Forecasting S&P 500 Stock Prices with LSTM

Vivian Xia

Northwestern University

MSDS458: Artificial Intelligence & Deep Learning

Syamala Srinivasan

March 8, 2022

Abstract

Stock market data is a timeseries that has periodic cycles and patterns throughout time. With these temporal patterns, the stock market prices can be forecasted. LSTM model architecture will be used to capture the temporal patterns and forecast the next day closing prices. The LSTM model is able to remember the temporal patterns from past sequences to be used as context in predicting the next closing prices.

The model will use the S&P 500 historical data from February 28, 2017 to February 28, 2022 from Yahoo! Finance to forecast the closing prices for the next day. S&P 500 is a stock market index that factors in the top 500 companies in the United States. Because it tracks the performance of those companies, the index provides a general overview of the United States stock market. To factor in the COVID-19 pandemic effect on the stock market, some experiments will use COVID-19 probable new cases per day historical data from Centers for Disease Control and Prevention. The experiments will include LSTM models with a lag of 30 and 15 days as well as models with one input, closing prices, and two inputs, closing prices and probable new cases. The models and their performance will be analyzed and visualized to evaluate the best model to forecast S&P 500 closing prices.

Introduction

The Standard and Poor's 500 or S&P 500 is a stock market index featuring the top 500 publicly traded companies in the United States. These companies include a range of industries such as health care, information technology, finance, energy, etc., providing one of the best gauges of U.S. equity and the stock market (Kenton, 2022). Therefore, predicting the S&P 500 Index closing prices will provide a general overview of closing prices and behavior of most large U.S. stocks.

The long short term or LSTM model was chosen to forecast the closing prices of the next day with the use of S&P 500 historical data. Historical data is temporal data containing temporal correlations or patterns that are relevant in the prediction of the future prices. LSTM models have a model structure that can capture these temporal correlations and memorizing it to be used as historical context in its predictions.

The goal is to use LSTM models to build a S&P 500 closing stock price forecaster. The S&P 500 five-year historical data from February 28th, 2017 to February 28th, 2022 is downloaded from Yahoo! Finance. Because COVID-19 impacted the stock market, the probable new cases per day from the Centers for Disease Control and Prevention is also used in a few experiments (CDC Case Task Force, 2022). The input and output data are preprocessed into time step lags that to be inputted into the model. Each model and its performance will be analyzed to recommend the best S&P 500 forecaster model among the models experimented.

Literature review

For stock market prediction, there are many ways to model the future stock prices. One study compares the performance of artificial neural networks, recurrent neural networks, and LSTM in stock market value prediction. The experiments used the same dataset for all models. The dataset included ten features with one target, which were rearranged to predict the target value for different lags. Using mean absolute error and mean squared error to measure performance, LSTM performed the best and had the most accurate results for the all the experimented time lags while ANN had the worst performance. This study noted that the only notable disadvantage of the LSTM model was that the runtime took longer than the other models (Nabipour et al., 2020).

The stock market was impacted with the emergence of COVID-19, so the models pre-pandemic may not be as accurate in their predictions. With the outbreak of Severe Acute Respiratory Syndrome, SARS, the Dow Jones index reduced by 15% due to the associated fear and uncertainty of the virus before it returned to previous levels and even hit new highs. Because the pandemic affects multiple sectors, there can be a large-scale effect on stock prices. To model the predicted stock values, this study takes into consideration the sentiment from Yahoo Finance Message Board on COVID-19 in investment decisions. By factoring in COVID-19 sentiments, the proposed LBL-LSTM model performed better than the pre-pandemic existing algorithms by 4% ((Gurav & Kotrappa, 2020).

Methods

Packages

The package yfinance is installed and imported to download the historical stock price data for S&P 500. The os library is imported to save the models. The packaging library is used to check that the notebook has the required tensorflow and keras version for it to run. Matplotlib and seaborn are used to visualize and explore the data as well as model performance. The library pandas and numpy are imported to format the data. Sklearn is used to scale the values, split the data into training, validation, and test set, and measure the performance of the models. The tensorflow library is imported to build the model.

Data Wrangling

Experiment A: Input – Closing Price, Time Step – 30

The data consists of historical stock prices of S&P 500 from February 28, 2017 to February 28, 2022. This 5-year historical data is downloaded from Yahoo! Finance. The closing

stock prices of each day will be used to predict the closing stock price of the next day, so only the "Close" column is taken into consideration when exploring and building the model.

The closing stock price data for the five years is visualized. The resulting graph in Figure 2 shows some weak yearly periodicity where there are evident dips in price right before 2019, a few months into 2020, and a smaller but still evident dip a few months before 2021. Taking a closer look, a graph of the closing prices from the last year of January 1, 2021 to January 1, 2022 is visualized in Figure 3. The visual shows monthly periodicity patterns. Overall, the stock prices increase during that year, but there are some steeper dips around twice a month (Chollet, 2021). The data's descriptive statistics show that the data does not seem to have outliers.

The data is preprocessed by scaling the values between 0 and 1. Scaling it will help the model train faster due to the smaller differences between values. The data is then split into training, validation, and test sets where the training set has 0.70, validation has 0.15, and test set has 0.15 of the data (Loukas, 2021). Based off the exploratory data analysis, the time step is set to 30 to capture the monthly patterns. The training set inputs and targets are created using the training set and converting it into a dataset matrix. A list of 30 values representing the 30 steps is added as the input for the corresponding predicted value, which is the 31st time step and added to the array representing the predicted values. This process is repeated to create the inputs and outputs of the training, validation, and testing set. The inputs are reshaped to include the number of features which is one feature, closing price (Brownlee, 2020). The data is then ready to be used in the Experiment A's model.

Experiment B: Input – Closing Price, Time Step – 15

Experiment B uses a smaller time step of 15 days rather than the 30 that was used in Experiment A. This experiment is used to see if the patterns in the lag of 15 days produces better

performance results than using lag of 30 days. Similar to Experiment A's preprocessing steps, the only difference is creating a training, validation, and test set with 15 days of closing prices as the input. The output is the 16th time step. The data is then used to train and evaluate the models.

Experiment C: Input – Closing Price and New Probable Cases, Time Step – 30

Like the prior experiments, Experiments C has the same goal of predicting the closing price for the next day. However, this experiment will use the inputs, closing price and new probable COVID-19 cases. This multivariate experiment undergoes similar preprocessing steps as Experiment A's but with two input variables rather than one. Along with the closing prices, the experiments also use COVID-19 data downloaded from Centers for Disease Control and Prevention.

