# Stock Market Analysis and Different Methods of Predicting Stock Prices


| WEILIN ZHOU | MINGZHU YU | CHUN-CHIEH TSENG | WENTAO CHEN | ZHONGCHENG TU |
|---|---|---|---|---|
| wz2563 | my2691 | ct3057 | wc2768 | zt2286 |
| Department of Statistics | Department of Statistics | Department of Statistics | Department of Statistics | Department of Statistics |
| Columbia University | Columbia University | Columbia University | Columbia University | Columbia University |


## 1. Introduction

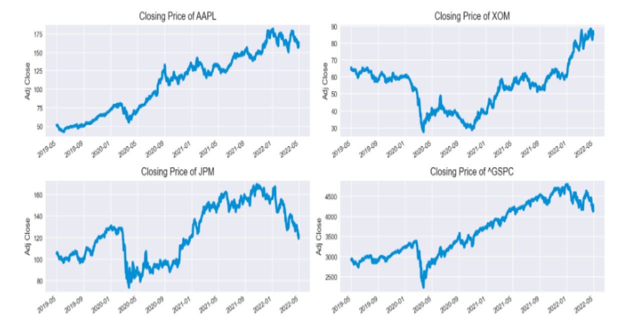
The stock market can be seen as an open marketplace in the financial industry. People, including individual and institutional investors, are trying to bid for the best price relying on their most updated information. The basic trading strategy usually divided into two different methods. One is fundamental analysis, which mainly focuses on analyzing economic and financial factors to measure a security's intrinsic value. Another method is technical/quantitative analysis, which is used to predict the stock's future price based on its past performance.

With enormous pieces of information in today's fast-changing landscape, applying mathematical computations and numbers to identify potential trading opportunities is becoming more and more significant. Deep learning model based on neural networks has attracted great attention to develop trading strategies, which may greatly improve the stock return forecasts compared to the basic linear model. The main goal of this project is to investigate probable algorithms, e.g ARIMA model, LSTM-networks, and GRU model to predict the future performance of the stock market more efficiently.

## 2. Performance of the stock

We will be looking at data from the stock market and we mainly pick up specific stock from tech/oil/financial industry and S&P 500. We also calculate the moving average of each stock and visualize them. The moving average can be used to help traders identify buying and selling opportunities and make decisions.

**Closing Price of Each Stock**



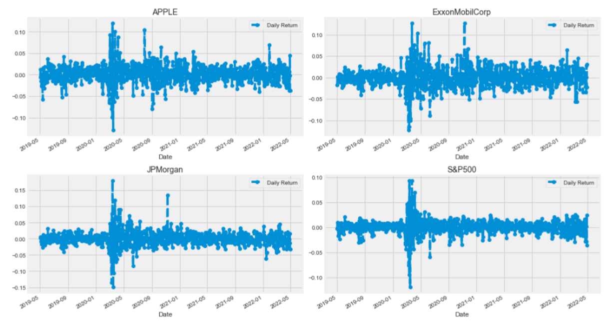

***Source：Yahoo Finance***

**Moving Average of Each Stock**


***Source：Yahoo Finance***

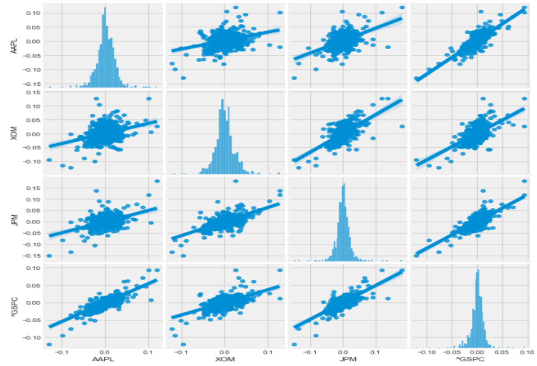
We then calculate the daily past return of each stock and plot them and compare the daily percentage return of two stocks to check how they are correlated.

**Return Performance of Each Stock**




***Source: Yahoo Finance***

**Correlation Matrix**




***Source: Yahoo Finance***

## 3. Dataset and Preparation

We pick up Apple and S&P 500 stock as our dataset and the 'Adj Close' will be the only numerical values we keep. The control window of data we utilized is from 01/01/2018 to 29/04/2021 and we separate the train and test set at 01/01/2022.

