Development of a LSTM Neural Network to Predict Closing Stock Price for Apple Inc.

Sanchit S. Shah

Data Science and AI Academy, North Carolina State University

DSC 412: Exploring Machine Learning

Professor Trenton Embry

November 16, 2024

Abstract

In today's day and age, AI and Machine Learning are seen as the goal of the technology industry, being implemented into almost every product and service being provided to consumers. Now, banks, trading firms, and fund management organizations are all seeking to implement such technologies into their workflow to take advantage of the significant amounts of available data available and maximize overall productivity [1]. However, a novel concept that has yet to be explored is whether analysts can make valuable use of the available data to make projections regarding securities which they are tracking. Therefore, this project attempts to develop a deep learning LSTM model to predict closing stock price for Apple Inc (NYSE: AAPL), a multinational technology company known for its consumer electronics and software.

Background

It is important to be able to distinguish some terms which will be used through this process. First, Artificial Intelligence (AI) is a form of intelligence capable of being expressed by machines/computing systems. Furthermore, Machine Learning (ML) is a subset of AI which allows a model to learn and improve over time based on provided data. Deep Learning (DL) is a type of ML which uses neural networks (like how the human brain's neurons are wired) to teach computers how to make decisions based on vast quantities of data. Furthermore, since this project is in the financial domain, some ideas about the stock market are also important to know. The stock market is where individuals can buy and sell shares of stock for a unit price (which fluctuates over time due to market volume or the number of transactions, earnings, and various other externalities). Apple Inc. (NYSE: AAPL) is a publicly traded company whose latest stock price as of November 16th, 2024, is $225.00. Finally, some financial terms which will be referenced throughout the project are Relative Strength Index (RSI), Volume Moving Average

(VMA), Exponential Averages (EA), Moving Averages (MA), and Moving Average Convergence/Divergence (MACD). RSI is a technical indicator which uses a momentum oscillator to calculate the average gains divided by the average losses over a 14-day period [4]. Volume Moving Average (VMA) is the average number of transactions (buy and sell) over a 20-day period [3]. The Exponential and Moving Averages show how the data fluctuates over time, respectively. MACD is another technical indicator which uses an oscillator function to identify buy/sell points. MACD is calculated by subtracting the 26-day exponential moving average from the 12-day exponential moving average [5].

Data Analysis

The exploratory data analysis was conducted on a CSV file containing the following pieces of data on Apple's stock price from 01/01/2010 – 08/30/2024: Open, High, Low, Close, Adjusted Close (after hours), Volume. First, the Open Price and Close Price over time were analyzed. As seen in Figure 1 below, the Open Price and Close Price over time relatively follow the same trend, starting around $7 in 2010 and peaking close to $225 in late 2024.