The raw data is condensed by grouping the same dates together and finding the sum of each date and its corresponding variable. It is then left merged on date with the stock price dataframe with dates and closing prices. Because there were no existing COVID-19 cases until two years ago, the merged dataframe shows that there are missing dates and missing values for the COVID-19 data for 729 days of data. The missing values are replaced with zero to represent zero cases.

The merged dataframe is explored using a confusion matrix to analyze the correlation, a measure of strength in relationship, between the closing prices and new cases and new probable cases. The probable new cases have a larger correlation value with the closing prices than new cases. Despite it having the larger correlation value with closing price, probable new cases only have a correlation value of 0.61 with closing price. The closing price and probable new cases for the five years of historical data are visualized in line graphs.

Similar to the univariate experiments, this experiment scales the data to between 0 and 1 and splits it into a training, validation, and test set. Because the Experiment A models performed better than Experiment B, the time step of 30 was taken for the two inputs in this experiment. The input is converted into a dataset matrix with 2 values for each time step for 30 timesteps. The output of the 31st time step is added to the output set (Brownlee, 2020). This process is used on the training, validation, and test set. The preprocessed data is then used in the models.

Experiment D: Input – Closing Price and New Probable Cases, Time Step – 30, Dates filtered to existing COVID-19 cases

This experiment uses the same closing prices and probable new cases as Experiment C as well as preprocessing steps. The only difference is that it right merges the closing price and probable new cases datasets instead of a left merge, resulting in a dataframe whose dates begin when the COVID-19 cases begin to accumulate. The dataframe shows that there are some missing values and dates from the stock market dataframe. The missing values are due to the stock market being closed on weekends and holidays, while probable new cases are reported every day. The rows with missing values are filtered out. The correlation values are observed between the closing stock prices and probable new cases. The correlation value between closing stock prices and probable new cases decreased from Experiment C's correlation value. The correlation value between the two is 0.52, representing a weak positive correlation. The two inputs are visualized as line graphs starting from January 2020, the date when COVID-19 cases start accumulating.

After scaling the inputs, the dataset is split into training, validation, and test sets. Because there are less days available to train, tune, and evaluate the model on, the dataset is split with 0.58 for training, 0.22 for validation, and 0.2 for test sets, resulting in 307, 116, 107 days,

respectively. The input and output of each set is converted using the same process as Experiment C with a lag of 30 days.

Modeling

An long short term memory or LSTM model will be used to build the forecaster. This model architecture captures temporal correlations from the sequences and memorizes these past patterns to be used as an input and context for the current step. It takes in the input of these past temporal correlations as the hidden state of memory and the current set of sequences to predict the closing price of the next day. Along with the predicted closing price of the next day, the model also outputs the hidden state of memory for the current cell state. The model takes into consideration long-term and short-term memory with the use of input, output, and forget gates. These gates allow the model to remember the relevant and forget the irrelevant patterns (Srinivasan, 2022).

The model is created using a Sequential class. The models will all use LSTM layers where each cell in a LSTM layer capture a temporal correlation. A bidirectional wrapper on the LSTM is used in one experiment, which runs two LSTM layers with one reading the input forwards and the other reading the input backwards. This model is used to observe if there are any temporal correlations from viewing the sequences backwards that would improve the performance of the model. A dropout layer is also used in all the models as a regularization technique where a random 0.5 of the nodes is dropped for each iteration. The output layer uses the activation function ReLu and one node for the predicted closing value. The models use a mean squared error for its loss function, Adam for its optimizer, and mean absolute error for its performance metric. A fixed batch size of 32 and number of epochs of 50 is used for all

experiments. The early stopping callback is used to prevent overfitting. It will stop model training when the validation MAE score does not improve for three epochs. For each experiment, the performance is analyzed by visualizing the training and validation loss and MAE score. The predicted prices for the test set will be compared with the actual prices using a line graph (Loukas, 2021). The performance and hyperparameters for each model are recorded in a table in Figure 1.

Results

Experiment A: Input – Closing Price, Time step – 30

Experiment A.1

This model architecture uses two LSTM layers with 20 cells each, a dropout layer of 0.5, and a dense layer of 1 node with the activation function ReLu. The test MAE score is 0.03, which is good. The performance metric plots in Figure 4 show that the training and validation loss and MAE scores are similar to each other, indicating there is no overfitting. The use of early stopping helped to prevent the training and validation graph from going in opposite directions. Both loss and MAE plots show that the graphs are headed in the correct direction of zero. Figure 5 shows the closing prices of actual and predicted stock prices for the test set. The prediction values lag behind the actual values. It does not seem to do a very good job with predicting when the dips and spikes occur. However, the model is able follow upward and downward trends, so it does a good job predicting smoother changes in prices.

Experiment A.2

This experiment has increased model complexity with two sets of LSTM and dropout layers. Similar to Experiment A.1, the LSTM will have 20 nodes and the dropout will be 0.5. The performance metric plots show that the training and validation MAE values are similar, so there

is no overfitting. The MAE and loss values are also close to 0, which is good. The test MAE score is the same as the preceding experiment's despite having ten times the parameters. The comparison of actual and predicted stock price of the test set in Figure 6 looks shows a lag in the predicted prices. Comparing the preceding experiment's and this experiment's line graphs in Figure 7, this experiment did a better job predicting closer to the actual prices. Experiment A.1's predicted graph generally underestimated the values. This experiment's graph shows that it does not do as good of a job in predicting dips as that of the preceding experiment.

Experiment A.3

This experiment uses the same architecture as Experiment A.2, but 40 nodes, instead of 20, are used in the LSTM layers. The test MAE score is the same as the other experiment's. The performance metric plots show that there is no overfitting since the training and validation values are similar. From the comparison stock price visual with the predicted values of the other experiment's and the actual values in Figure 8, this model does not seem to lag even more than Experiment A.2's when it comes to dips. This model is more accurate when it comes to predicting the actual values during upward trends but does a bad job predicting downward trends. The other models do a better job predicting downward trends, especially Experiment A.1.

Experiment A.4

This model uses the same architecture as Experiment 2 with two LSTM layers of 20 nodes and 2 dropout layers. The first LSTM, however, has a bidirectional wrapper to extract patterns on the sequences forwards and backwards. The test MAE is larger than the other experiments at 0.08. The performance metric plots show that the training and validation graphs are similar to each other. There is no large difference between their values. Figure 9 shows that the predictions were very far from the actual values, especially compared to the other experiment

predicted values. This model's predictions lags behind the actual values as well as underestimates them. It does not do as well as the other models, so the patterns extracted from the backwards sequences are not helpful. Of all the Experiment A models, Experiment A.1 had the best performance because the visualization shows that it lagged the least and also was able to follow downward and upwards trends well. It also had the least among of parameters to train but had the same and better test MAE score as the other models.

