Stock Market Movement Prediction

XCS229ii - Final Report

Authors: Han Yu

Last Updated: May 22, 2021

Abstract

In this project, I have developed several LSTM-based machine learning models (via three different architectures) to predict the next day's stock price of **Alphabet Inc Class C** (NASDAQ: **GOOG**) via the information derived from the previous X days. In addition to evaluating the model performance by looking at standard metrics such as Root Mean Square Error (RMSE), a simple trading strategy has also been implemented to simulate the trading events based on the predicted stock prices. The final return of investment calculated at the end of simulation is also used as a metric to evaluate and compare the model performances when deployed in a "real" algorithmic trading system.

Introduction

Accurately predicting the stock markets movement is a complex task as there are hundreds of events and pre-conditions for a particular stock to move in a particular direction. We want to understand whether we could build a ML model to predict the future stock price movement, how accurately it could be, and whether the insights derived from the ML model can be further applied to derive a reasonable algorithmic trading strategy.

In [1], the author provided insightful explanations on why long short-term memory (LSTM) neural networks can be used to learn the patterns from time series data for prediction. The author also provided very informative explanations of the basic building blocks in LSTM, which are designed to deal with the vanishing gradient issues encountered in the standard recurrent neural networks.

In [2], the authors evaluated the tree-based models and neural network models to predict four stock market share groups. Tree-based models include decision tree, bagging, random forest, boosting, gradient boosting, and XGBoost; neural network models include standard neural network, recurrent neural network, and LSTM neural network. Based on the design of experiments, LSTM is shown to be the best model for prediction among all stock market groups based on the prediction error measures results.

In [3], a generator-discriminator architecture has been developed to predict stock price movements, where a Generative Adversarial Network (GAN) with LSTM is used as generator, and a Convolutional Neural Network is used as a discriminator. The author claims that although there aren't many applications of GANs being used for predicting time-series data, such architecture should work since we want to 'generate' data for the future that will have similar distribution as the the historical trading data, and generator and discriminator is sort of playing a minmax game until the generator is so good at producing indistinguishable data to "fool" is the discriminator. LSTM is used as the generator since LSTM can keep track of all previous data points and capture patterns developing through time, while avoiding the vanishing gradient issue of the stand recurrent neural network. CNN is used as the discriminator because CNN is powerful at extracting features from features and it also works well on spatial data (e.g., data points that are closer to each other are more related to each other, than data points spread across, this also holds for some time series data such as daily stock prices).

In [4], the author discussed how to design and backtest a financial trading strategy using open source Python tools. Although the focus in [4] is not machine learning (e.g., only a very simple decision tree model was developed to predict buy/sell signals), the process discussed in [4] is actually very interesting in practice — if we can use machine learning to develop models and generate some trading strategy, how to assesses the viability of such trading strategy by discovering how it would play out using historical data, before we have some confidence to employ this trading strategy going forward.

In [5], the authors introduced a deep reinforcement learning (DRL) library FinRL that facilitates beginners to expose themselves to quantitative finance and to develop their own stock trading strategies. The DRL models which have been implemented in FinRL are also explained in [6].

Inspired by those prior arts, this project examines the stock price prediction performance of three popular architectures in sequence modeling, which includes Long Short-Term Memory (LSTM), Bidirectional LSTM (Bi-LSTM), and Encoder-Decoder. Experiments have been designed to evaluate the prediction performance among those three architectures when short sequence(7 days' historical data), medium sequence (14 days' historical data), and long sequence(28 days' historical data) are provided as inputs to train and predict the next day's stock price; a simple trading strategy has also been implemented to evaluate the return of investment using the models trained via those experiments.

The rest of this report is divided into 3 sections. Overall methodology and experimental protocols are entailed in the Methods section; experiment results and analysis are provided in the Results section; conclusion and potential future work is summarized in the Discussions section.

Methods

Hypothesise

1) Stock markets are not 100% random and totally unpredictable; price movement on a given day is an unknown function of some leading indicators and trailing indicators from the past n days.
2) Sequence models (e.g., RNN, LSTM) can be applied for time series forecasting, and therefore, we can potentially build sequence models to predict stock prices.

Data Source

We will predict the price movements of **Alphabet Inc Class C** (NASDAQ: **GOOG**). For this purpose, we will use the daily stock price data of GOOG from Jan 1st, 2010 to April,30, 2021, derive features from the raw dataset, and build a ML model to predict the Adjusted Close Price for the next day.

Exploratory Data Analysis

Exploratory data analysis is briefly summarized in Figure 1, Figure 2, and Figure 3. The original raw data source only contains 7 columns, including the “Date of Trading”, the “Highest” and the “Lowest” , the “Open” and “Close” stock price for the trading day, the the total number of shares that are actually traded (bought and sold) during the trading day, and the final close stock's value after accounting for any corporate actions of the trading day (Adj Close).

	Date	High	Low	Open	Close	Volume	Adj Close
0	2010-01-04	313.579620	310.954468	312.304413	312.204773	3927065.0	312.204773
1	2010-01-05	312.747742	309.609497	312.418976	310.829926	6031925.0	310.829926
2	2010-01-06	311.761444	302.047852	311.761444	302.994293	7987226.0	302.994293
3	2010-01-07	303.861053	295.218445	303.562164	295.940735	12876685.0	295.940735
4	2010-01-08	300.498657	293.455048	294.894653	299.885956	9484016.0	299.885956

Figure 1. Information Contained in the Raw Data

	High	Low	Open	Close	Volume	Adj Close
count	2851.000000	2851.000000	2851.000000	2851.000000	2.851000e+03	2851.000000
mean	765.924279	751.685989	758.711410	759.039803	3.125121e+06	759.039803
std	456.518406	446.725011	451.061251	451.830707	2.625549e+06	451.830707
min	220.314209	216.005356	218.336624	217.220810	7.922000e+03	217.220810
25%	351.490036	344.095245	348.294510	347.679306	1.401700e+06	347.679306
50%	638.700012	625.500000	634.330017	632.590027	2.101849e+06	632.590027
75%	1094.167542	1069.784973	1082.234985	1082.440002	4.199783e+06	1082.440002
max	2452.377930	2402.280029	2410.330078	2429.889893	2.976073e+07	2429.889893

Figure 2. Statistical Summary of the Raw Data

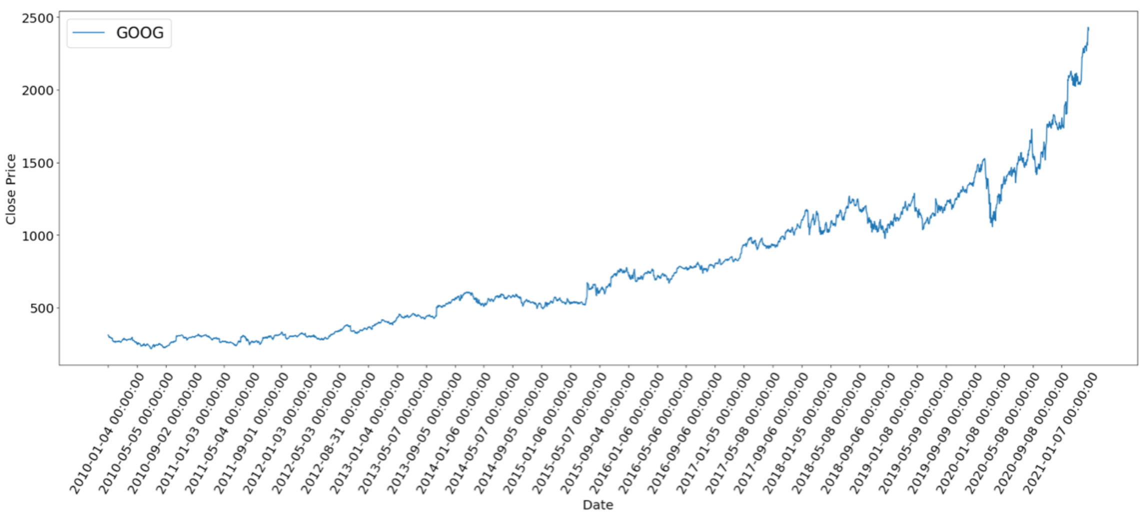


Figure 3. Time Series Plot of the Stock Closed Price

Feature Selection

Selected features to build and train the machine learning models are summarized in Table 1.