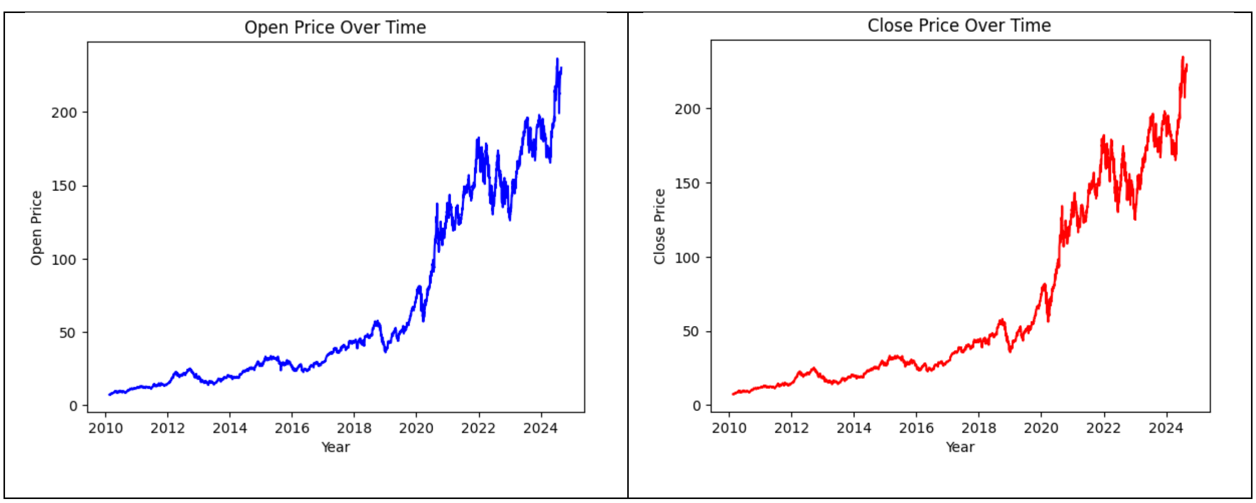
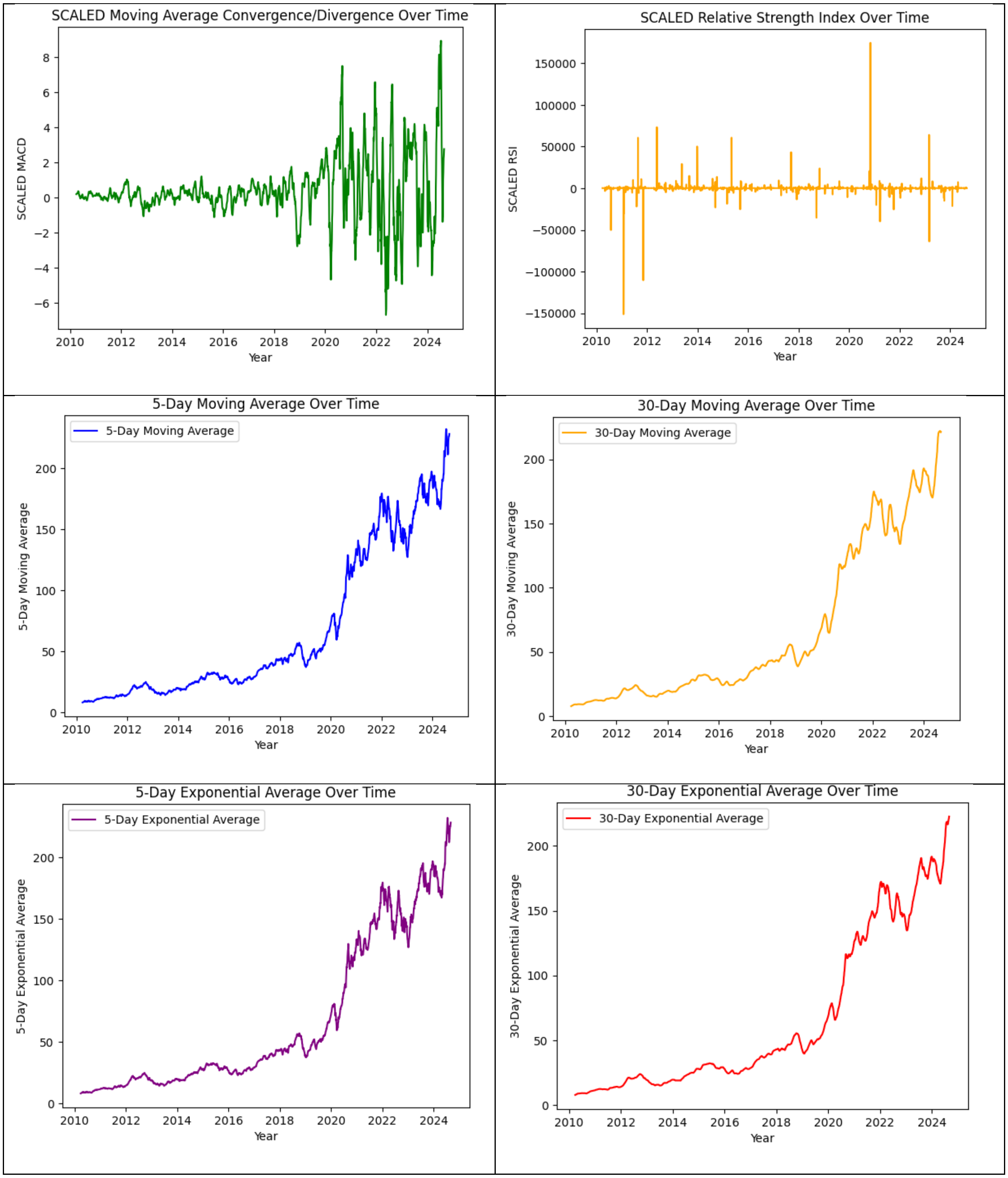