Experiment B: Input – Closing Price, Time step – 15

Experiment B.1

This experiment uses the same model architecture as Experiment A.1 except for the input shape of (15,1) since it uses a time step of 15 instead of 30. The test MAE score is greater than its Experiment A.1 counterpart, so this model does not do as well. The performance plot shows that there is no overfitting since the values for validation and training are close. The visualization of the predicted and actual closing prices in Figure 10 show that the predicted prices lag and underestimate the values. The predicted values are generally less than the actual values. Compared to Experiment A.1, this experiment lags more during dips and spikes, so it does not predict those as well. Experiment A.1's predicted values are also more similar to the actual values than this model's predicted values.

Experiment B.2

This model has the same architecture as Experiment A.2's with a different input shape. The test MAE improve compared to Experiment B.1. It has the same score as Experiment A.1, A.2, and A.3 at 0.03. The performance plots show no overfitting since the training and validation graphs are similar. As seen by this model predicted the test set prices much better than the preceding model's predicted prices. It can be seen in Figure 11 that this model still lags in

predicting dips and spikes but follows the downward and upward trends well. The predicted values are also more similar to the actual values than the preceding experiment's predicted values. Compared to Experiment A.2, this model seems to predict the actual values less well since the it underestimates the values more.

Experiment B.3

This experiment uses increased model complexity compared to Experiment B.2 by using the same model architecture as Experiment A.3. The test MAE is good and similar to the preceding experiment's. The performance plots show that there is no overfitting. The stock price prediction visual, Figure 12, shows that there is a lag for drops in price, but the lag for increases in price is not bad. It also seems to estimate the values well. In Figure 13, this model does better lag-wise than its counterpart, Experiment B.3. It also predicts the values more accurately. Figure 14 shows that this model is more accurate in its predictions than that of Experiment B.2. This model does a better job predicting the spikes than Experiment A.2, as seen in Figure 15. In Figure 16, Experiment A.1's model does a better job predicting dips than this model, but this model does a better job predicting the spikes.

Experiment C: Input – Closing Price and New Probable Cases, Time step – 30

Experiment C.1

This experiment uses the same model architecture as the Experiment A.1, the best model so far. The input shape is the only difference between that model and this model. This model has in input shape of (30,2) due to the two inputs. The test MAE is good and the same as Experiment A.1's. The performance plots show that the training and validation graphs are similar, so there is no overfitting. In Figure 17, the model's predicted test set prices do not lag as much as the other models did. It does not seem to predict spikes as well as it does drops. It does a good job

approximating the values. Figure 18 shows the predicted values in comparison to Experiment A.1's predicted values. From the visualization, Experiment A.1 predicts the values more accurately. Experiment A.1 predicts the spikes and drops better than this model.

Experiment C.2

This model has the same architecture as Experiment A.2. The test MAE is the 0.03 so it does a good job of predicting the values. The performance metric plots do not indicate overfitting. There is a slight difference between training and validation MAE, but the difference is only 0.04. In Figure 19, as compared to Experiment C.1, this model does a better job predicting the actual values but lags more during spikes and drops. Figure 20 shows the comparison of the predicted values with Experiment A.1's values. Experiment C.2 lags more in predicting the drops and spikes than Experiment A.1.

Experiment D: Input – Closing Price and New Probable Cases, Time step – 30, Dates filtered to existing COVID-19 cases

Experiment D.1

This experiment uses the same single-layer LSTM architecture as Experiment A.1 but with an input shape of (30,2) for the two inputs. The test MAE is 0.14, the largest MAE score of the experiments. The performance metric plots show that there is no overfitting since the training and validation loss and MAE values are similar. The visual in Figure 21 shows that this model did a bad job predicting the prices of the test set. It underestimated the values as well as did not predict most of the drops and spikes. It also lags in prediction of those drops and spikes. Overall, this model did very badly.

Conclusion

The LSTM models were able to successfully capture temporal correlations that provided context for the predicted closing price for the next day. From these experiments, the best model was Experiment A.1, the single layer LSTM with an input of closing price and time step of 30. This model, along with other models, had the best test MAE score oof 0.03. Unlike the other models with the same test MAE score, Experiment A.1 also had the least number of parameters that needed to be trained.

Experiment A and B differed by the time step used to create the datasets. From Experiment B, it can be observed that the model was able to extract more relevant temporal correlations from a time step of 30 than a time step of 15. Experiment A.1 and Experiment B.1 had the same model architecture except for the input shape due to the difference in time lag. Experiment A.1 had the better test MAE score than Experiment B.1. With the use of a multi-layer LSTM for Experiment B.2, the model was able to achieve the same test MAE score as Experiment A.1, but with ten times as many parameters. Therefore, the simpler Experiment A.1 was the better model and a lag of 30 had more relevant temporal correlations.

Experiment C and D aimed to include COVID-19 data as an additional variable for the model to factor in when predicting the stock market closing prices for the next day. During exploratory data analysis, it was evident that there was a weak positive relationship between the COVID-19 new probable cases and closing prices. In Experiment C.1, the model did achieve the same test MAE score as Experiment A.1's model. Looking at the visual in Figure 18, the Experiment C.1 model was able to more accurately predict the value of the stock each day compared to Experiment A.1 but lagged in its ability to predict the dips as accurately as Experiment A.1. The COVID-19 data used in Experiment C did seem to help with the

predictions especially with the spikes in prices. However, Experiment A.1 seemed to best predict the dips and spikes, even with the increased complexity model of Experiment C.2.

Experiment D used data starting when the COVID-19 cases began in order to see if the model would consider the weights for the hidden states differently than that of using training data without any COVID-19 cases for the first two years. Experiment D.1 did much worse than its counterpart Experiment C.1. It had a larger test MAE score and did not seem to predict spikes in the prices and also lagged in the predicted drops of the data. Overall, the addition of the COVID-19 probable new cases data did not result in a big difference in performance for Experiment C or D. The recommended S&P 500 closing stock price forecasting model is Experiment A.1's single-layer LSTM model.

References

Brownlee, J. (2020, August 27). *How to develop LSTM models for time series forecasting*. Machine Learning Mastery. Retrieved 2022, from https://machinelearningmastery.com/how-to-develop-lstm-models-for-time-series-forecasting/

CDC Case Task Force. (2022). *United States covid-19 cases and deaths by State over time*. Centers for Disease Control and Prevention. Retrieved 2022, from https://data.cdc.gov/Case-Surveillance/United-States-COVID-19-Cases-and-Deaths-by-State-o/9mfq-cb36

Chollet François. (2021). *Deep learning with python*. *O'Reilly*. Manning Publications. Retrieved 2022.