Feature	Definition	Rationale
ret_ema	Exponential Moving Average[1] of the daily return percentage which is calculated based on the "Adjusted Close Price" within a specified period X	Trading activities can be influenced based on the past return, which could be a leading indicator for future stock price movement.
vol_ema	Exponential Moving Average of the daily trading volumes	Historical momentum of stock trading volume could be a leading

[1] Note: Exponential Moving Average are used so that we can place greater weight on recent data points than ones that are further away from the current time period

		indicator for future stock price movement.
obv_ema	Exponential Moving Average of the On Balance Volume within a specified period X	Use the change in the volume pressure as an indicator of stock price movement. A positive volume pressure would result in higher prices as the demand surges, and likewise a negative volume pressure would eventually result in lower prices.
atr	The true range indicator is taken as the greatest of the following: current high less the current low; the absolute value of the current high less the previous close; and the absolute value of the current low less the previous close. The ATR is then a moving average, generally using 14 days, of the true ranges.	A stock experiencing a high level of volatility has a higher ATR, and a low volatility stock has a lower ATR. Volatility could be a leading indicator for future stock price movement.
willr	A momentum indicator that is the inverse of the Fast Stochastic Oscillator.	Readings from 0 to -20 are considered overbought. Readings from -80 to -100 are considered oversold.
bb_mb	A Bollinger band defines 2 lines that are set 2 deviations apart from the time series's simple moving average. It sets a constraint on the possibility of the next day's price as most of the price action happens within this area.	The closer the bands, the more volatile the price movement and hence the more likely a current trend may be ending or even reversing.

Table1. Summary of Selected Features

Correlation heatmap of the selected features is provided in Figure 4. Those features have been used to build and train the machine learning models.

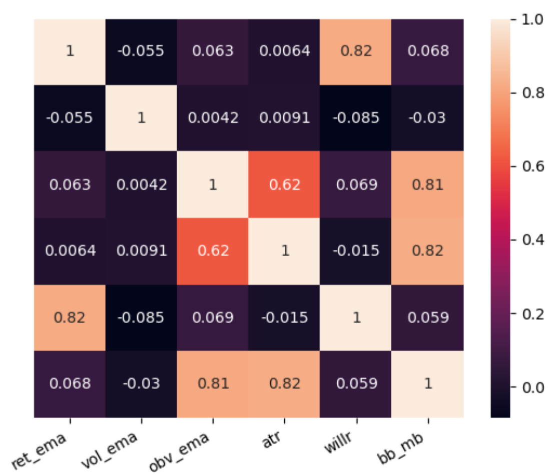


Figure 4. Correlation Heatmap of the Selected Features

Data Processing

Split of the Data: 80% for Train, 10% for Dev, and 10% for Test

Metrics

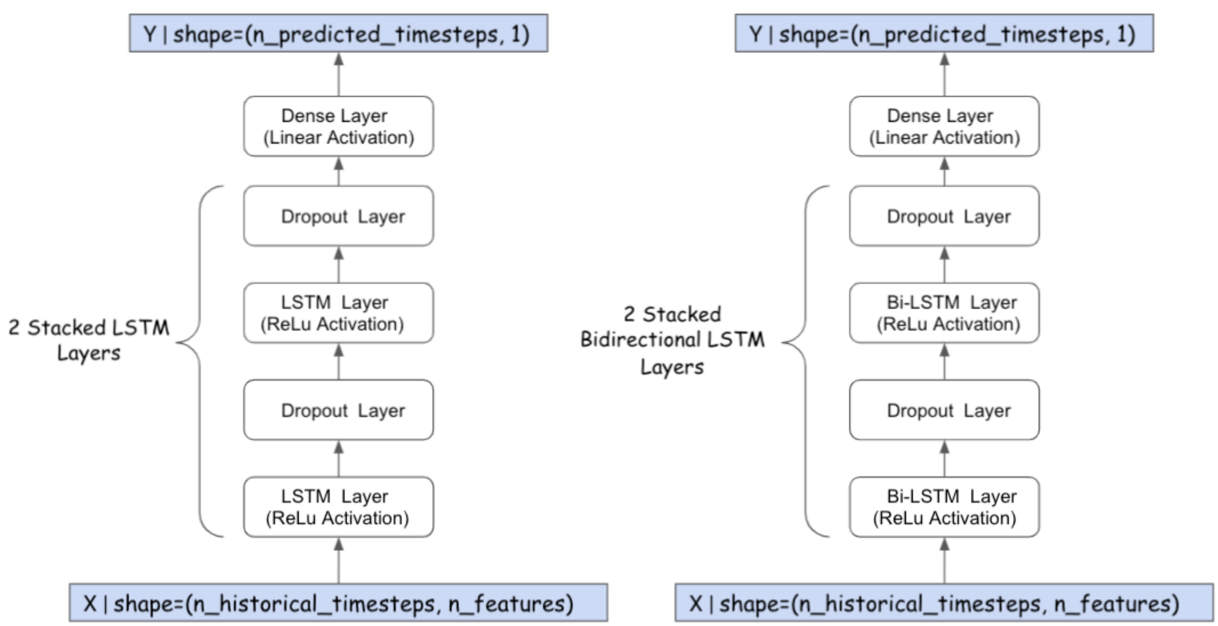
- Rooted Mean Square Error (RMSE) calculated using the predicted stock prices and the actual stock prices will be used as the one of the metrics to evaluate the model performance.
- A simple trading strategy is also implemented, and the return of investment via the trading strategy by using the stock price prediction from the machine learning models is also used as a metric to evaluate the model performance.

Model

LSTM offers a number of benefits when it comes to multi-step time series forecasting, for example:

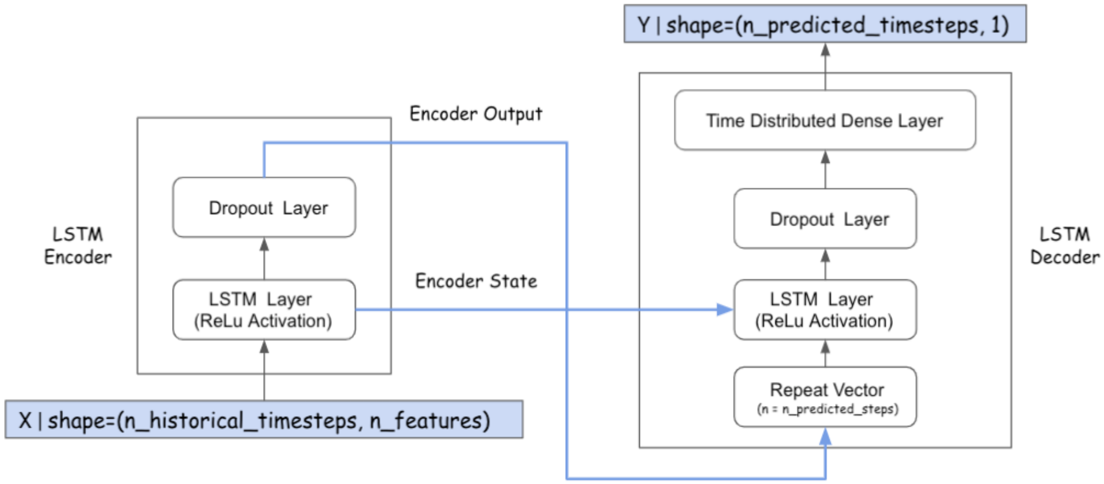
- LSTM is designed to take sequence data as input. It directly supports input sequences with multiple features at each time step.
- LSTM layer + Dense Layer is able to map input sequence data directly to an output vector, which can be used to represent forecasts of output at multiple time steps.
- LSTM based encoder + decoder architecture can also be used for forecasts of time series data.

Three models have been developed to predict stock price movement. The architectures of those models are illustrated Figure 5.



(a) 2 Stacked LSTM Layers

(b) 2 Stacked Bidirectional LSTM Layers



(c) LSTM Encoder + LSTM Decoder

Figure 5. Architecture of There LSTM-based Models

- Long Short-Term Memory (LSTM) is a type of recurrent neural network that addresses the vanishing gradient problem in vanilla RNNs through additional cells, input and output gates. Intuitively, vanishing gradients are solved through additional additive components, and forget gate activations, that allow the gradients to flow through the network without vanishing as quickly.

- Bidirectional LSTM (Bi-LSTM) is a sequence processing model that consists of two LSTMs: one taking the input in a forward direction, and the other in a backwards direction. Bi-LSTM effectively increases the amount of information available to the network, improving the context available to the algorithm (e.g. knowing what words immediately follow and precede a word in a sentence).

- Encoder-Decoder architecture was designed to tackle a specific task known as sequence-to-sequence learning. The encoder is a stack of several recurrent units (LSTM or GRU cells for better performance) where each accepts a single element of the input sequence, collects information for that element and propagates it forward. A summary of the input sequence is captured via the hidden vector, and the decoder, which is also a RNN, takes the hidden vector as an input and generates the output sequence.

A 2-layer stacked LSTM model has been developed as a baseline model, then a 2-layer stacked bidirectional LSTM model and a LSTM based Encoder-Decoder model have been further developed to compare with the baseline model.

Results

Experiments to Evaluate Model Performance for Stock Price Prediction

For each of the model architecture, we have built and trained model for the following experiments:

- Use features from past 7 Days/14 Days/ 28 Days as inputs and use next day's Adjusted Close Price as target to build and train a model; deploy the model to predict stock price using the test dataset; calculate the RMSE using the predicted stock price and the real stock price.

Experiments and analysis results based on the test data are summarized in Table 2 to compare Root Mean Square Error (RMSE) among the models (use GOOG stock price data from April 2020- April 2021as test dataset).