Figure 1: Open and Close Prices Over Time

After this, the financial metrics discussed earlier (RSI, VMA, EA, MA, MACD) were analyzed. A positive RSI indicates that a stock may be overbought or overvalued, whereas a negative RSI indicates that a stock may be oversold or undervalued, while 0 indicates a balance between bullish and bearish positions [4]. In the case of Apple's stock, the RSI is relatively close to 0 with some extreme peaks such as in 2011 and 2012 when the positions of -150,000 and approximately -100,000 were reached. Furthermore, there was another extreme peak in 2021 which exceeded 150,000. There are some peaks close to 50,000 which appear over time due to various externalities. Furthermore, the EA and MA over time followed a similar trend compared to the Open and Close prices indicating long-term fluctuations. An MACD of 0 indicates that the stock is relatively stable while an MACD that differs from 0 indicates volatility [5]. The MACD is relatively close to 0 in the early years, however, as it nears 2020 and beyond, there are significant fluctuations which are consistent with the aggressive growth of Apple's stock price near the 2020 and 2021 point. This indicates a significant number of short-term fluctuations which are not captured by the long-term chart. Finally, VMA helps put Apple's daily volume in context. Even though it may appear that the amount of people that are buying Apple stock is increasing (at an increasing rate as well) it is not nearly as high as it was back in 2010. All charts are displayed below in Figure 2. After this preliminary analysis, since this is a time series problem and there are lots of fluctuations in the data over the short-term and long-term, the model of preference was a LSTM (long short-term memory) Neural Network. This methodology will be explained in a future section.


SCALED Moving Average Convergence/Divergence Over Time
SCALED MACD
Year
SCALED Relative Strength Index Over Time
SCALED RSI
Year
5-Day Moving Average Over Time
5-Day Moving Average
5-Day Moving Average
Year
30-Day Moving Average Over Time
30-Day Moving Average
30-Day Moving Average
Year
5-Day Exponential Average Over Time
5-Day Exponential Average
5-Day Exponential Average
Year
30-Day Exponential Average Over Time
30-Day Exponential Average
30-Day Exponential Average
Year

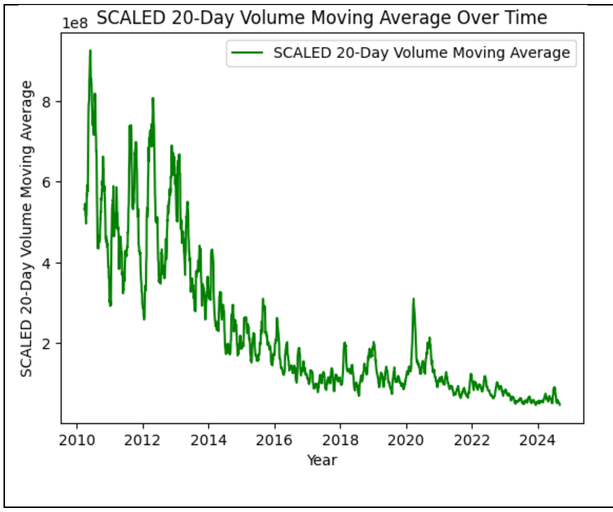


Figure 2: MACD, RSI, MA, EA, and VMA Over Time

Data Augmentation

Since this data came from a reputable source (extracted from Yahoo Finance), the data did not have to be cleaned and there were no NaN values to be removed. The data was not combined either as each column maintained its own identity. The existing data was used to calculate the new financial metrics, however. The only manipulation which was conducted was scaling the data using sklearn MinMaxScaler to make the values easier to work with in terms of the machine learning aspect of this project.

Model Selection

As indicated above, since this is a complex dataset which includes time series and long-term/short-term fluctuations in the data, it becomes apparent that the Long Short-Term Memory (LSTM) Neural Network is the best fit for working with this data. LSTMs are a type of recurrent neural network which is specialized to learn and predict sequential data based on historical information [6]. LSTM uses specialized gates/neurons to manage this memory and avoid the "vanishing gradient" problem in which the gradients used to update the weights in a neural network become extremely small during backpropagation [2].

Training Methodology

The model was first trained on a few base models including Linear Regression, SVR, and Random Forest since they are known to be useful when analyzing data fluctuations. The baseline comparison was the Linear Regression model's RMSE of 0.013. To train the deep learning model, the data was split into a training and testing set with 80% of the data going to model training and the other 20% going to model testing. Specifically considering date ranges, this involved an approximately 10-year period with training data going from 01/01/2012 to 10/08/2020 and testing data going from 10/09/2020 to 12/21/2022. The idea was to use the training data to then predict the stock price for the testing date range. The activation function used in the LSTM model was ReLU function which can be represented by f(x) = max(0, x). The ReLU function was used to prevent the "vanishing gradient" problem as described above. The optimizer used was 'adam,' the standard optimizer used with TensorFlow-based neural networks. Usually, 10-11 epochs are recommended for small quantities of data, however, considering the amount of data that was being trained, 50 epochs were run. However, around the 10^{th} to 11^{th} epoch mark, the model has around the same amount of loss as the last epoch. This model did not use hyperparameter tuning since the results did not seem to be skewed in a particular direction (by a specific input parameter). Furthermore, the model did not have significant overfitting or underfitting issues due to the very low loss of 0.011.