Gurav, U., & Kotrappa, S. (2020). Impact of COVID-19 on Stock Market Performance Using Efficient and Predictive LBL-LSTM Based Mathematical Model. *International Journal on Emerging Technologies*, *11*(4), 108–115. Retrieved 2022, from https://ssrn-com.turing.library.northwestern.edu/abstract=3670565

Kenton, W. (2022, February). *The S&P 500 Index: Standard & Poor's 500 Index*. Investopedia. Retrieved 2022, from https://www.investopedia.com/terms/s/sp500.asp

Nabipour, M., Nayyeri, P., Jabani, H., Mosavi, A., Salwana, E., & S., S. (2020). Deep Learning for Stock Market Prediction. *Entropy*, *22*(8), 840. https://doi.org/10.3390/e22080840

Srinivasan, S. (2022). *MSDS 458 AI/Deep Learning: Sync Session #4* [Zoom Cloud

Recordings].

Appendix

Figure 1. Model performance.

Experiment	Description	Training	Validation	Testing	Process Time (seconds)	Epochs	Total Parameters	Hyperparameters
Experiment A: Input – Closing Price, Time Step – 30								
A.1	Single Layer LSTM	0.02	0.04	0.03	17.00	21	1,781	LSTM - 20 Dropout - 0.5 Dense - 1
A.2	Multi-Layer LSTM	0.04	0.04	0.03	15.40	7	11,191	LSTM - 20 Dropout - 0.5 LSTM - 20 Dropout - 0.5 Dense - 1
A.3	Multi-Layer LSTM (increased model complexity)	0.04	0.03	0.03	16.10	6	19,721	LSTM - 40 Dropout - 0.5 LSTM - 40 Dropout - 0.5 Dense - 1
A.4	Multi-Layer Bidirectional LSTM	0.04	0.06	0.08	24.50	10	8,421	Bidirectional LSTM - 20 Dropout - 0.5 LSTM - 20 Dropout - 0.5 Dense - 1
Experiment B: Input – Closing Price, Time Step – 15								
B.1	Single Layer LSTM	0.04	0.05	0.07	7.05	9	1,781	LSTM - 20 Dropout - 0.5 Dense - 1
B.2	Multi-Layer LSTM	0.04	0.05	0.03	14.50	10	11,191	LSTM - 20 Dropout - 0.5 LSTM - 20 Dropout - 0.5 Dense - 1
B.3	Multi-Layer LSTM (increased model complexity)	0.03	0.02	0.03	17.50	12	19,721	LSTM - 40 Dropout - 0.5 LSTM - 40 Dropout - 0.5 Dense - 1
Experiment C: Input – Closing Price and New Probable Cases, Time Step – 30								
C.1	Single Layer LSTM	0.04	0.03	0.03	9.41	9	1,860	LSTM - 20 Dropout - 0.5 Dense - 1
C.2	Multi-Layer LSTM	0.04	0.08	0.03	16.50	8	5,141	LSTM - 20 Dropout - 0.5 LSTM - 20 Dropout - 0.5 Dense - 1
Experiment D: Input – Closing Price and New Probable Cases, Time Step – 30, Dates filtered to existing COVID-19 cases								
D.1	Single Layer LSTM	0.12	0.11	0.14	4.69	5	1,861	LSTM - 20 Dropout - 0.5 Dense - 1

Figure 2. Closing prices from 02/28/2017 to 02/28/2022.

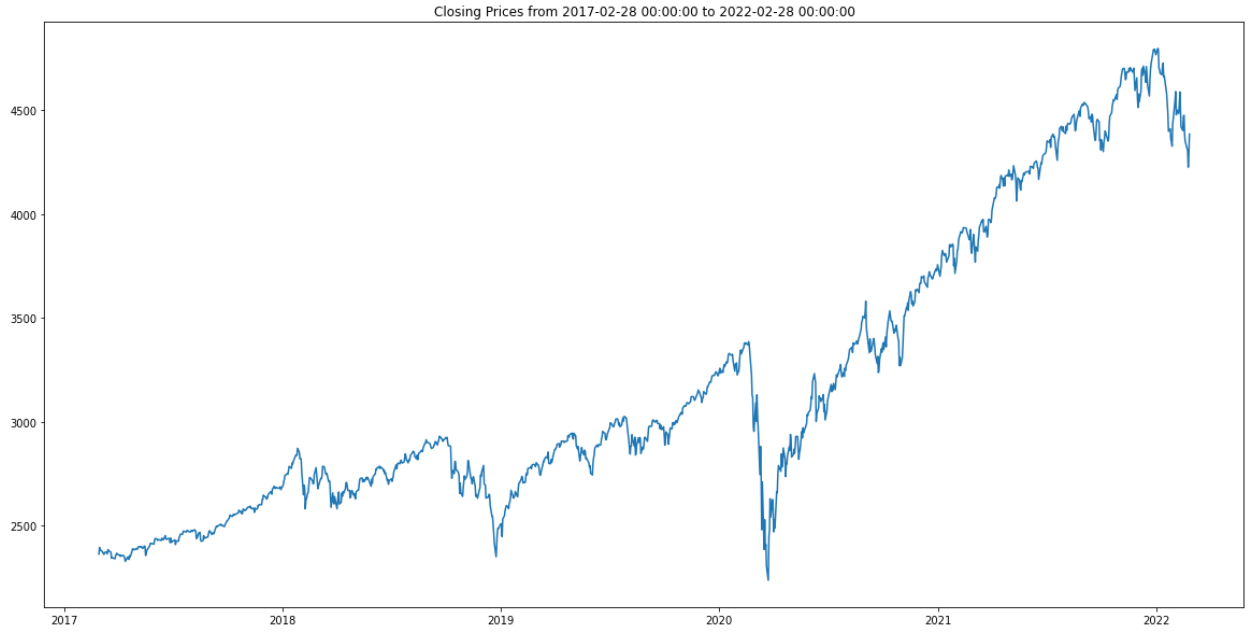

Figure 3. Closing prices from 01/01/2021 to 01/01/2022.