Model in the Experiment	**LSTM (RMSE)**	**Bi-LSTM (RMSE)**	**Encoder-Decoder (RMSE)**
Use Features from Past 7 Days to Predict next Day's Adjusted Close Price	60	58	64
Use Features from Past 14 Days to Predict next Day's Adjusted Close Price	138	111	59
Use Features from Past 28 Days to Predict next Day's Adjusted Close Price	233	110	106

Table 2. Summary of Experiment Results for Stock Price Prediction

All three models are able to learn the overall trends of the stock price using the historical data points with reasonable prediction performance via the test data. One observation from the current experiment is that as more historical data points are used to train a model to predict next day's price, Bi-Directional LSTM model and LSTM based Encoder-Decoder model seem to have better performance compared to the baseline LSTM model (in terms of RMSE).

Detailed model prediction results for each experiment are provided in Figure 6 - Figure 14.

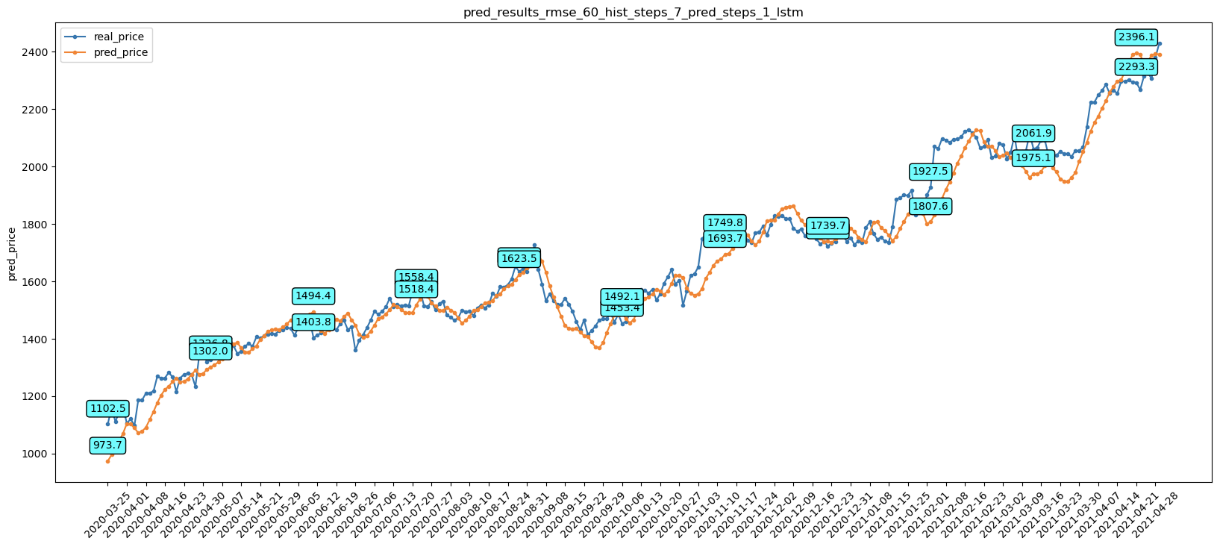


Figure 6. Use Features from Past 7 Days to Predict next 1 Day's Adjusted Close Price (LSTM, RMSE = 60)

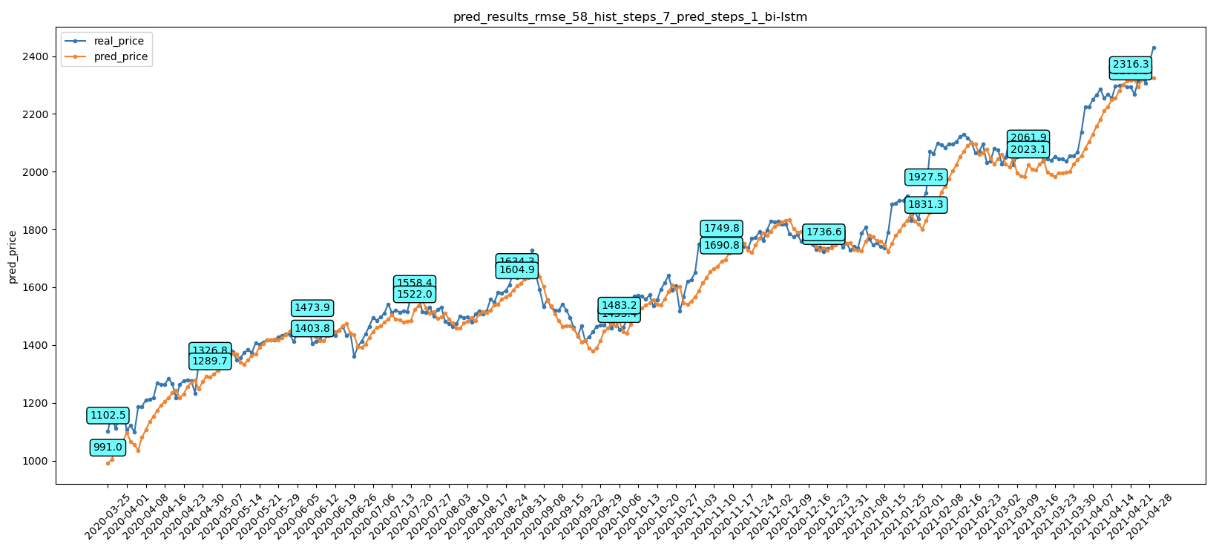


Figure 7. Use Features from Past 7 Days to Predict next 1 Day's Adjusted Close Price (Bi-LSTM, RMSE = 58)

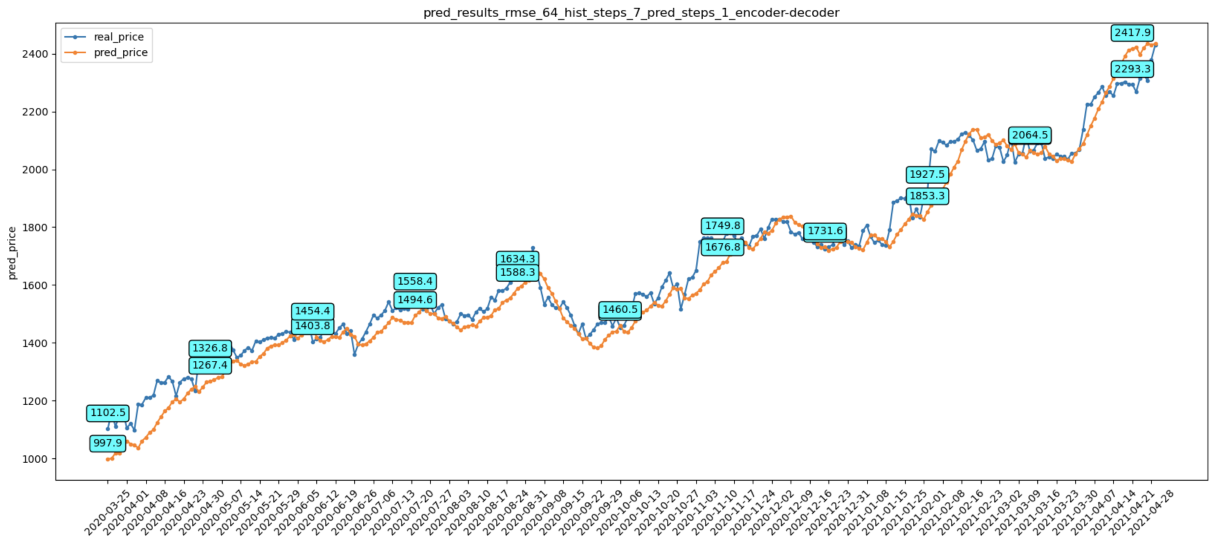


Figure 8. Use Features from Past 7 Days to Predict next 1 Day's Adjusted Close Price (Encoder-Decoder, RMSE = 64)

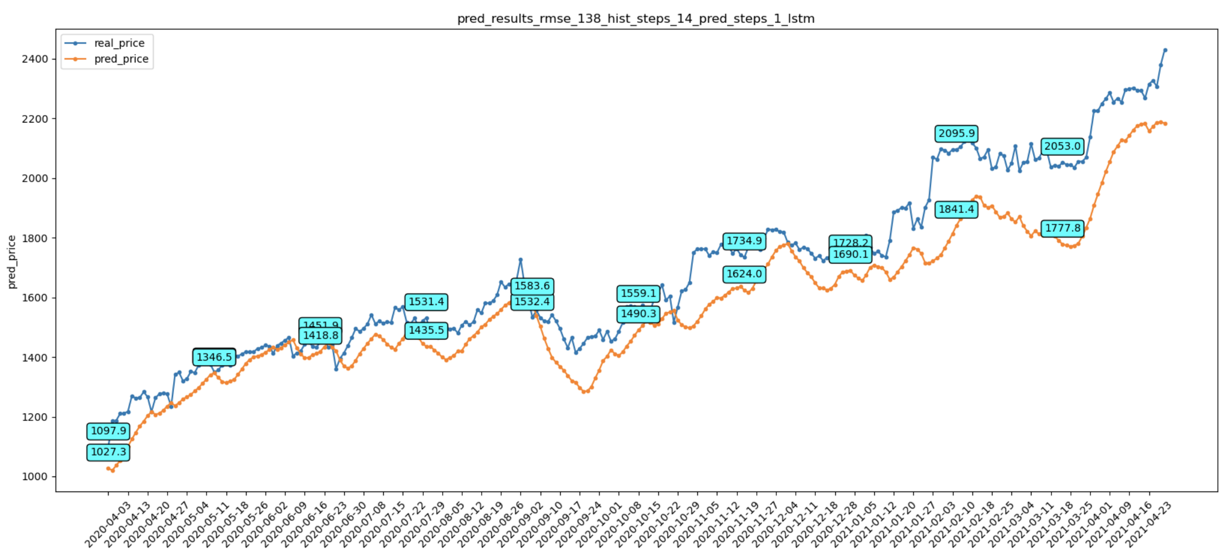


Figure 9. Use Features from Past 14 Days to Predict next 1 Day's Adjusted Close Price (LSTM, RMSE = 138)