Results

Upon training the model on the training date range above, the model was saved in the .h5 file format to prevent having to train it repeatedly. Since this is time series data, the individual outputs do not matter; rather, the expected output is a chart showing the predicted stock price during the training dates above. In Figure 3 below, the output of the model is plotted, displaying the predicted stock price with the actual stock price for that period. Visually, the model does a pretty good job of following the general pattern of the actual stock price. One thing to note is that the dates scale was normalized (since it is difficult to work with dates even with Pandas Date-Time format). This is why the x-axis goes from 0 to 700 on time. This really indicates a start date of 10/09/2020 and runs until the end of the testing period. Even though it cannot be directly proven that the analysis and features implemented during the feature engineering stage of the project had a direct impact on such favorable results, it is believed that these features helped provide some shape to the model in terms of analyzing the short-term fluctuations.

Figure 3: Model results showing original and predicted price.

Future Work

Though the model did meet expectations, there are some areas of improvement. For example, the training time can be reduced by running fewer epochs so that more time can be spent with tuning and trying to better fit the model. Furthermore, different activation functions like sigmoid as well as different optimizers like BinaryCrossentropy can also be used since both are used regularly with time series data. There are two parts to future projects to try. First of all, using the model to predict a future stock price (for example, training the model up to a specific date and attempting to predict the genuine future stock price). This prediction would be compared with the closing price once that day arrives to determine the accuracy of the mode. Furthermore, the second part of this extension would be to train the model on other stocks like Google (NYSE: GOOG), Microsoft (NYSE: MSFT), and Amazon (NYSE: AMZN). One key lesson learned from this project is that the actual machine learning/model training part is a very small challenge compared to the data preprocessing, scaling/normalization, and feature engineering to enhance the model’s output.

Stakeholder Acknowledgements

The stakeholders for this project are a wide range of stakeholders. This includes various users in academia including students and professors who may want to conduct research using this model. This might also include industry professional and trading firms which wish to incorporate this model into an existing pipeline/framework to enhance and/or validate models that they may have in place. The model could have a negative effect in two primary ways. Firstly, if the model does not correctly predict prices, this will mislead stakeholders and potentially jeopardize decisions that they make. Secondly, if the model gets into the wrong hands, it may be misused which could negatively impact the market. Obviously, the current model is not mature enough to present this threat, however, on a larger scale, if the model were to be extremely precise across a wide range of stocks, then it would indicate such a threat. Stakeholders cannot necessarily “try” the model but once the model is (1) able to predict genuine future prices and (2) exhibit robustness across a wide variety of stocks, stakeholders would be able to attempt this to a stock and period of their choosing.

Conclusion

In this project, a LSTM Neural Network was developed to predict the stock price of Apple Inc. (NYSE: AAPL). After data preprocessing, analysis, augmentation, and engineering (feature implementation) the model was trained on data from 2012-2020 and then tested to predict price from 2020-2022. The model does a solid job of predicting stock prices as seen visually as well as a quantitative loss of 0.011. Finally, the limitations/shortcomings, future work, and stakeholder dynamics are addressed.

Citations

[1] Rakshit, Sandip, et al. "Exploratory Review of Applications of Machine Learning in Finance Sector." *Advances in Data Science and Management*, 2022, pp. 119–125, https://doi.org/10.1007/978-981-16-5685-9_12.

[2] Moghar, Adil, and Mhamed Hamiche. "Stock Market Prediction Using LSTM Recurrent Neural Network." *Procedia Computer Science*, vol. 170, 2020, pp. 1168–1173, https://doi.org/10.1016/j.procs.2020.03.049.

[3] Khand, Salma, et al. "The Performance of Exponential Moving Average, Moving Average Convergence-Divergence, Relative Strength Index and Momentum Trading Rules in the Pakistan Stock Market." *Indian Journal of Science and Technology*, vol. 12, no. 26, 1 July 2019, pp. 1–22, https://doi.org/10.17485/ijst/2019/v12i26/145117.

[4] Fidelity. "What Is RSI? - Relative Strength Index - Fidelity." *Fidelity.com*, www.fidelity.com/learning-center/trading-investing/technical-analysis/technical-indicator-guide/RSI.

[5] Fernando, Jason. "Moving Average Convergence Divergence – MACD Definition." *Investopedia*, 16 Sept. 2024, www.investopedia.com/terms/m/macd.asp.

[6] Chugh, Aakarsha. "Deep Learning | Introduction to Long Short Term Memory." *GeeksforGeeks*, 16 Jan. 2019, www.geeksforgeeks.org/deep-learning-introduction-to-long-short-term-memory/.