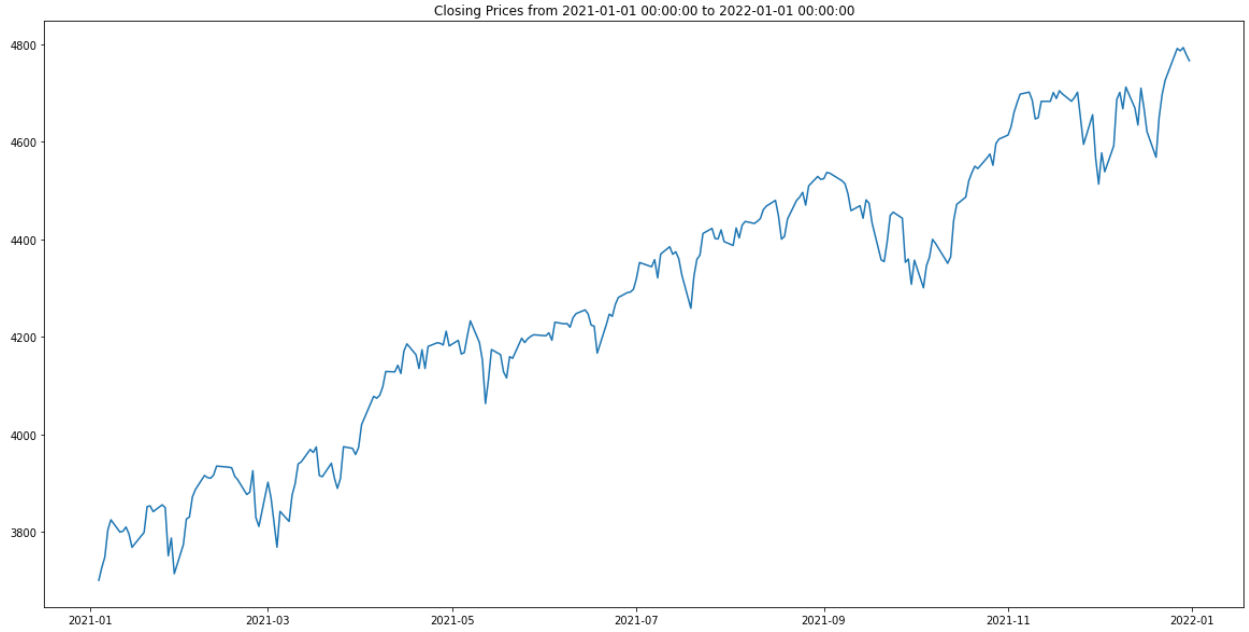

Figure 4. Experiment A.1 performance plots.

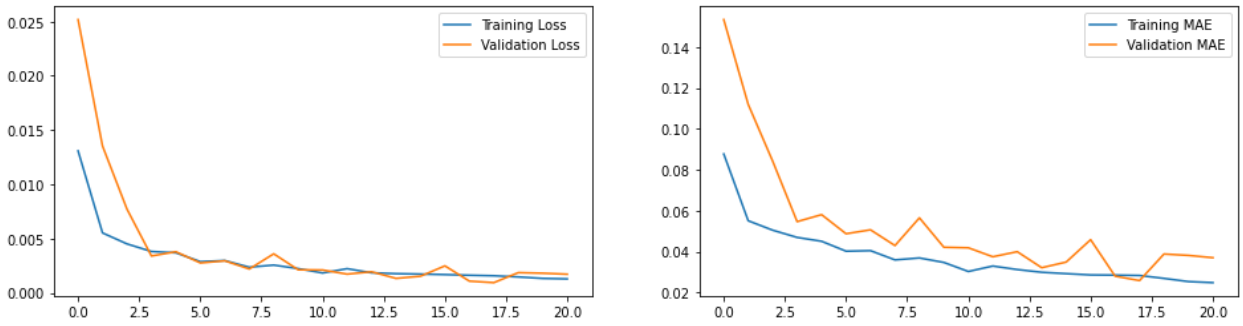

Figure 5. Compare Experiment A.1 predicted and actual test set values.

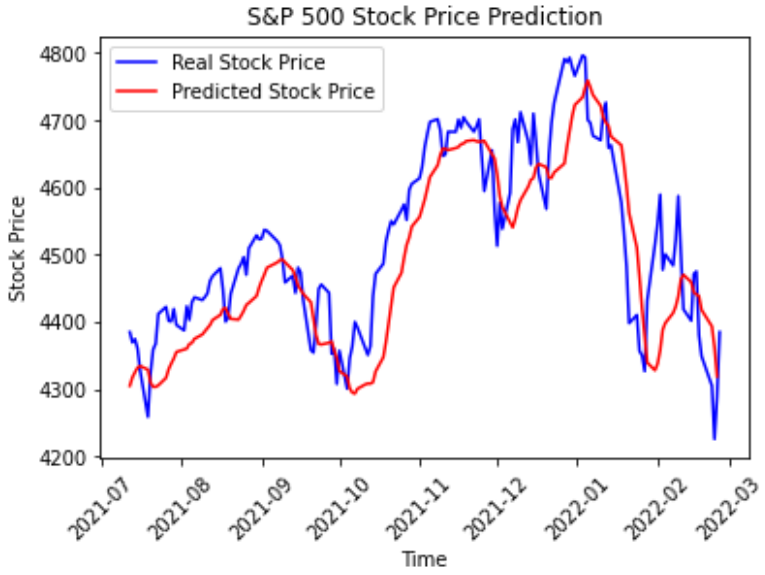

Figure 6. Compare Experiment A.2 predicted and actual test set values.

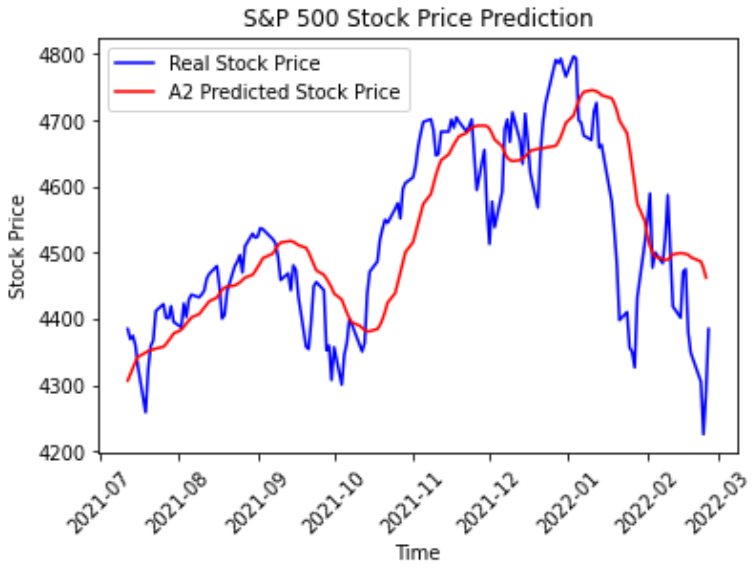

Figure 7. Compare Experiment A.1 and A.2 predicted test values and actual values.