**Train Dataset**

| Date | Open | High | Low | Close | Adj Close | Volume |
|---|---|---|---|---|---|---|
| **2018-01-02** | 86.129997 | 86.309998 | 85.500000 | 85.949997 | 81.530235 | 22483800 |
| **2018-01-03** | 86.059998 | 86.510002 | 85.970001 | 86.349998 | 81.909676 | 26061400 |
| **2018-01-04** | 86.589996 | 87.660004 | 86.570000 | 87.110001 | 82.630585 | 21912000 |
| **2018-01-05** | 87.660004 | 88.410004 | 87.430000 | 88.190002 | 83.655037 | 23407100 |
| **2018-01-08** | 88.199997 | 88.580002 | 87.599998 | 88.279999 | 83.740425 | 22113000 |
| ... | ... | ... | ... | ... | ... | ... |
| **2021-12-23** | 332.750000 | 336.390015 | 332.730011 | 334.690002 | 333.999390 | 19617800 |
| **2021-12-27** | 335.459991 | 342.480011 | 335.429993 | 342.450012 | 341.743378 | 19947000 |
| **2021-12-28** | 343.149994 | 343.809998 | 340.320007 | 341.250000 | 340.545837 | 15661500 |
| **2021-12-29** | 341.299988 | 344.299988 | 339.679993 | 341.950012 | 341.244415 | 15042000 |
| **2021-12-30** | 341.910004 | 343.130005 | 338.820007 | 339.320007 | 338.619843 | 15994500 |

1007 rows × 6 columns

**Validation Dataset**

| Date | Open | High | Low | Close | Adj Close | Volume |
|---|---|---|---|---|---|---|
| **2022-01-03** | 335.350006 | 338.000000 | 329.779999 | 334.750000 | 334.059265 | 28865100 |
| **2022-01-04** | 334.829987 | 335.200012 | 326.119995 | 329.010010 | 328.331116 | 32674300 |
| **2022-01-05** | 325.859985 | 326.070007 | 315.980011 | 316.380005 | 315.727173 | 40054300 |
| **2022-01-06** | 313.149994 | 318.700012 | 311.489990 | 313.880005 | 313.232330 | 39646100 |
| **2022-01-07** | 314.149994 | 316.500000 | 310.089996 | 314.040009 | 313.391998 | 32720000 |
| ... | ... | ... | ... | ... | ... | ... |
| **2022-04-22** | 281.679993 | 283.200012 | 273.380005 | 274.029999 | 274.029999 | 29405800 |
| **2022-04-25** | 273.290009 | 281.109985 | 270.769989 | 280.720001 | 280.720001 | 35678900 |
| **2022-04-26** | 277.500000 | 278.359985 | 270.000000 | 270.220001 | 270.220001 | 46518400 |
| **2022-04-27** | 282.100006 | 290.970001 | 279.160004 | 283.220001 | 283.220001 | 63477700 |
| **2022-04-28** | 285.190002 | 290.980011 | 281.459991 | 289.630005 | 289.630005 | 33646600 |

81 rows × 6 columns

## 4. Methods and Experiments

We built 4 models to predict future price movements and compare the results between them. They are 1) Linear Regression Model; 2) Time Series Model – ARIMA; 3) RNN with LSTM Model; 4) RNN with GRU Model.

### 4.1 Linear Regression Model

Linear regression model helps to identify the relationship between a dependent variable and one or more independent variables. The basic concept behind the model is:

$$y = \beta_0 + \beta_1 X + \epsilon$$

We apply Exponential moving average (EMA) over a 20-day period as our independent variable to describe the price movements in the past. Our model trained on 856 training samples and generated predicted values based on these training results.

$$EMA = \frac{2}{n+1} * (Close - Previous\ EMA) + Previous\ EMA$$

We check the accuracy of our model fitting the data by examining the model coefficients and mean absolute error (MAE) and coefficients of determinations $R^2$,where:

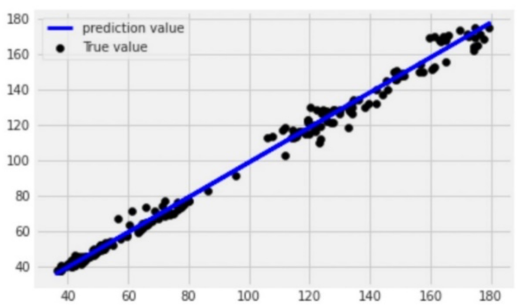
$$MAE = \frac{\sum_{i=1}^{n} |y_i - x_i|}{n} \ \&\ R^2 = 1 - \frac{RSS\ (sum\ of\ squares\ of\ residuals)}{TSS\ (total\ sum\ of\ squares)}$$

Table 1: Summary of Linear Regression Model

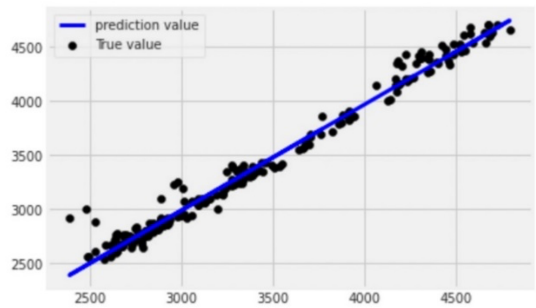
| | APLLE | S&P500 |
|---|---|---|
| **Model Coefficients** | 0.9868 | 0.9804 |
| **Mean Absolute Error** | 2.7316 | 63.1642 |
| **Coefficient of Determination** | 0.9924 | 0.9794 |

We assume that a lower MAE value suggests a good fit and the closer our coefficient of the correlation value is to 1.0 the better. Here are plots of our predicted values after conducting liner regression:

**'APPLE' Performance**




**'S&P500' Performance**




From the plot we can see that the trained model has pictured a general prediction method for stock prices and our model seems fit the data well with both $R^2$ are close to 1, though MAEs is slightly high.

## 4.2 Time Series Model – ARIMA

Time-series forecasting is widely used for non-stationary data, whose statistical properties are not constant over time. AutoRegressive Integrated Moving Average (ARIMA) Model is widely used to make forecast for time-series data by converting it to show stationary patterns. ARIMA model usually need three parameters to be specified, which are:

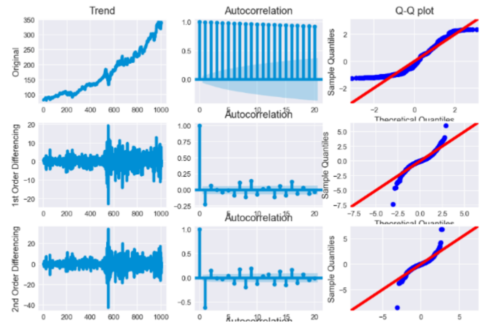
**p:** the number of lag observations

**d:** the degree of differencing

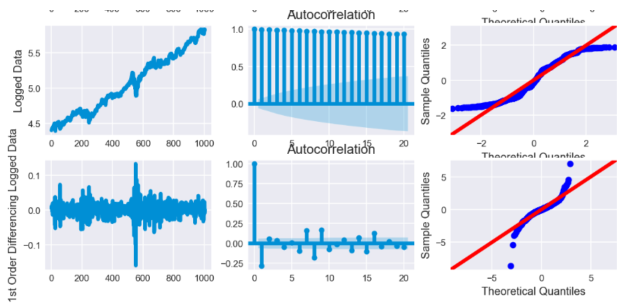
**q:** the size of the moving average window

We choose 1st Differencing/2nd Differencing/logarithm to make our data more stationary. Here are the data performance after we make the transformation:

**Original Data with Differencing**

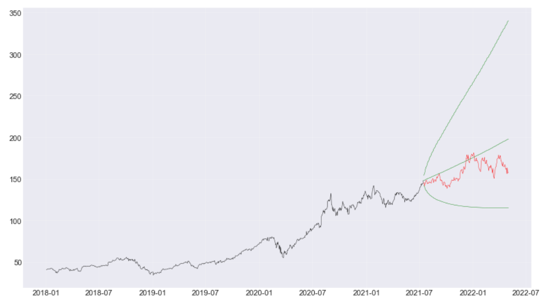



**Logged Data with Differencing**




Autocorrelation refers to the degree of correlation between same variables between two successive time intervals. From the plot we can see that the first order logged data has a small autocorrelation and its distribution is closer to normality. This shows a good stationary pattern. Therefore, we choose first-order logged data to fit ARIMA model.

