Demystifying Long-Short Term Memory Model to Predict Stock Prices

Ruichen Kong, Hanyu Wang, Li Zhang, research advisor: Vitaly Ford

October 1, 2022

Abstract

Correctly predicting stock prices is a dream of any investor and one of the most effective ways to make profits. "Long Short-Term Memory" (LSTM) [7] neural networks is a sophisticated machine learning algorithm containing a "memory cell" that can hold learned information in memory for a long period of time. Our first research question is whether LSTM is usable in predicting the stock market. Furthermore, if LSTM is not suitable in predicting stock market on its own, can we improve it to facilitate better stock prediction? If it works, can we improve the accuracy of its predictions? Therefore, we pose the second research question: can we improve the LSTM model beyond just changing the LSTM model's parameters, such as the number of hidden layers, number of hidden nodes per layer, epochs, and batch size? Also, is it possible to apply other techniques to improve the LSTM model besides changing its parameters? To address these research questions, first, we tested the short-term prediction effect of LSTM with two different methods. One method is about finding how many days the predicted trend matched the real stock market movement. The other one is concentrated around the idea of determining for how many days the predicted price fits in the range between maximum and minimum prices of the stock at the specific time frame. Next, we conducted experiments to identify how well LSTM can predict prices long term (for instance, one week ahead). We discovered that the results were not satisfactory in both short- and long-term predictions. Hence, we concluded that the traditional LSTM is not suitable for predicting stock price, and it should be improved. To determine the best outcome, we experimented with LSTM parameters to determine the best set of configurations. The essence of this work is the concept of particle swarm optimization that facilitated slight improvements of the traditional LSTM model as it helped us to identify the best fit parameters. Our research confirmed that the use of the traditional LSTM model alone is meaningless for the stock market forecasting. Even slight improvements to LSTM do not produce satisfactory results.

1 Introduction

The stock market has been one of the major battlefields for many investors, and predicting stock prices is one of the most effective ways for them to make profits. Traditionally, the stock prices are predicted with methods such as K-line diagrams [13] and Yin-Yang lines [4]. Statistical methods like Weighted Moving Average [3] and Exponential Smoothing [6] are also very popular. Nowadays, with the prosperity of the computer science field, machine learning has become a dominating research area in this domain.

One of the most widely used machine learning algorithms is neural networks that process information, imitating human brains. There are many types of neural networks, and Recurrent Neural Networks (RNN) [2] are known to be convenient and helpful as time series predictors due to their ability to process information of any length regardless of the size of the model and remember each data throughout the time.

Compared to a standard RNN, a special kind of RNN called "Long Short-Term Memory" (LSTM) is more sophisticated as it contains a "memory cell" that can hold the information in memory for a long period of time. In this paper, we propose to use the LSTM model to predict stock prices for some major companies as well as some relatively small ones and to test the accuracy of this model to determine if it is practical and effective to be applied in stock markets.

2 Motivation

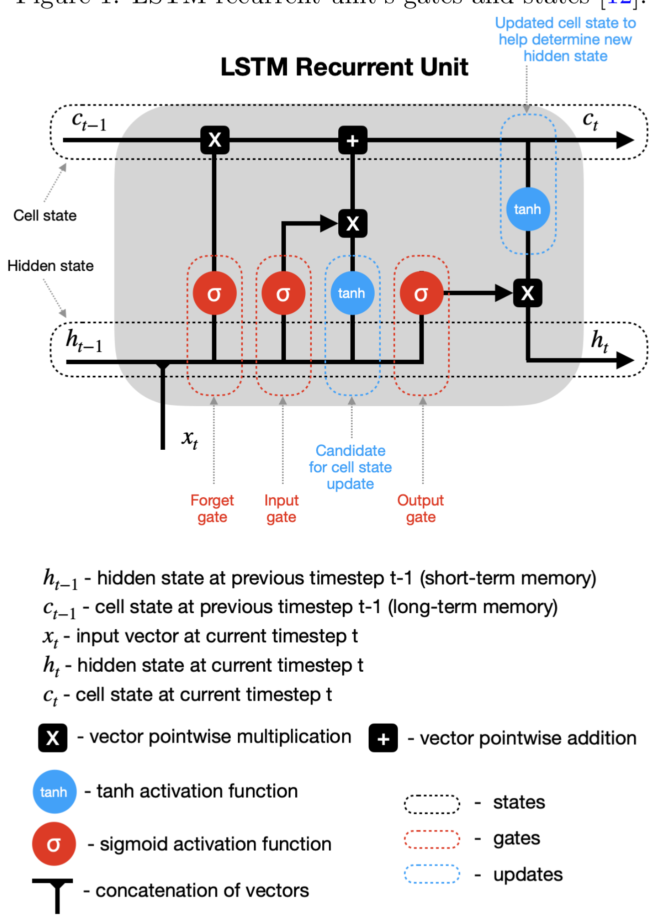
LSTM has demonstrated its excellent predictive ability in many fields, especially those that deal with time series data. For instance, LSTM has been successfully applied in such tasks as human action recognition [5], protein homology detection [9],robot control [11],and market Prediction [10]. Based on its strong forecasting ability, people began to use LSTM to predict the stock market. However, they are divided on whether it can successfully predict the stock market. Some researchers believe that using LSTM to predict the stock market can only be useful for teaching, others feel that LSTM can successfully predict the stock market for profit. In this paper, we decided to address the feasibility of applying LSTM on the stock data.