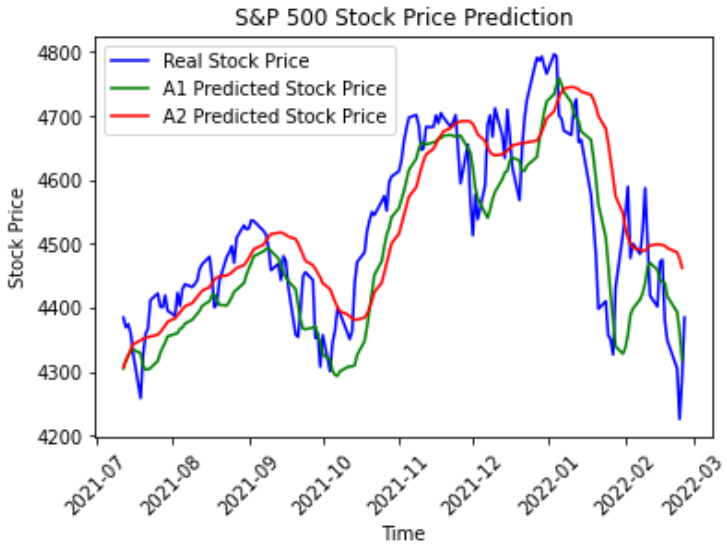

Figure 8. Compare Experiment A.1, A.2, and A.3 predicted test values and actual values.

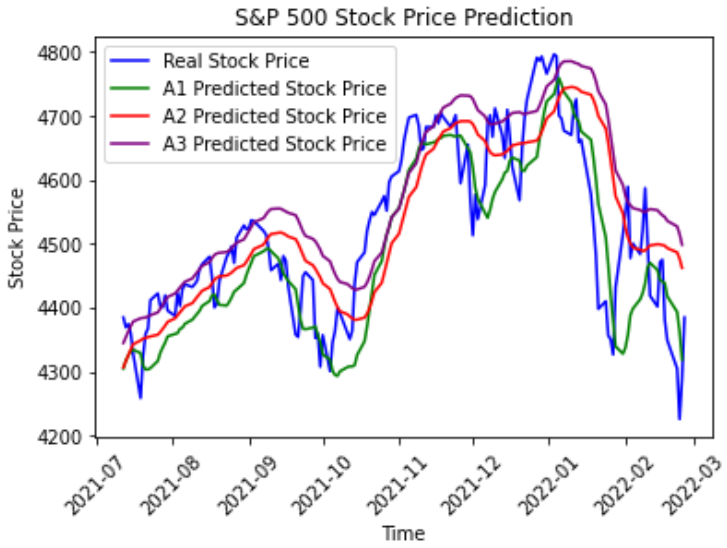

Figure 9. Compare Experiment A.1, A.2, and A.4 predicted test values and actual values.

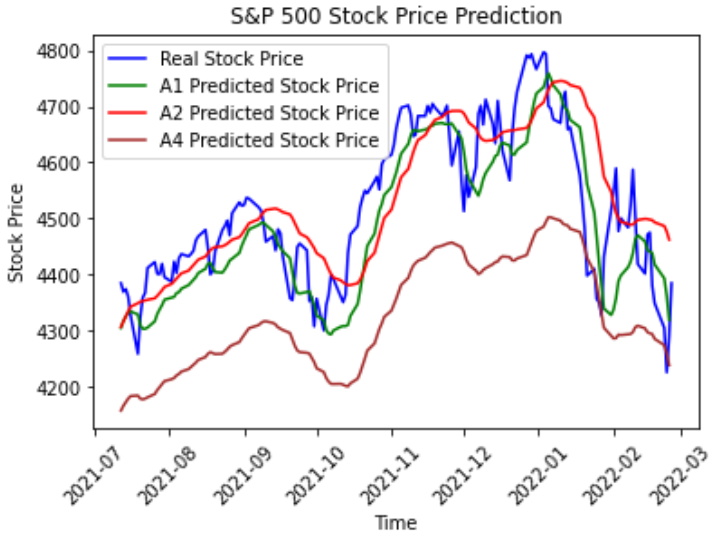

Figure 10. Compare Experiment A.2, B.1, and B.2 predicted test values and actual values.

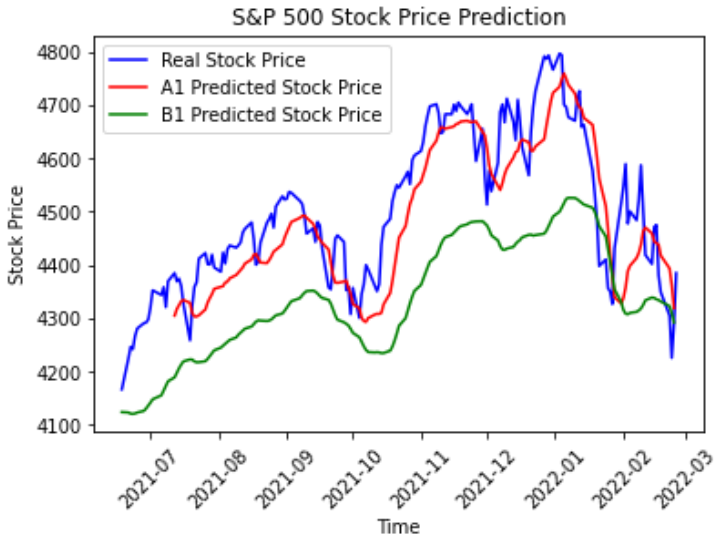

Figure 11. Compare Experiment A.2, B.1, B.2 predicted test values and actual values.

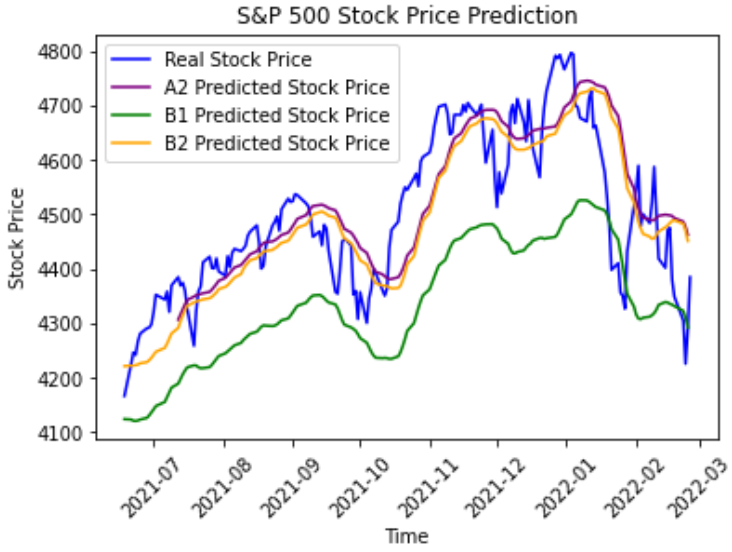

Figure 12. Compare Experiment B.3 predicted test values and actual values.