**Prediction of AAPL**



**Prediction of S&P500**


Table 2: Summary of Time Series Model

| | APLLE | S&P500 |
|---|---|---|
| **MSE** | 289.68 | 34934.59 |

We split the dataset into train and test sets and use train sets to fit the model, and generate a prediction for each element on the test set. From the plot and summary, we can see that the prediction of Apple stock price seems reliable and with higher prediction accuracy compared to the prediction of S&P500. This may be explained by the fact that the performance of S&P500 are more easily influence by the macro environment. The sudden conflict between Russian & Ukraine and the new that Fed will hike rates made great impact on the stock market, which will make prediction imprecisely.

### 4.3 Recurrent Neural Network Models

As financial institutions begin to embrace artificial intelligence, machine learning is increasingly utilized to help make trading decisions. Recurrent Neural Networks (RNN) are a class of neural networks specifically designed to handle sequential data. Here, we will introduce two models in the field of recurrent neural networks to make predictions.

### 4.3.1 LSTM – Long Short Term Memory Model

One method for predicting stock prices is using a Long Short-Term Memory neural network(LSTM) for time series forecasting. LSTMs are an improved version of recurrent neural networks (RNNs). LSTMs are a type of RNN that remember information over long periods, making them better suited for predicting stock prices.

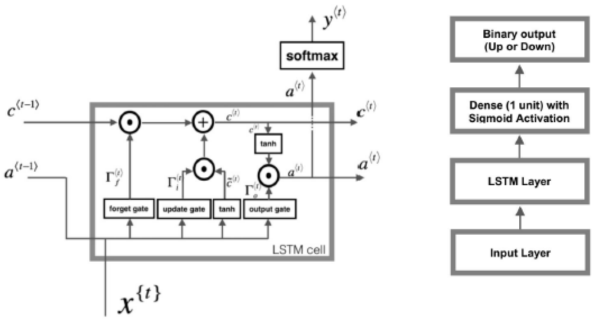
The key to LSTM is the cell state, the horizontal line that runs through the top of the diagram. With only a few linear interaction, it's easy for information to simply flow through. However, the LSTM can remove or add information to the cell state. Carefully controlled by three gates. Input gate, forget gate and output gate.

**The Input Gate:** add information to the cell state

**The Forget Gate:** remove the information that is no longer required

**The Output Gate:** select the information to be shown as output

**LSTM unit Network**

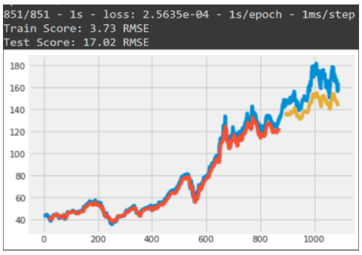



**Gate Structure**

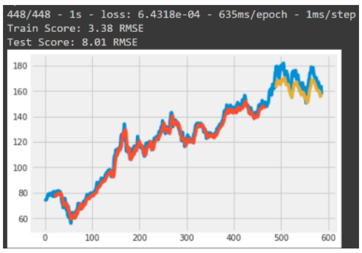
| | | |
|---|---|---|
| Input Gate | Is cell updated? | $C_t = \tanh(W_c * [h_{t-1}, x_t] + b_c)$ |
| Forget Gate | Is memory set to 0? | $f_t = \sigma(W_f * [h_{t-1}, x_t] + b_f)$ |
| Output Gate | Is current info visible? | $O_t = \sigma(W_\sigma * [h_{t-1}, x_t] + b_o)$ |

From the figure shown below we can see that when the epochs reaches to 100, we can clearly see that the model starts to overfit the data heavily for larger epochs. The model does not learn any pattern or context instead is just memorize the training data we have. This will lead to that out validation loss will increase for longer epochs. The validation error is always larger than the training error which we try to avoid. Then our future action could be finding the best epochs by choosing optimal number of epochs.

