0420
LSTM	0.0181	0.0207

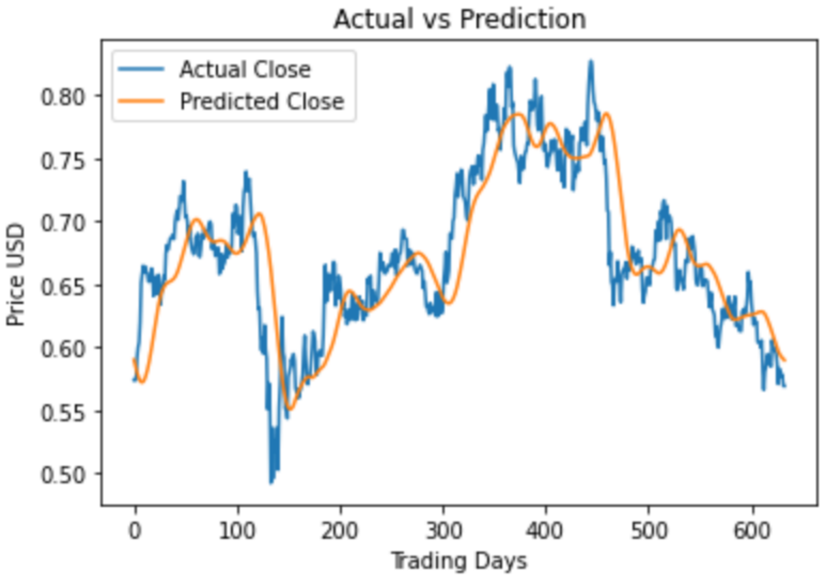

Figure A.13: Linear Regression model of MIELY

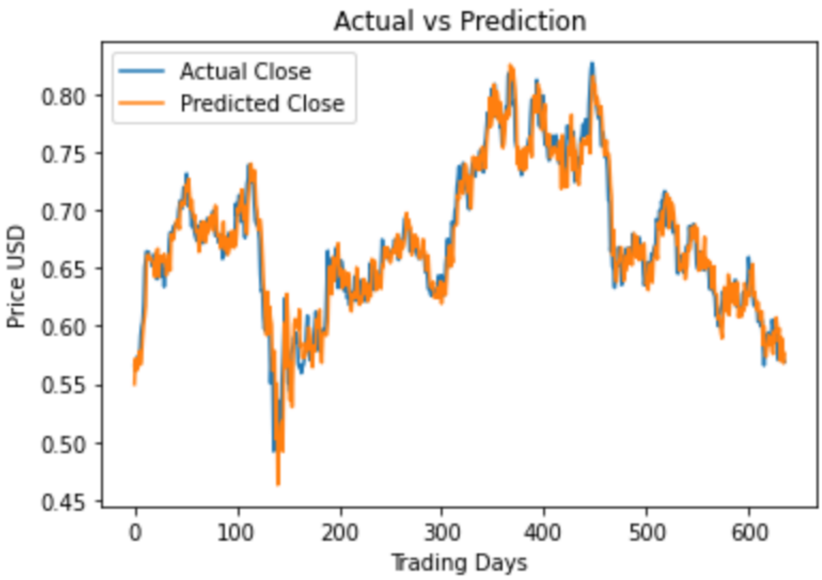

Figure A.14: LSTM model of MIELY

References

[1] https://hbr.org/2020/09/are-you-ready-for-the-quantum-computing-revolution

[2] https://www.fortunebusinessinsights.com/quantum-computing-market-104855

[3] https://towardsdatascience.com/illustrated-guide-to-lstms-and-gru-s-a-step-by-step-explanation-44e9eb85bf21

[4] http://colah.github.io/posts/2015-08-Understanding-LSTMs/

[5] https://en.wikipedia.org/wiki/Long_short-term_memory

[6] https://www.kaggle.com/datasets/nourajo/quantum-computing-companies-stock-data

[7] https://www.alpharithms.com/predicting-stock-prices-with-linear-regression-214618/

[8] https://www.cmcmarkets.com/en/trading-guides/leading-and-lagging-indicators

[9] Donald E. Knuth (1986) *The TEX Book*, Addison-Wesley Professional.

[10] Leslie Lamport (1994) *LATEX: a document preparation system*, Addison Wesley, Massachusetts, 2nd ed.

[11] https://stackoverflow.com/questions/61511094/what-s-the-advantage-of-using-lstm-for-time-series-predict-as-opposed-to-regress