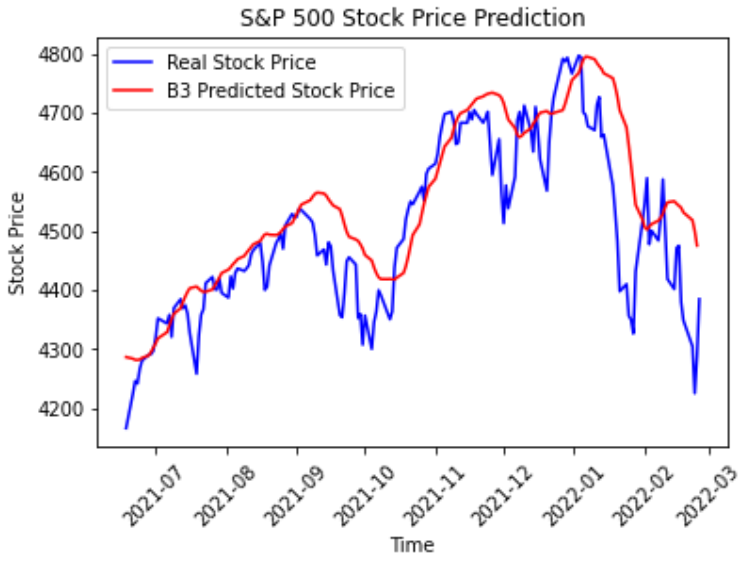

Figure 13. Compare Experiment A.3, B.3 predicted test values and actual values.

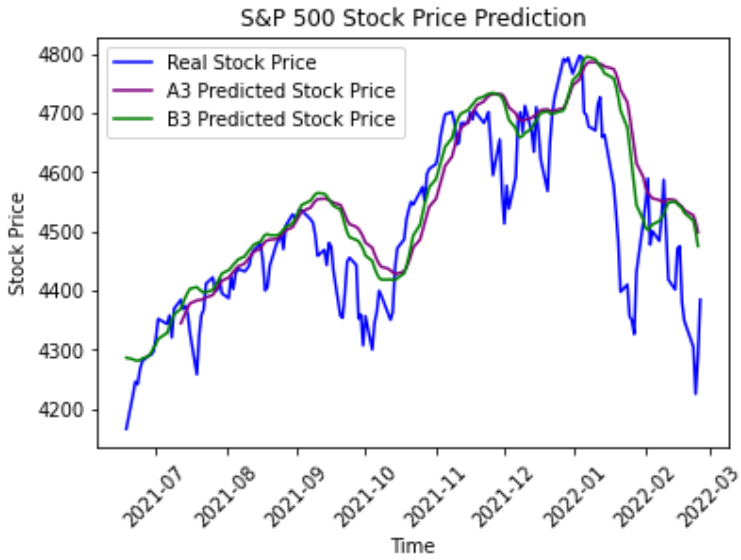

Figure 14. Compare Experiment B.2, B.3 predicted test values and actual values.

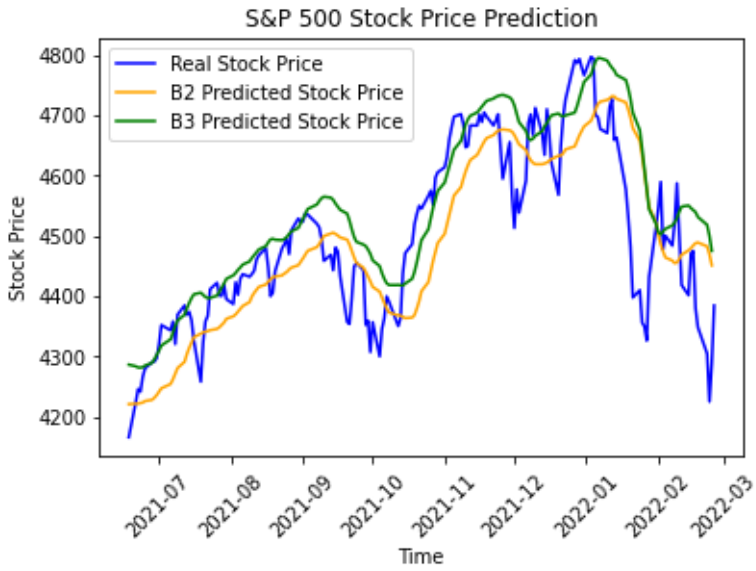

Figure 15. Compare Experiment A.2, B.3 predicted test values and actual values.

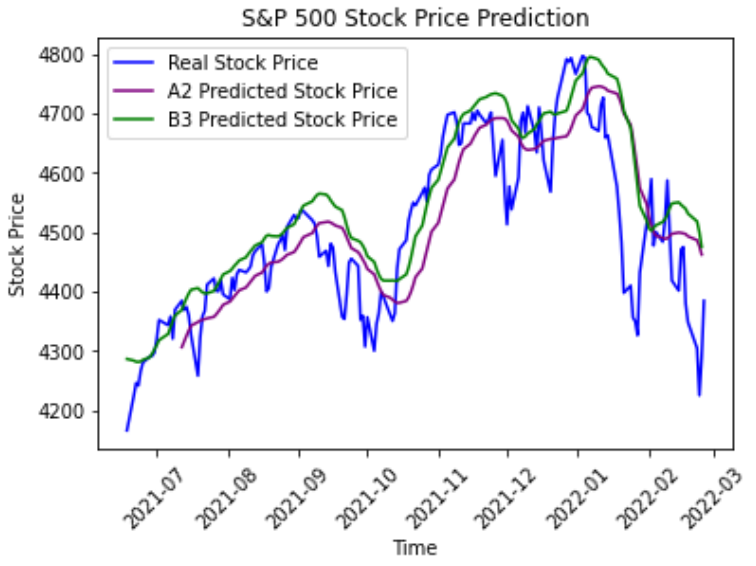

Figure 16. Compare Experiment A.1, B.3 predicted test values and actual values.

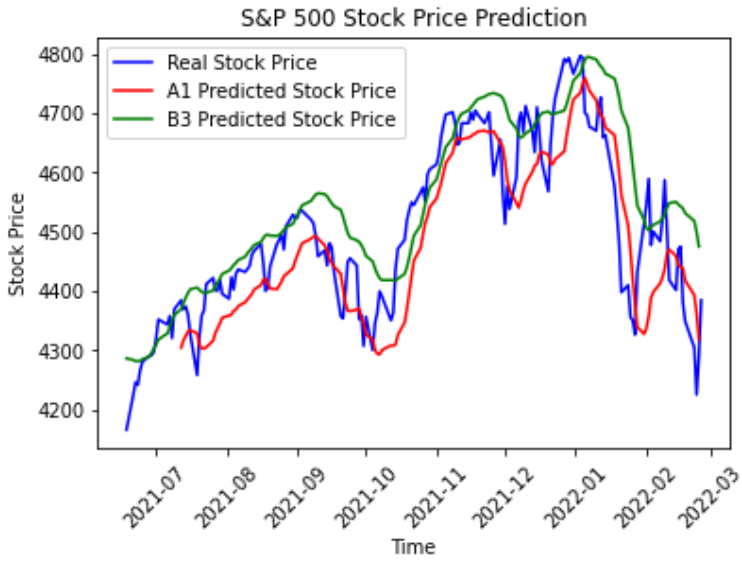

Figure 17. Compare Experiment C.1 predicted test values and actual values.