Consequently, our first research question is whether LSTM is usable in predicting the stock market, or if it is just a model that can be applied to stocks for educational purposes only. This research question leads to an important notion that should be addressed as well. Specifically, if LSTM cannot predict the stock market on its own, can we improve it to facilitate better stock prediction? If it works, can we improve the accuracy of its predictions? Therefore, we posed the second research question: can we improve the LSTM model beyond just changing the LSTM model's parameters, such as the number of hidden layers, number of hidden nodes per layer, epochs, and batch size? Can we use other techniques to improve LSTM that are outside of the LSTM model itself (such as augmenting the input data or applying addition data processing methodologies)?

To answer the above-mentioned questions, we conducted many different experiments that demonstrated the level of effectiveness of LSTM in different model configurations. Additionally, we evaluated a few approaches that could improve the accuracy of LSTM.

3 LSTM Model

Figure 1: LSTM recurrent unit's gates and states [12].



The LSTM neural network has three types of layers: one input layer, one output layer, and hidden layers. In our research, we use 3 hidden layers in addition to the input and output layers. A standard LSTM model's unit (neuron) contains forget gates, input gates, and out gates as shown in Figure 1. Forget gates are closely related to whether information should be remembered by using sigmoid functions ranging between 0 and 1. If the information needs to be strongly remembered, the sigmoid function approaches 1, otherwise it approaches 0. Input gates are used to assist in identifying important elements to add to the cell state. Only the information that is considered important by the input gate can enter the cell. The output gate determines the output value based on the cell state.

By updating the cell state, the hidden state, and returning the latest cell and hidden states to the recurrent cell, the loop continues until we reach the end of the sequence.

4 Short- and Long-term Predictions

4.1 Short-term prediction

We conducted the short-term prediction experiments in which the LSTM model is used to predict the price of the following day based on the previous sequence of days. Four different methods were utilized to determine if short-term prediction achieves satisfactory results. The methods are described below.

4.1.1 Visual Evaluation of the Stock Patterns

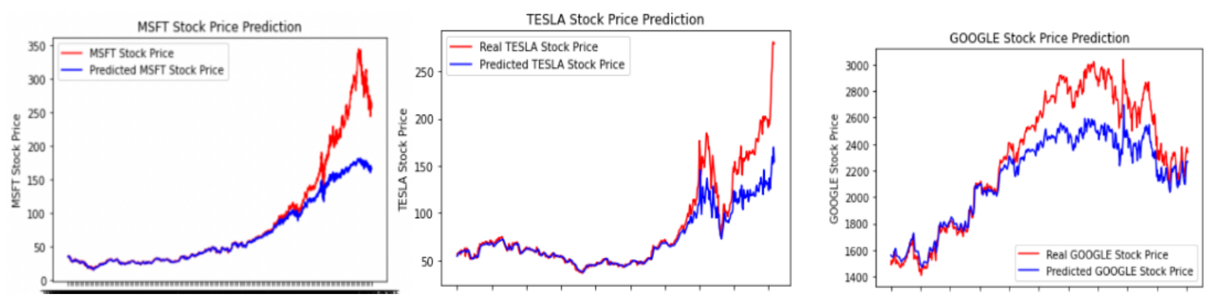


Figure 2: Visualization of LSTM application on common stocks to predict prices

On Figure 2, the red line reprsents the real data and the blue line - the predicted data. We can see that the predicted data follows the trend of the real data. It seems that LSTM predicts the trends relatively well based on these graphs. However, such a visual methodology is very subjective and can only be used as an initial stage of analysis. Thus, we continued investigating LSTM application in stock prediction using the other three methods.