**851 Days APPLE Closing Price**

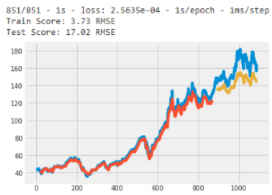


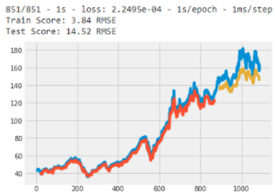

**448 Days APPLE Closing Price**

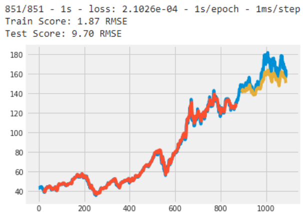


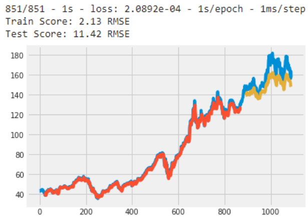

We applied MSE to test our models' accuracy and errors in the predictive models. We tried different amount of dataset used in predicting same stock and same epochs and we found that training with small dataset yields better MSE.

**APPLE Stock Performance Starting 2018**

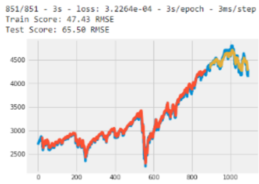


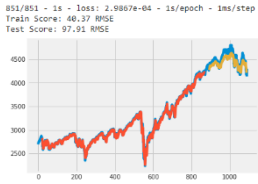



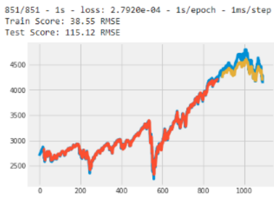




**S&P500 Performance Starting 2018**



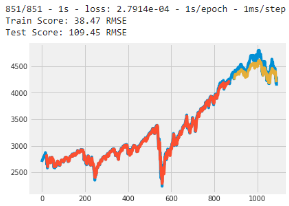




Table 1: Summary of LSTM Model

| MSE/RMSE | APPLE | S&P500 |
|---|---|---|
| **Epochs = 25** | 289.68/17.02 | 4290.25/65.5 |
| **Epochs = 50** | 210.83/14.52 | 9586.37/97.91 |
| **Epochs = 75** | 94.09/9.7 | 13252.61/115.21 |
| **Epochs = 100** | 130.42/11.42 | 11979.3/109.45 |

One issue occurred when training the neural network is overfitting. When the number of epochs used to train exceeds some extent will make model unable to perform well on a new dataset. Higher epochs will yields to the fact that the accuracy of training dataset is high but gives lower accuracy on the test set.

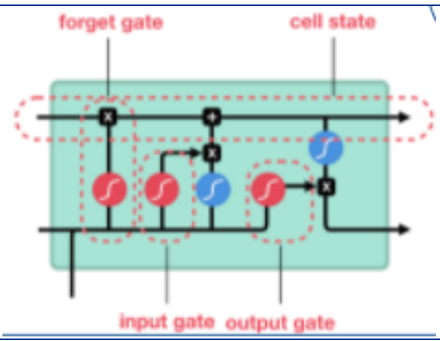
Overall we can see that training with less data and suitable epochs can improve our testing result and at the same time allow us to have better forecasting and prediction values.

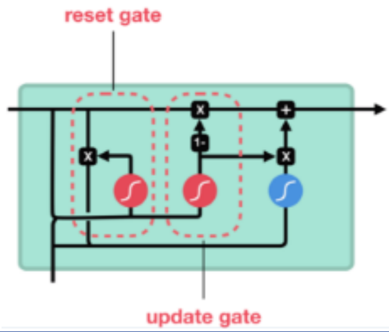
### 4.3.2 Gated Recurrent Unit (GRU) Model

The GRU is the newer generation of Recurrent Neural networks and is pretty similar to an LSTM. GRU's got rid of the cell state and used the hidden state to transfer information. It only has two gates, a reset gate and update gate. (Michael, 2018) The update gate helps model to decide what past information can pass as the new information. The reset gate is another gate is used to decide how much past information to forget. To set our model, we add 4 hidden layers. For each layer, we drop out 20% nodes to stabilize the GRU Model. We also set 20 days as a window to predict the price in the next window.

**LSTM Network**

**GRU Network**




From the graph above we can see that GRU's has fewer tensor operations compared with LSTM, which is the reason why GRU's can train data faster than LSTM's. However, we can't define that GRU is the better model than LSTM, there isn't a clear winner which one is better. It depends on the certain case that people use.