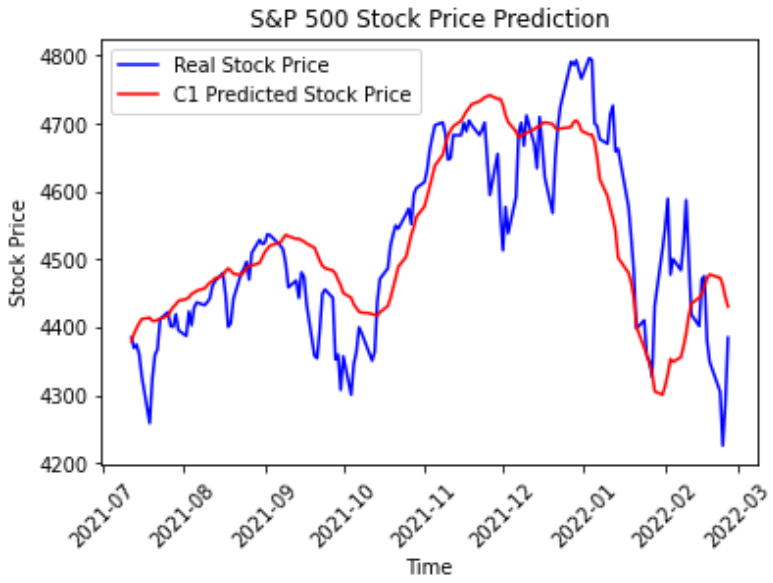

Figure 18. Compare Experiment C.1 and A.1 predicted test values and actual values.

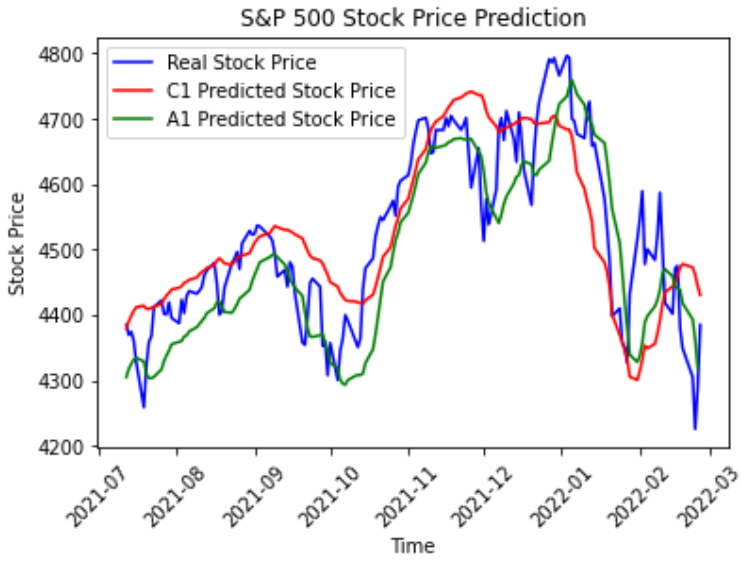

Figure 19. Compare Experiment C.1 and C.2 predicted test values and actual values.

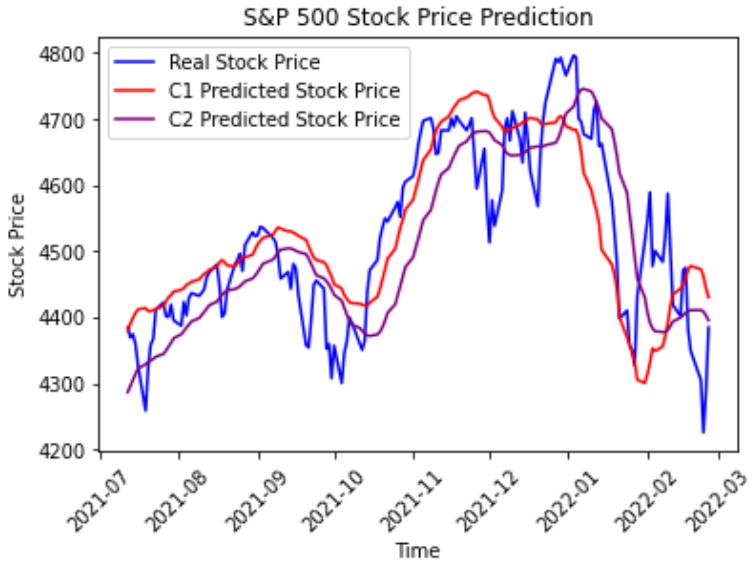

Figure 20. Compare Experiment C.2 and A.1 predicted test values and actual values.

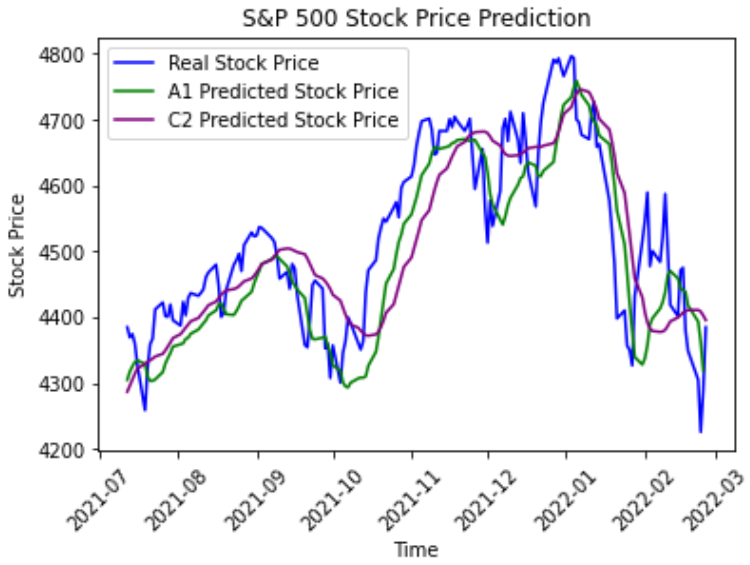

Figure 21. Compare Experiment D.1 predicted test values and actual values.