4.1.2 LSTM Predicted Price Comparison with the Actual Maximum and Minimum

On Figure 3, the red line is the real maximum stock price, the yellow line is the real minimum stock price, and the blue line is the LSTM's predicted stock price. We want to identify how many days the predicted stock price fits between the maximum and minimum stock prices. Therefore, we analyzed three data sets: the number of days when the stock price is greater than the minimum price, the numbers of days when the stock price is less than the maximum price, and our predicted set of prices, using LSTM. Based on these data, we can calculate the percentage of successful prediction between highest and lowest prices using the following variable c (where 378 is the total number of the test data).

(the numbers of days where the stock price is greater than min price)	a = [196 269 141]
(the numbers of days where the stock price is less than max price)	b = [307 178 280]
the percentage of prediction between highest and lowest is	c=(a + b)/378 -1

If the predicted stock price is between the actual minimum and maximum prices, it means that the predicted stock price actually appeared on that day, and we deem it as a successful prediction outcome. However, our final results demonstrate that the correct percentage of the prediction within

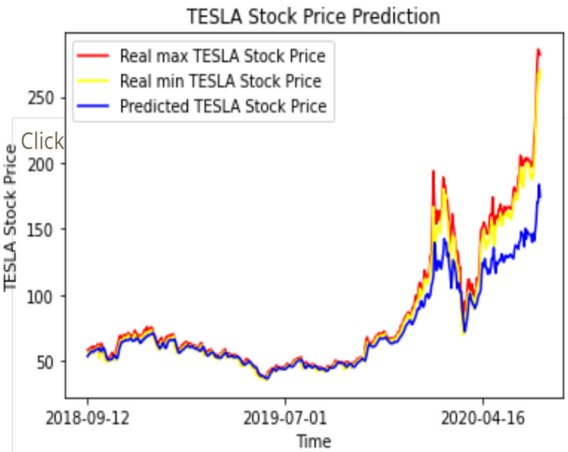


Figure 3: Tesla's price prediction

the range of the highest and lowest actual prices is only about 20%. Thus, LSTM did not achieve a satisfactory prediction.

4.1.3 Comparison of the Predicted and Actual Stock Market Trends

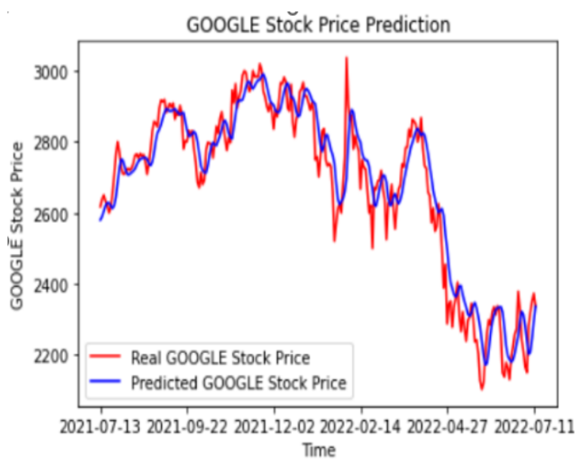


Figure 4: Google's short-term prediction

Company name	Total days	Correctly predicted days
Google	251	121
Microsoft	375	192
Hanover Insurance group	74	32
Heritage Insurance Holdings, Inc	74	40

Table 1: Total predicted days & correctly predicted days

Based on the Figure 4, we can visually determine that the predicted price (blue) closely follows the actual price (red). However, we decided not to draw conclusions prematurely by looking at the graph in this method. Instead, we developed a script to evaluate how many days the predicted trend matched the real stock market movement. The results (Table 4.1.3) demonstrated that 121 days were correctly predicted among 251 days, meaning that LSTM predicts the trend the same way as people tossing a coin to guess whether the stock price would rise or fall.