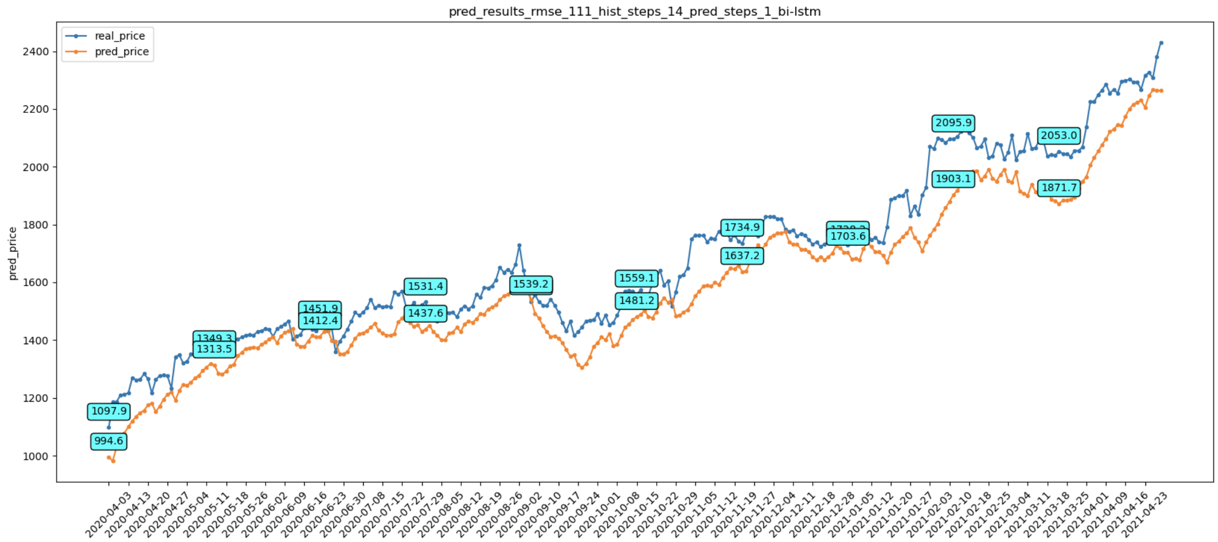


Figure 10. Use Features from Past 14 Days to Predict next 1 Day's Adjusted Close Price (Bi-LSTM, RMSE = 111)

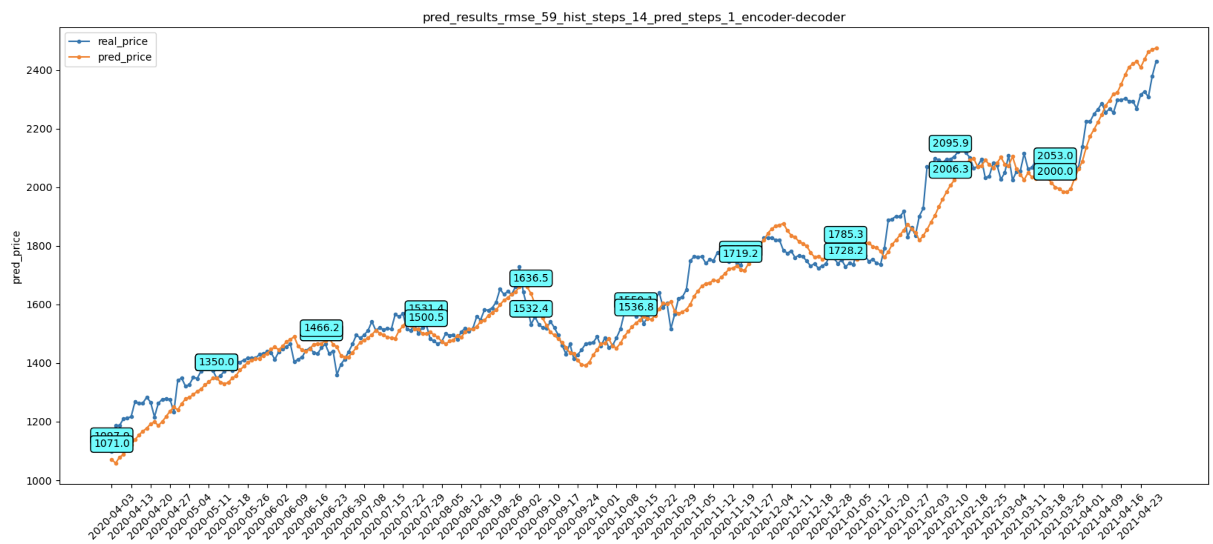


Figure 11. Use Features from Past 14 Days to Predict next 1 Day's Adjusted Close Price (Encoder-Decoder, RMSE = 59)

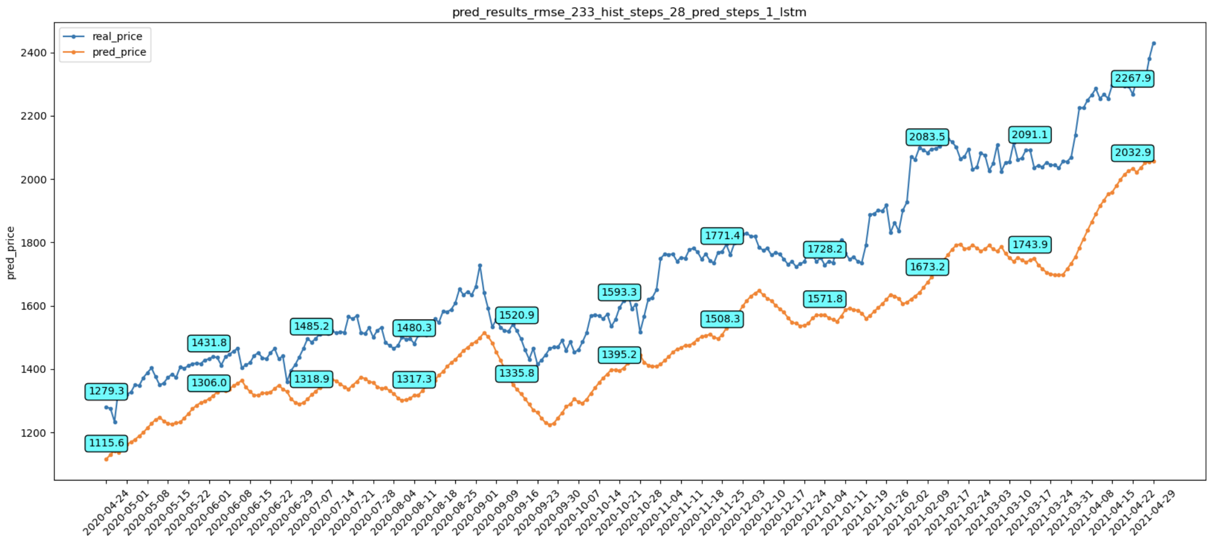


Figure 12. Use Features from Past 28 Days to Predict next 1 Day's Adjusted Close Price (LSTM, RMSE = 233)