We apply GRU model on the price forecast and it can be seen that the accuracy of our prediction is high. It can predict the peaks precisely, but it also has some constraints. It will appear peaks latter than real time.

**Apple Stock Price Prediction**


**S&P500 Stock Price Prediction**

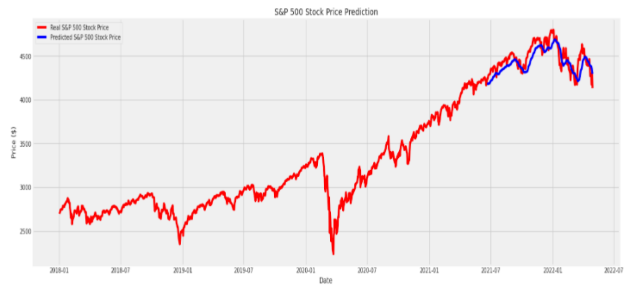



Table 1: Summary of GRU Model

| | APPLE | S&P500 |
|---|---|---|
| **MSE** | 37.3876 | 11512.0608 |
| **MAE** | 4.7938 | 90.5938 |
| **RMSE** | 6.1145 | 107.2942 |

## 5. Conclusion

This project established a forecasting framework to predict stocks' prices. We proposed, developed, trained, tested four models: Linear Regression, ARIMA, LSTM, GRU to make trading strategies. Based on our analysis, we find that LSTM and GRU Model are more preferred. Linear regression model needs specific assumption and might not be suitable prediction when suffering short-term volatility in stocks. Time Series model will yield large error if external events happen suddenly and may not provide a precise prediction in a long-term period. LSTM and GRU model yield better results. It has been never easy to invest a set of assets, the abnormally of financial market does not allow simple models to predict future assets with higher accuracy. Markets are affected by many factors such as political, industrial development, market news, social media and economic environment. Potential enrichment for this project may include:

- Include some non-technical features to make predictions
- Having longer time horizons when selecting dataset, which may enhance the accuracy of models
- Set specific investing targets and train the model to achieve it
- Exploring more new models.

Appendix:

*Choose optimal number of epochs to train a neural network in Keras*. GeeksforGeeks. (2020, June 8). Retrieved May 11, 2022, from https://www.geeksforgeeks.org/choose-optimal-number-of-epochs-to-train-a-neural-network-in-keras/

Kumar, D. V. (2021, April 5). *Hands-on guide to LSTM recurrent neural network for stock market prediction*. Analytics India Magazine. Retrieved May 11, 2022, from https://analyticsindiamag.com/hands-on-guide-to-lstm-recurrent-neural-network-for-stock-market-prediction/

*L1 penalty and sparsity in logistic regression*. scikit. (n.d.). Retrieved May 5, 2022, from https://scikit-learn.org/stable/auto_examples/linear_model/plot_logistic_l1_l2_sparsity.html

Moghar, A., & Hamiche, M. (2020, April 14). *Stock market prediction using LSTM recurrent neural network*. Procedia Computer Science. Retrieved May 11, 2022, from https://www.sciencedirect.com/science/article/pii/S1877050920304865

Phi, M. (2020, June 28). *Illustrated guide to LSTM's and GRU's: A step by step explanation*. Medium. Retrieved May 11, 2022, from https://towardsdatascience.com/illustrated-guide-to-lstms-and-gru-s-a-step-by-step-explanation-44e9eb85bf21

Ridhijhamb. (2021, September 2). *Stock price prediction: Multivariate time series*. Kaggle. Retrieved May 11, 2022, from https://www.kaggle.com/code/ridhijhamb/stock-price-prediction-multivariate-time-series/notebook

*Using a keras long short-term memory (LSTM) model to predict stock prices*. KDnuggets. (n.d.). Retrieved May 11, 2022, from https://www.kdnuggets.com/2018/11/keras-long-short-term-memory-lstm-model-predict-stock-prices.html

West, Z. (2021, December 3). *Predicting stock prices with linear regression in python*. αlphαrithms. Retrieved May 11, 2022, from https://www.alpharithms.com/predicting-stock-prices-with-linear-regression-214618/