4.1.4 Evaluation of the Stock Price Prediction Error

We calculated the stock price prediction error based on the difference between predicted data and actual data, divided by the actual data on the specific day (the LSTM parameters and their respective error values are presented in Equation 1). The results convey that there is about 11% difference between the actual and predicted stock prices. In other words, LSTM's predicted values are on average within 11% margin from the actual prices.

$$
\begin{gathered}
Error\ value = 0.1147 \\
Percent\ train = 0.8,\ Model\ unit = 160,\ Batch\ size = 32,\ Timesteps = 30 \\
Error\ value = 0.1155 \\
Percent\ train = 0.8,\ Model\ unit = 160,\ Batch\ size = 32,\ Timesteps = 60 \\
Error\ value = 0.0994 \\
Percent\ train = 0.8,\ Model\ unit = 160,\ Batch\ size = 64,\ Timesteps = 60 \\
Error\ value = 0.1090 \\
Percent\ train = 0.8,\ Model\ unit = 160,\ Batch\ size = 64,\ Timesteps = 30
\end{gathered} \tag{1}
$$

4.1.5 LSTM Configuration Challenges

While experimenting with LSTM to predict stock prices, we attempted to adjust model parameters in order to achieve the best prediction outcome. However, whenever we found a set of parameters that would produce the best result on one company's data, it would perform terribly on another. This is expected from stock market data as there are no two companies that are similar in stock price evaluation and trends. We also noticed that not only different model parameters led to achieving completely different results but also the exact same model parameters produced inconsistent results every time we would run the program on the same stock but different time frames. This means that LSTM needs to be constantly re-trained for stock prediction.

4.2 Long-term Prediction

For the long-term prediction, the future price of the stock was predicted based on the data including previously predicted one-day prices using the Rolling Window [1] idea, demonstrated on Figure 5. In other words, we ran LSTM to first predict Day 1 of the week based on the historical data. Then, we added the Day 1 prediction to the historical data, shifting the time series input stream by one day ahead and getting ready to predict Day 2. We continued the same process to conduct stock prediction long-term for the rest of the future days. Afterwards, we compared the results with the actual data from the stock market.

The results obtained for Microsoft and Google are shown in the Table 2 and Table 3.

Date	Pridicted	Real
06/08/2022	333.07	248.31
06/09/2022	410.58	243.86
06/10/2022	448.24	245.11
06/13/2022	456.59	245.11
06/14/2022	446.40	234.86

Table 2: Microsoft's predicted stock prices and its real stock prices

From the above-mentioned tables, we concluded that in our long-term prediction, the numbers as well as the trends of the actual stock prices and the predicted ones do not match and have significant variability.

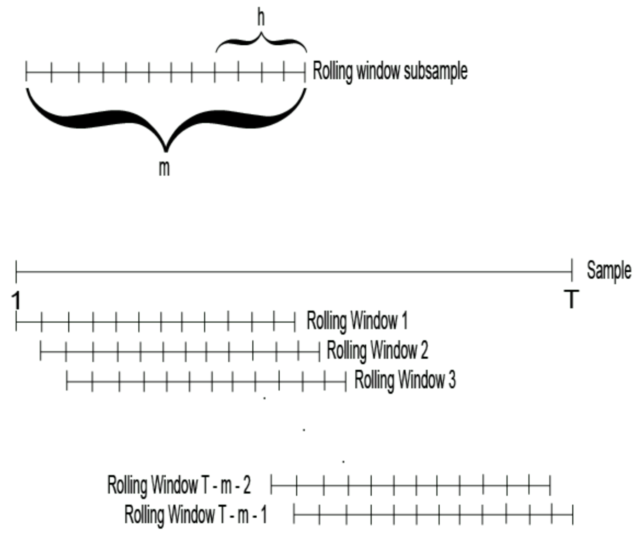


Figure 5: Rolling window

Date	Pridicted	Real
06/13/2022	117.15	112.64
06/14/2022	118.18	110.83
06/15/2022	118.35	112.96
06/18/2022	118.56	113.44
06/19/2022	118.80	111.73
06/20/2022	119.05	114.06
06/21/2022	119.31	115.09
06/22/2022	119.58	111.81
06/25/2022	119.86	108.88
06/26/2022	120.14	107.43
06/27/2022	120.43	109.60
06/28/2022	120.72	112.80
06/29/2022	121.01	113.40

Table 3: Google's predicted stock prices and its real stock prices

5 Determining the Best LSTM Model Parameters

To identify whether the best combination of parameters exists and determine what it would be if it does, we developed multiple LSTM run processes for each of the 9 companies we selected by measuring the error value for each case to find the best combination. For each of the companies, we ran the model with each combination of the model parameters several times, and calculated the average of the errors.

For the stock data from 9 companies, we used 80% of historical data as training data, and the remaining data were utilized as testing data. The model unit (number of hidden neurons per hidden layer) was set as 100, 300, 500, and 700, while the batch size was 32 and 64, which let us obtain 8 combinations in total as shown in Figure 6.