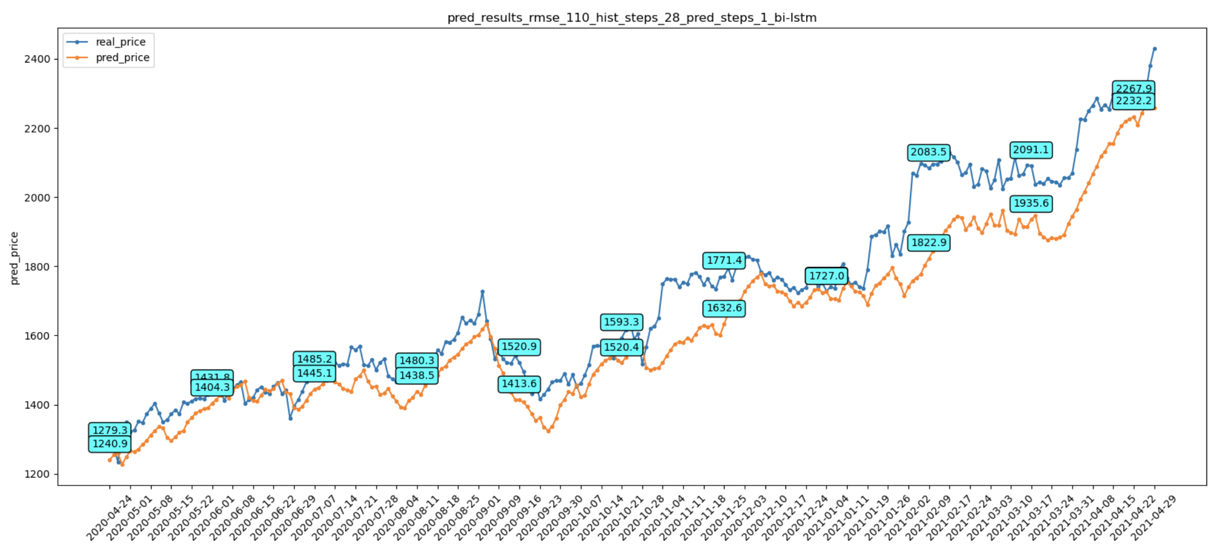


Figure 13. Use Features from Past 28 Days to Predict next 1 Day's Adjusted Close Price (Bi-LSTM, RMSE = 110)

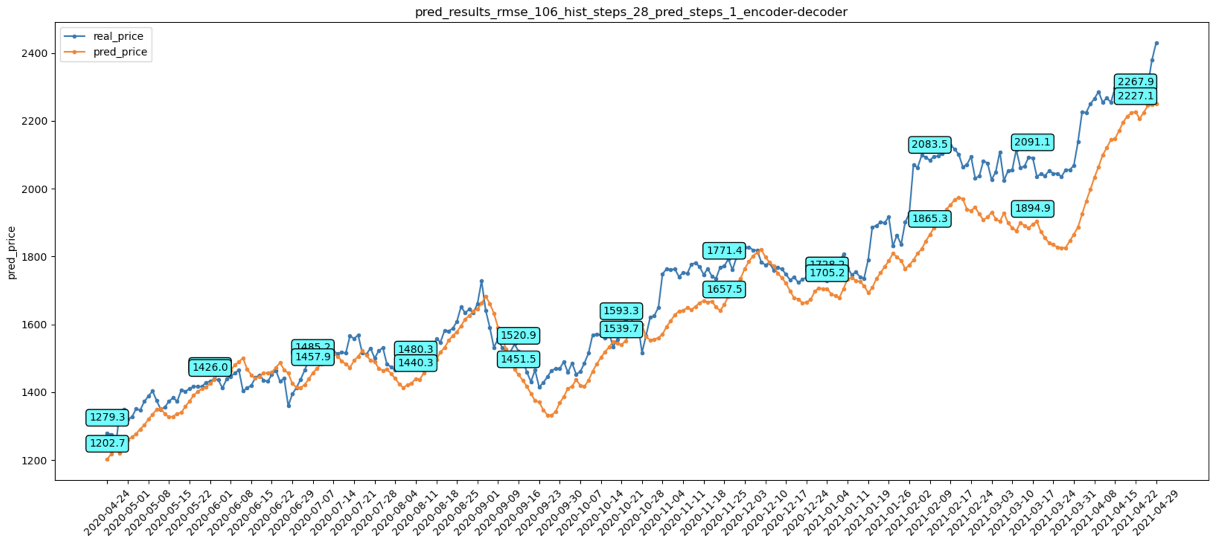


Figure 14. Use Features from Past 28 Days to Predict next 1 Day's Adjusted Close Price (Encoder-Decoder, RMSE = 106)

Experiments to Evaluate Model Performance via Return of Investment

Passive Trading Strategy as a Baseline

Assume we invest $100,000 to buy GOOG stocks at a predefined time (t_start), and then we hold the stocks until a predefined time(t_end), we can calculate the overall assets return as:

Total Assets[t_end] = $100,000 + INT($100,000/price[t_start]) * (price[t_end] - price[t_start])

Simple Trading Strategy

Starting off with $100,000 as the initial investment fund, can we derive a simple trading strategy based on the predicted stock price change from the models to earn money overtime?

A simple trading strategy is implemented which will trigger an event to buy as much stock as possible when there is over a predicted 0.3% increase in price movement for the next day, and trigger an event to sell as much stock as possible when there is over a predicted 1% decrease in price movement for the next day. See sample logs of trading events below which are generated from the implemented trading strategy.

Assets Return via Applying Trading Strategy based on Stock Price Prediction

For each of the model architecture, we have built and trained model via the following experiments:

- Use features from past 7 Days/14 Days/ 28 Days as inputs and use next day's Adjusted Close Price as target to build and train a model; deploy the model to predict stock price using the test dataset; trade the stocks based on predicted stock price by following the predefined trading strategy; calculate the total final assets and the final owned stock shares in the asset; compare the return of investment to the passive trading strategy.

Experiments and analysis results are summarized in Table 3 to compare the return of investment among the three LSTM based models (use GOOG stock price data from April 2020-April 2021).

	LSTM		**Bi-LSTM**		**Encoder-Decoder**	
Model in the Experiment	**Total Final Assets**	**Final Stock Shares**	**Total Final Assets**	**Final Stock Shares**	**Total Final Assets**	**Final Stock Shares**
Use Features from Past 7 Days to Predict next Day's Adjusted Close Price	**$321,032** +46% wrt passive strategy	**132 shares** +42 shares wrt passive strategy	**$511,346** +133% wrt passive strategy	**210 shares** +120 shares wrt passive strategy	**$380,814** +74% wrt passive strategy	**156 shares** +66 shares wrt passive strategy
Use Features from Past 14 Days to Predict next Day's Adjusted Close Price	**$325,346** +47% wrt passive strategy	**133 shares** +43 shares wrt passive strategy	**$528,599** +139% wrt passive strategy	**217 shares** +126 shares wrt passive strategy	**$363,608** +64% wrt passive strategy	**149 shares** +58 shares wrt passive strategy
Use Features from Past 28 Days to Predict next Day's Adjusted Close Price	**$239,393** +26% wrt passive strategy	**98 shares** +20 shares wrt passive strategy	**$404,222** +113% wrt passive strategy	**166 shares** +88 shares wrt passive strategy	**$298,888** +58% wrt passive strategy	**123 shares** +45 shares wrt passive strategy

Table 3. Summary of Experiment Results for Return of Investment

Note: While the buy threshold and the sell threshold used in the trading strategy will impact the final results and therefore they should be further optimized via a more rigorous approach, the key here is to apply the same trading strategy to compare the performance of the three models by looking at the return of investment.