The Figure 6 shows the four companies among all of the companies we selected, as well as the average errors we obtained. There is only a slight difference among the average of errors. Generally speaking, when batch size equals 32 or 64, whether model units are 300 or 500 does not matter. This, in turn, leads to the conclusion that it is impossible to identify the best parameters for the LSTM model because all combinations we have attempted achieved equal results.

We determined that when the batch size equals 32, 700 model units always achieves the worst prediction among all of the conducted experiments. The same applied to the pair of parameters, where 64 was chosen to be the batch size and 100 defined the model units. Such a result demonstrates that it is impossible to establish the best batch size and model unit as they are all relatively equal in

Ticker Name	Batch_size	100	300	500	700
ENB	64	2.35987%	2.31649%	2.16593%	2.17779%
ENB	32	1.75935%	1.75429%	1.97823%	2.26364%
COP	64	3.35373%	3.21174%	3.03601%	3.08520%
COP	32	2.83871%	2.72202%	2.80816%	2.91890%
CVX	64	2.40066%	2.38536%	2.30833%	2.21003%
CVX	32	2.11512%	2.02711%	2.06744%	2.26281%
TTE	64	2.18369%	2.14469%	2.09972%	2.07014%
TTE	32	2.18692%	2.19756%	2.09816%	2.92448%

Figure 6: Configuring LSTM parameters

their performances: except for the two cases mentioned above, the average error for the other cases was approximately 2.5%.

6 Result Evaluation based on the Same Model Parameters

The following diagrams (Figures 7, 8, 9) depict the histogram and probability distribution of the error values for different model parameters, which can show that the prediction results are completely different even though we kept the same model parameters all of those experiments.

We can see that the error values are mainly distributed within a certain range. Interestingly, the probability distribution graph is not like a normal distribution graph, which is what we would expect it to be.

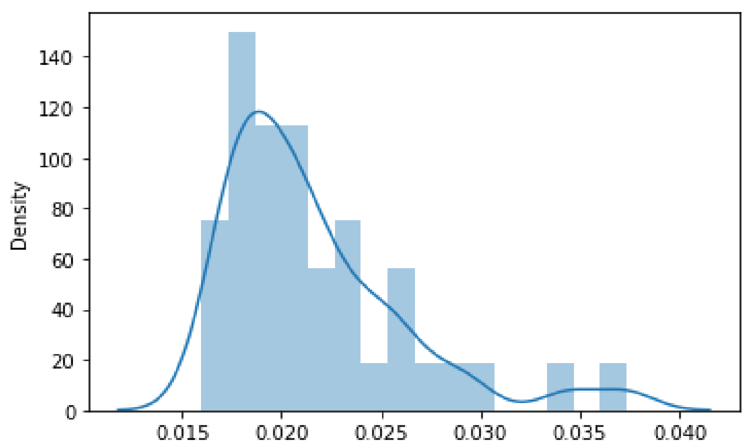


Figure 7: Johnson&Johnson stock

7 Possible Reasons for LSTM to Perform Poorly

We used a simple statistical prediction method to compare with LSTM during this research study. The statistical method uses today's price as a prediction of tomorrow's price. In fact this produces better results (!) on the mean error value compared to LSTM.

It is important to note that LSTM is not even comparable to the fool forecasting method that always assumes that the price will not rise for the next day. Specifically, the LSTM's error value for predicting stock prices for the following day is on average approximately 2.5%, whereas the same error for the fool forecasting method is on average only about 1.5%.

The possible reasons for LSTM to perform poorly are:

1. Too much data to learn, and much of the data values are far away from the future prices that LSTM needed to predict.

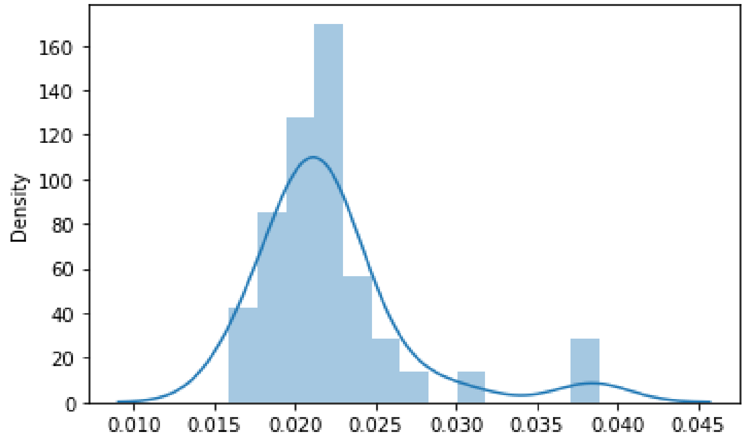


Figure 8: BP stock

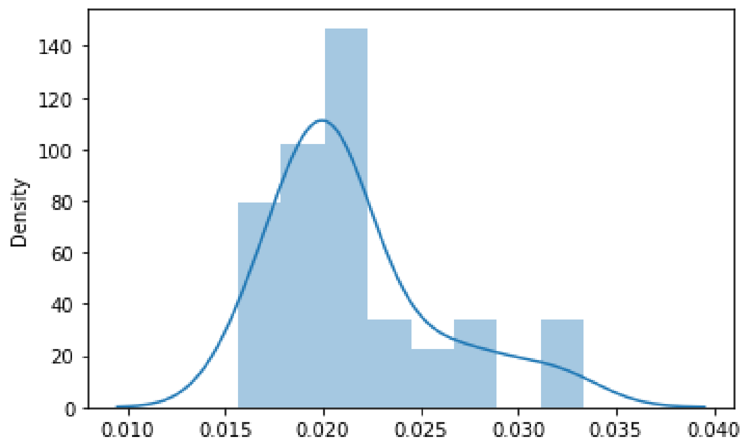


Figure 9: Pfizer stock

2. The stock market is in general nearly impossible to predict.

7.1 Correlation Analysis

LSTM in our study used 60 days of historical data to predict the future prices, and we believe that some days that are far away from today have little to no influence on today's price. To prove this, we did a correlation study for some stocks.

While p (the correlation significance level) is close to 0, the closer x (Pearson product moment correlation coefficient) is to 1, the greater the correlation. As a result, the correlation coefficient decreases with time from the current day. Not just for COP stock, but for ENB and TTE stocks as well. The x's in the correlations with today's price for ENB stock for three consecutive days are 0.998, 0.997, 0.995. The x's in the correlations with today's price for ENB stock for three consecutive days are 0.997, 0.994, 0.991.

7.2 LSTM Performance After Special Case Classification

Another reason for LSTM not performing well could be because the LSTM model has to learn too many cases of different data. LSTM might achieve a better prediction result if it only learns cases that are closely related to the trends that it needs to predict. We classified stocks into different cases and built separate LSTM models for those cases.

The classification is based on the analysis of 60 days of data which includes whether it is smoothly changing or not, whether it is increasing or not, and whether the last day is the extreme value or not being ahead of the data that we aim to predict.

We designed a scale (0 to 4) and two characters (T and F) to help analyze the results visually. The format of the x-axis consists of the scale value followed by two characters that represent the below-mentioned information:
0: greatly decreased during the 60 days before prediction
1: modestly decreased during the 60 days before prediction
2: slightly decreased or increased during the 60days before prediction
3: modestly increased during the 60 days before prediction
4: greatly increased during the 60 days before prediction

The first character:
T: the price is volatile every day
F: the price is steady every day

The second character:
Null: the sample is too small to classify
T: the last day is the extreme value of the 60-day range
F: the last day is not the extreme value of the 60-day range

The following is the number of different cases for Johnson&Johnson stock (Figure 10), BP company stock (Figure 11), and PFIZER company stock (Figure 12) after classifying the stock data into different cases.

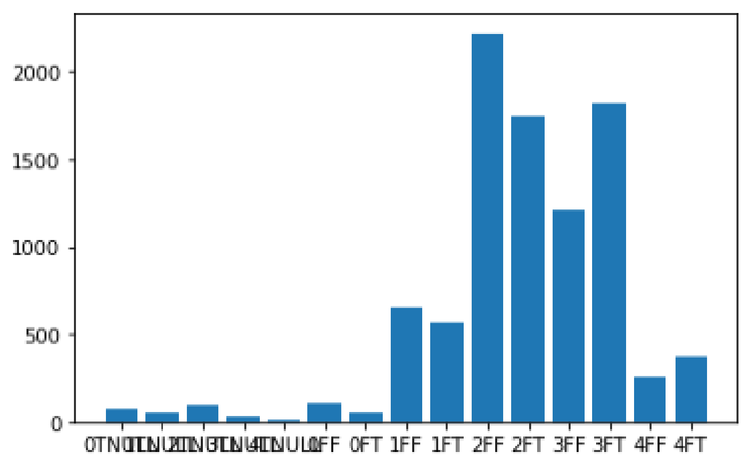


Figure 10: Johnson&Johnson stock

After developing different LSTM models for each case, we calculated the percentage of correctly predicting the stock prices as shown in Figures 13, 14, and 15.