See Figure 15 for sample logs of trading events and transaction history by following the implemented trading strategy using the predicted stock price from machine learning models.

```
Date : 2020-03-25   Activity: BUY    Stocks Trade: 90   Total Stocks: 90   Price: 1102   Total Assets: $100000
Date : 2020-03-26   Activity: HOLD   Stocks Trade: 0    Total Stocks: 90   Price: 1161   Total Assets: $105333
Date : 2020-03-27   Activity: HOLD   Stocks Trade: 0    Total Stocks: 90   Price: 1110   Total Assets: $100739
Date : 2020-03-30   Activity: HOLD   Stocks Trade: 0    Total Stocks: 90   Price: 1146   Total Assets: $103989
Date : 2020-03-31   Activity: HOLD   Stocks Trade: 0    Total Stocks: 90   Price: 1162   Total Assets: $105428
Date : 2020-04-01   Activity: SELL   Stocks Trade: 90   Total Stocks: 0    Price: 1105   Total Assets: $100281
Date : 2020-04-02   Activity: HOLD   Stocks Trade: 0    Total Stocks: 0    Price: 1120   Total Assets: $100281
Date : 2020-04-03   Activity: BUY    Stocks Trade: 91   Total Stocks: 91   Price: 1097   Total Assets: $100281
Date : 2020-04-06   Activity: HOLD   Stocks Trade: 0    Total Stocks: 91   Price: 1186   Total Assets: $108384
Date : 2020-04-07   Activity: HOLD   Stocks Trade: 0    Total Stocks: 91   Price: 1186   Total Assets: $108347
Date : 2020-04-08   Activity: HOLD   Stocks Trade: 0    Total Stocks: 91   Price: 1210   Total Assets: $110510
Date : 2020-04-09   Activity: HOLD   Stocks Trade: 0    Total Stocks: 91   Price: 1211   Total Assets: $110616
Date : 2020-04-13   Activity: HOLD   Stocks Trade: 0    Total Stocks: 91   Price: 1217   Total Assets: $111172
Date : 2020-04-14   Activity: HOLD   Stocks Trade: 0    Total Stocks: 91   Price: 1269   Total Assets: $115874
Date : 2020-04-15   Activity: HOLD   Stocks Trade: 0    Total Stocks: 91   Price: 1262   Total Assets: $115259
Date : 2020-04-16   Activity: HOLD   Stocks Trade: 0    Total Stocks: 91   Price: 1263   Total Assets: $115350
Date : 2020-04-17   Activity: HOLD   Stocks Trade: 0    Total Stocks: 91   Price: 1283   Total Assets: $117150
Date : 2020-04-20   Activity: HOLD   Stocks Trade: 0    Total Stocks: 91   Price: 1266   Total Assets: $115636
Date : 2020-04-21   Activity: HOLD   Stocks Trade: 0    Total Stocks: 91   Price: 1216   Total Assets: $111061
Date : 2020-04-22   Activity: HOLD   Stocks Trade: 0    Total Stocks: 91   Price: 1263   Total Assets: $115326
Date : 2020-04-23   Activity: HOLD   Stocks Trade: 0    Total Stocks: 91   Price: 1276   Total Assets: $116518
Date : 2020-04-24   Activity: HOLD   Stocks Trade: 0    Total Stocks: 91   Price: 1279   Total Assets: $116791
Date : 2020-04-27   Activity: SELL   Stocks Trade: 91   Total Stocks: 0    Price: 1275   Total Assets: $116479
Date : 2020-04-28   Activity: HOLD   Stocks Trade: 0    Total Stocks: 0    Price: 1233   Total Assets: $116479
Date : 2020-04-29   Activity: BUY    Stocks Trade: 86   Total Stocks: 86   Price: 1341   Total Assets: $116479
Date : 2020-04-30   Activity: HOLD   Stocks Trade: 0    Total Stocks: 86   Price: 1348   Total Assets: $117097
Date : 2020-05-01   Activity: HOLD   Stocks Trade: 0    Total Stocks: 86   Price: 1320   Total Assets: $114684
Date : 2020-05-04   Activity: HOLD   Stocks Trade: 0    Total Stocks: 86   Price: 1326   Total Assets: $115217
Date : 2020-05-05   Activity: HOLD   Stocks Trade: 0    Total Stocks: 86   Price: 1351   Total Assets: $117307
Date : 2020-05-06   Activity: HOLD   Stocks Trade: 0    Total Stocks: 86   Price: 1347   Total Assets: $116980
Date : 2020-05-07   Activity: HOLD   Stocks Trade: 0    Total Stocks: 86   Price: 1372   Total Assets: $119152
Date : 2020-05-08   Activity: HOLD   Stocks Trade: 0    Total Stocks: 86   Price: 1388   Total Assets: $120512
Date : 2020-05-11   Activity: HOLD   Stocks Trade: 0    Total Stocks: 86   Price: 1403   Total Assets: $121792
Date : 2020-05-12   Activity: SELL   Stocks Trade: 86   Total Stocks: 0    Price: 1375   Total Assets: $119426
Date : 2020-05-13   Activity: HOLD   Stocks Trade: 0    Total Stocks: 0    Price: 1349   Total Assets: $119426
Date : 2020-05-14   Activity: HOLD   Stocks Trade: 0    Total Stocks: 0    Price: 1356   Total Assets: $119426
Date : 2020-05-15   Activity: BUY    Stocks Trade: 86   Total Stocks: 86   Price: 1373   Total Assets: $119426
Date : 2020-05-18   Activity: HOLD   Stocks Trade: 0    Total Stocks: 86   Price: 1383   Total Assets: $120350
```

Figure 15. Logs of Trading Event Based on the Implemented Trading Strategy

The stock price of GOOG per share is doubled from April 2020 to April 2021, so even with passive trading strategy(keep holding the initial stock shares), the return of investment is very high (~2X).

Detailed trading events using models for each experiment are visualized in Figure 16 - Figure 24. The figures below annotated the event time when a “SELL” or “BUY” event is logged(RED for SELL event, Cyan for BUY event). Based on observation, one can conclude that the trading strategy based on predicted stock prices from the ML models is able to make decisions so that we can **“buy at low”** and **“sell at high” for most of the time**, and the final return of investment outperforms the passive trading strategy((hold the stock shares bought with the initial investment fund and no additional trading activities), with more total final assets and more stock shares to own in the long term.

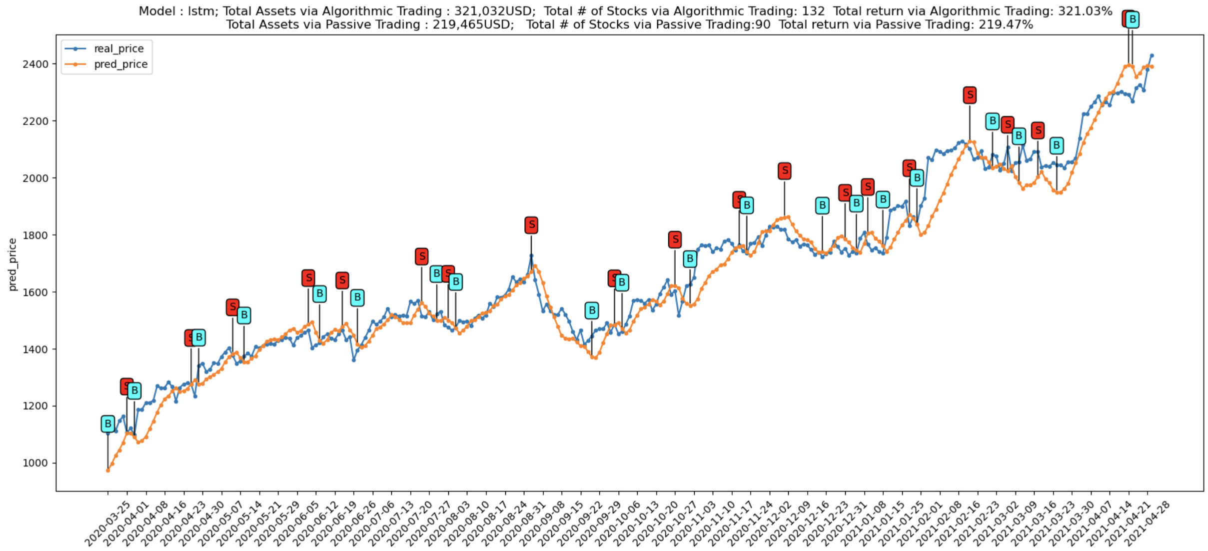


Figure 16. Trading Results based on LSTM Model,
(Model Trained using past 7 days' data to predict next day; total final assets:$321,032)

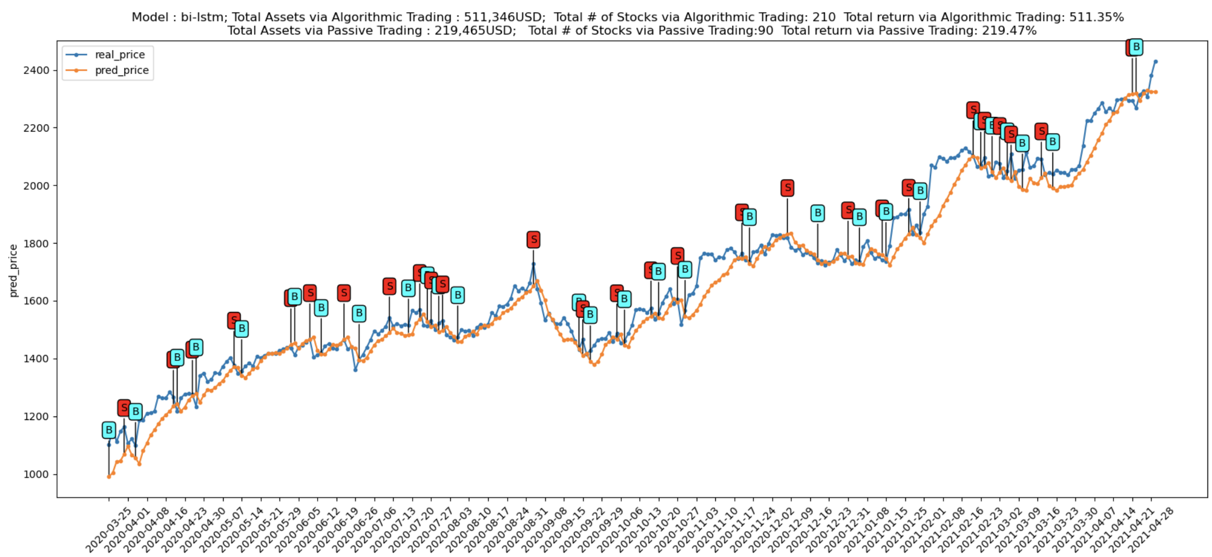


Figure 17. Trading Results based on Bi-directional LSTM Model
(Model Trained using past 7 days' data to predict next day; total final assets:$511,346)