The result of correct prediction is about 60%, much greater than the traditional LSTM model.

8 Conclusion and Future work

According to our experiments and analysis, we drew the conclusion that the traditional LSTM model is not capable of predicting stock prices accurately. With multiple model improvements applied to the traditional LSTM model, it can only slightly improve the accuracy of the prediction.

For future work, we recommend applying BERT (news analysis) [8] as well as PCA [14] (selecting the most influential input parameters) in addition to the traditional LSTM, and combining statistical analysis with LSTM.

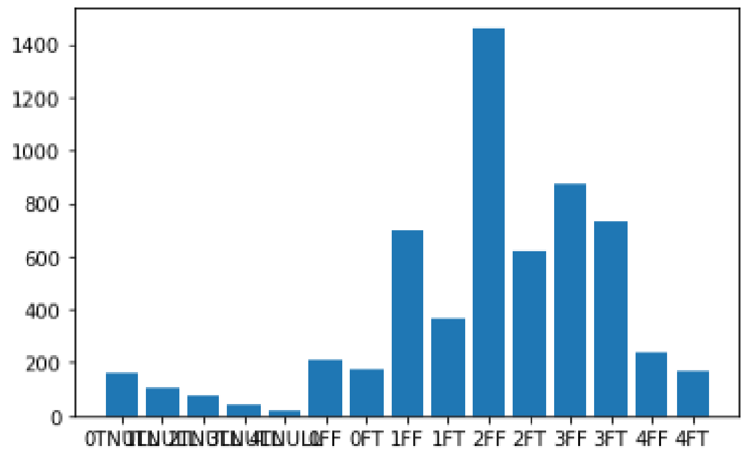


Figure 11: BP stock

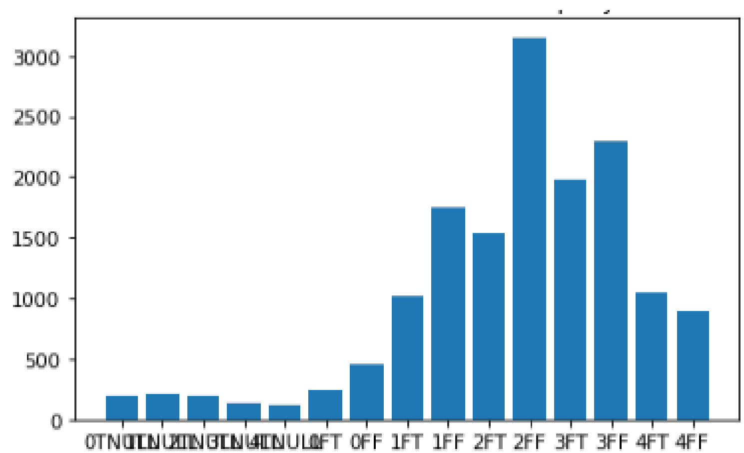


Figure 12: Pfizer stock

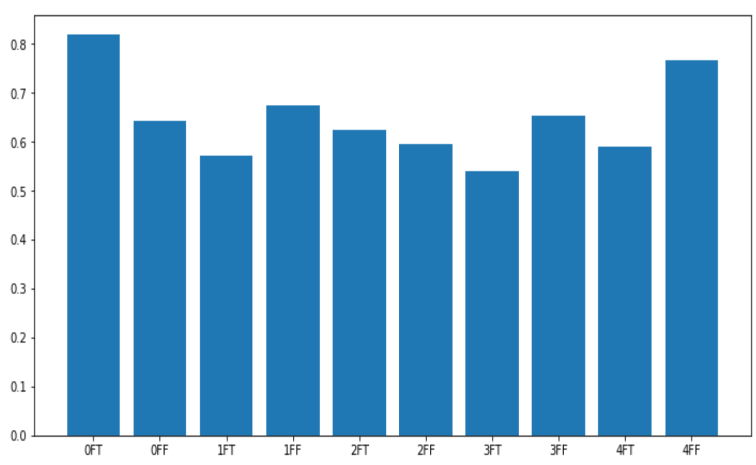


Figure 13: Johnson&Johnson stock

References

[1] Rolling-window analysis of time-series models - matlab & simulink. https://www.mathworks.com/help/econ/rolling-window-estimation-of-state-space-models.html.