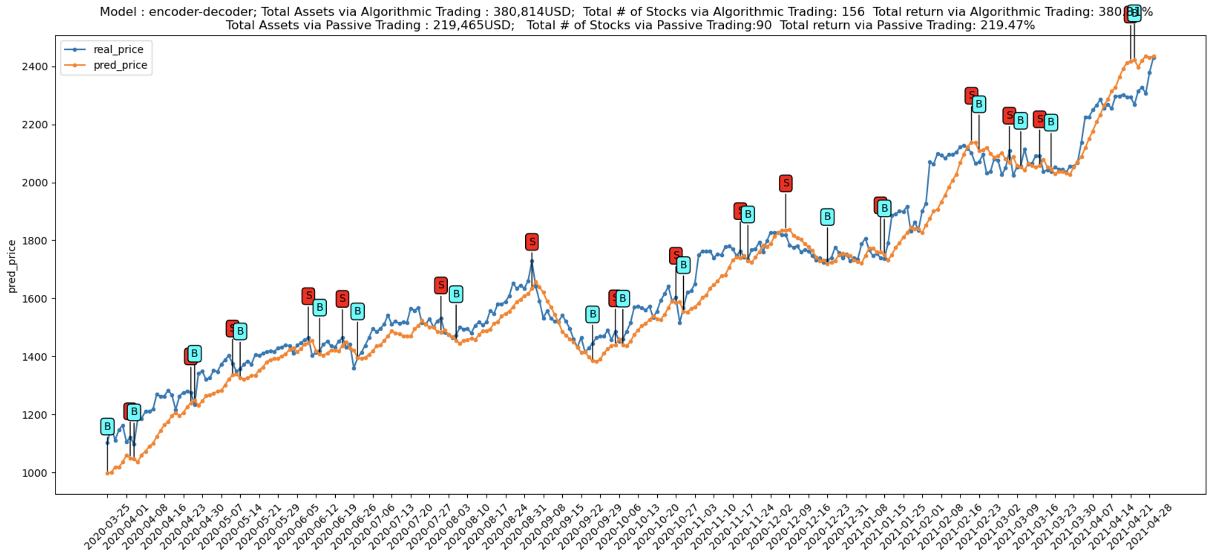


Figure 18. Trading Results based on Encoder-Decoder Model
(Model Trained using past 7 days' data to predict next day; total final assets:$380,814)

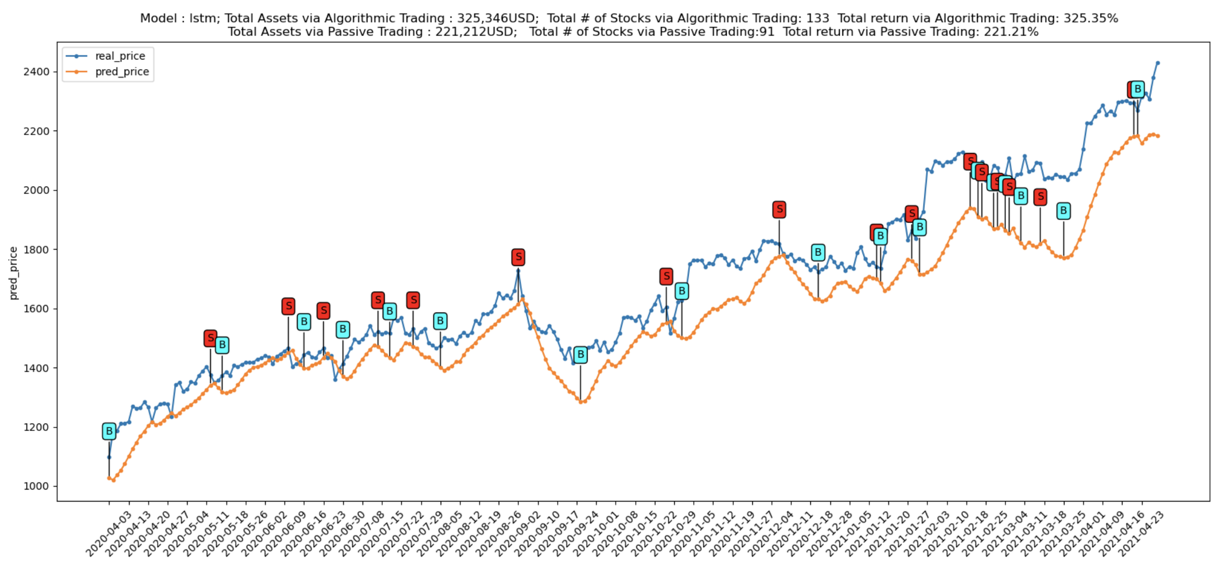


Figure 19. Trading Results based on LSTM Model,
(Model Trained using past 14 days' data to predict next day; total final assets:$325,346)

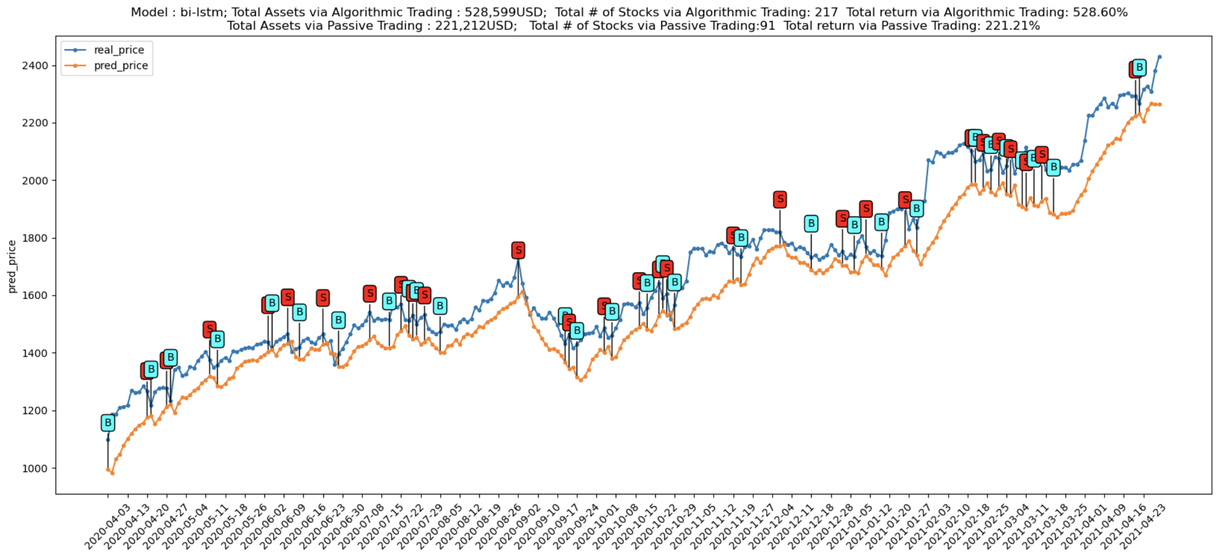


Figure 20. Trading Results based on Bi-directional LSTM Model,
(Model Trained using past 14 days' data to predict next day; total final assets:$528,599)

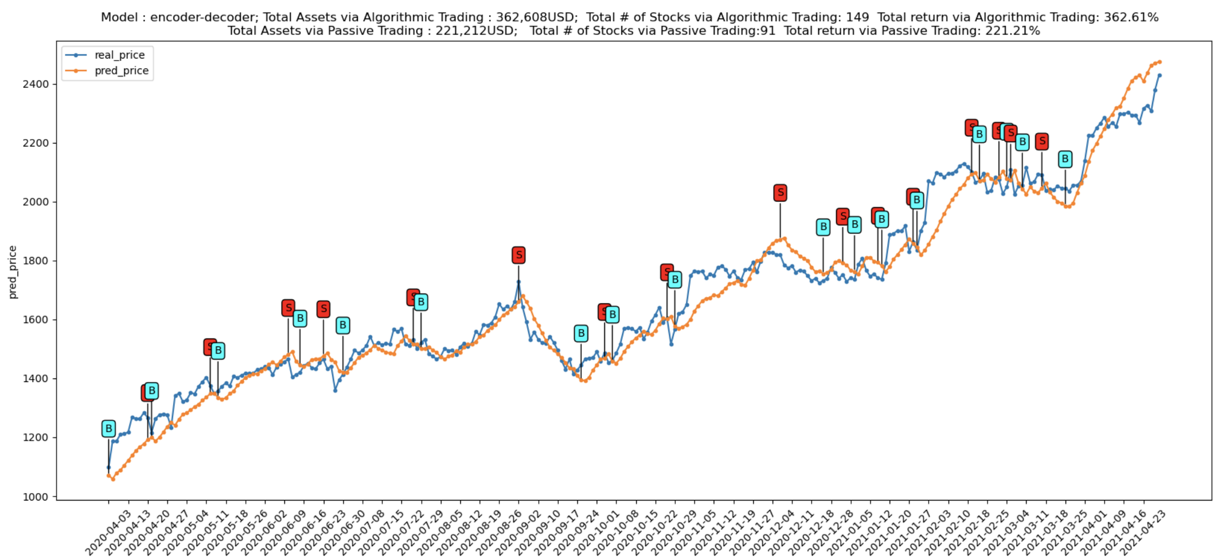


Figure 21. Trading Results based on Encoder-Decoder Model,
(Model Trained using past 14 days' data to predict next day; total final assets:$362,608)

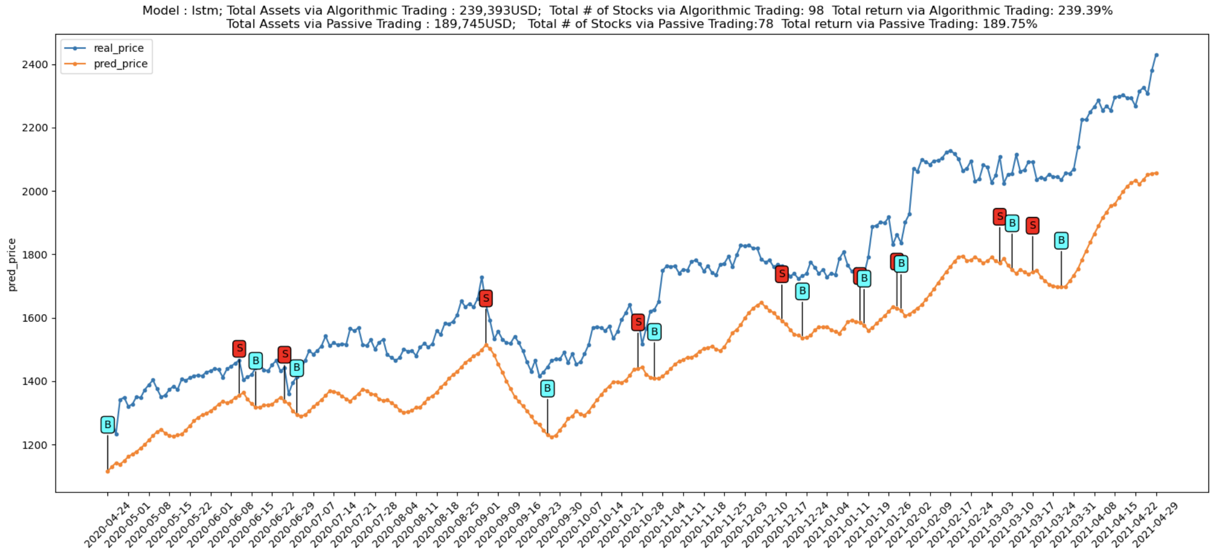


Figure 22. Trading Results based on LSTM Model,
(Model Trained using past 28 days' data to predict next day; total final assets:$239,393)

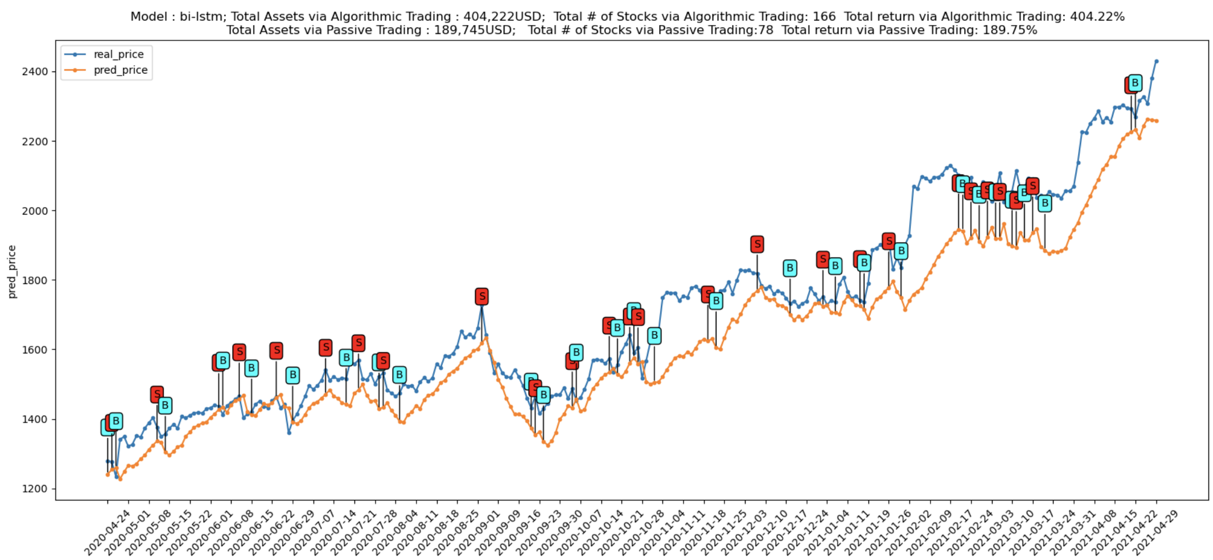


Figure 23. Trading Results based on Bi-directional LSTM Model,
(Model Trained using past 28 days' data to predict next day; total final assets:$404,222)

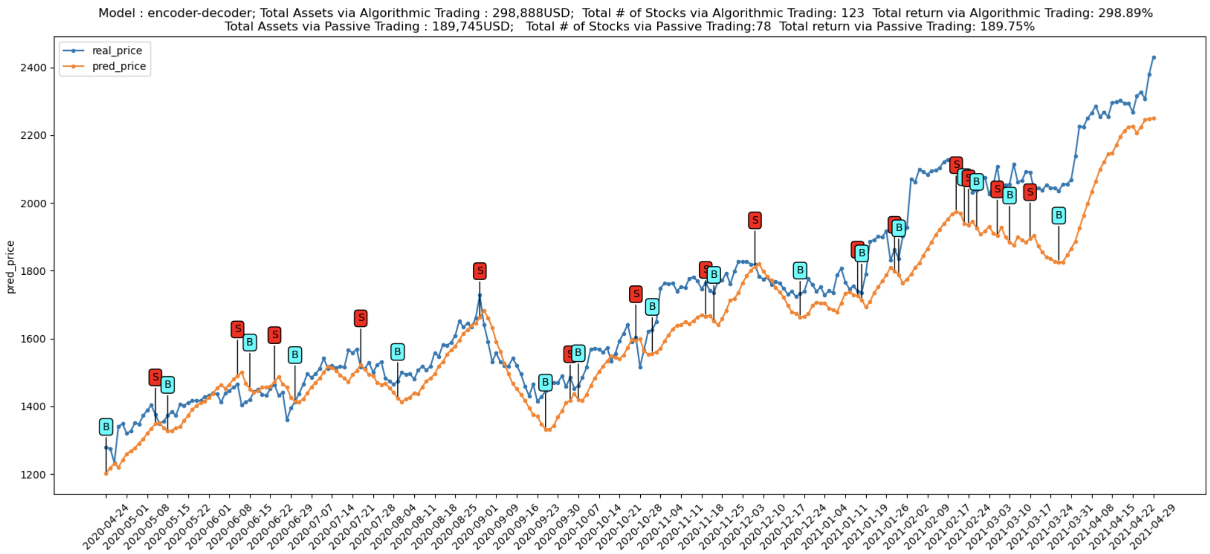


Figure 24. Trading Results based on Encoder-Decoder Model,
(Model Trained using past 28 days' data to predict next day; total final assets:$298,999)

Discussions

Based on the analysis of the experiment results, we can conclude that:

- The return of investments via the trading strategy based on the 3 machine learning models all beat the passive trading strategy. The Bidirectional LSTM model outperforms other models, and the LSTM based encoder-decoder model has a better performance compared to the baseline LSTM model in all experiments.

- Although the LSTM based encoder-decoder model seems to have better performance in terms of stock price prediction error (RMSE), the Bi-directional LSTM seems to have better performance in terms of predicting the percentage of price change so that the trading strategy applied based on model prediction results is able to "sell at high" and "buy at low" more accurately compared with other models, therefore yields better return of investment.

There are many ways to further improve the machine learning models and the overall trading strategy. A few ideas are summarized below:

- Tune the hyper parameters of the LSTM building block and play with drop out rate to optimize the model.

- Add other features such as quarterly earning results of the company to train the model (e.g., in many cases, stock price movement before and after the quarterly earning publication date is impacted by the earning results).

- Add other inputs/constraints to implement a more realistic trading strategy (e.g., add transaction fee when buy/sell stock).

- Implement a more rigorous way to derive the market entry/exit criterion instead of using simple hard-coded thresholds.

References

[1] Christopher Olah, Understanding LSTM Networks
[2] Mojtaba Nabipour, Pooyan Nayyeri, Hamed Jabani, Amir Mosavi, Deep learning for stock market prediction, Quantitative Finance, arXiv:2004.01497v1 [q-fin.ST]
[3] Boris Banushev,Using the latest advancements in deep learning to predict stock price movements, Towards Data Science.
[4] Sarit Maitra, Simple Way of Evaluating Algorithmic Trading Strategy, Medium.
[5] Xiao-Yang Liu, Hongyang Yang, Qian Chen, Runjia Zhang, Liuqing Yang, Bowen Xiao, Christina Dan Wang, FinRL: A Deep Reinforcement Learning Library for Automated Stock Trading in Quantitative Finance, arXiv:2011.09607v1 [q-fin.TR] .
[6] Yang, Hongyang and Liu, Xiao-Yang and Zhong, Shan and Walid, Anwar, Deep Reinforcement Learning for Automated Stock Trading: An Ensemble Strategy (September 11, 2020). Available at SSRN: https://ssrn.com/abstract=3690996.