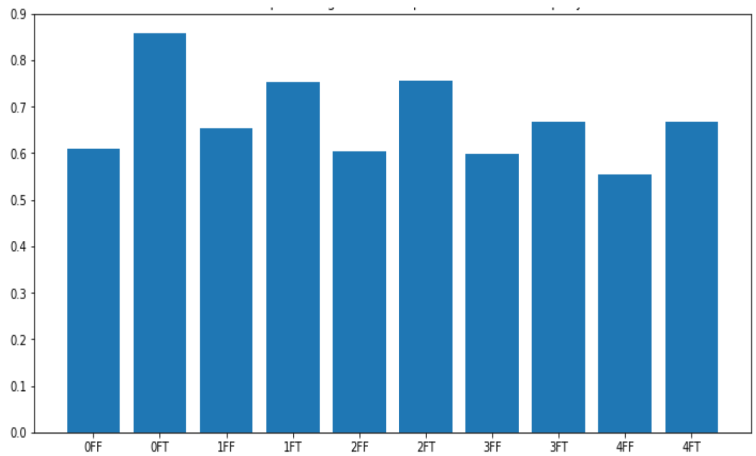


Figure 14: BP stock

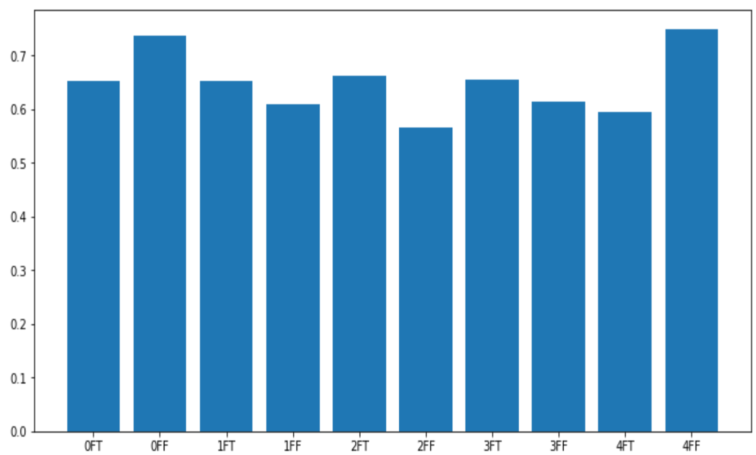


Figure 15: Pfizer stock

[2] What are recurrent neural networks? *IBM*.

[3] Weighted moving average (wma).). *Corporate Finance Institute*, 17 Feb. 2022.

[4] What do the yin and yang lines on the yin and yang line mean? *Shareholders Online School*, 19 Nov 2021.

[5] Moez Baccouche, Franck Mamalet, Christian Wolf, Christophe Garcia, and Atilla Baskurt. Sequential deep learning for human action recognition. In Albert Ali Salah and Bruno Lepri, editors, *Human Behavior Understanding*, pages 29–39, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg.

[6] Jason. Brownlee. A gentle introduction to exponential smoothing for time series forecasting in python. *Machine Learning Mastery*, 12 Apr. 2020.

[7] Jason Brownlee. A gentle introduction to long short-term memory networks by the experts. *Machine Learning Mastery*, 6 July 2021.

[8] Yingzhe Dong. Belt: A pipeline for stock price prediction using news. 12 Dec, 2021.

[9] Sepp Hochreiter, Martin Heusel, and Klaus Obermayer. Fast model-based protein homology detection without alignment. *Bioinformatics*, 23(14):1728–1736, 05 2007.

[10] M.S. Islam and E. Hossain. Foreign exchange currency rate prediction using a gru-lstm hybrid network. *Soft Computing Letters*, 3:100009, 2021.

[11] Hermann Mayer, Faustino Gomez, Daan Wierstra, Istvan Nagy, Alois Knoll, and Jurgen Schmidhuber. A system for robotic heart surgery that learns to tie knots using recurrent neural networks. In *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 543–548, 2006.

[12] Saul Dobilas. LSTM Recurrent Neural Networks — How to Teach a Network to Remember the Past. https://towardsdatascience.com/lstm-recurrent-neural-networks-how-to-teach-a-network-to-remember-the-past-55e54c2ff22e.

[13] et al. Tao, Lv. K-line patterns' predictive power analysis using the methods of similarity match and clustering.). *Mathematical Problems in Engineering*, 22 May 2017.

[14] HU Yu-wen. Stock forecast based on optimized lstm model. *College of Statistics, Jiangxi University of Finance and Economics, Nanchang 330013,China*, Nov. 2004.