Stock Market Prediction with Recurrent and Regression Neural Network

Kevin Wang (kevinw@g.ucla.edu)
Yihuan Huang (yihuan0408@gmail.com)
Department of Psychology, 502 Portola Plaza
Los Angeles, CA 90095 USA

Abstract

Predictions within the stock market have proven to be difficult [1] . However, with the help of Recurrent and Regression Neural Networks, we investigated the process in which machine learning can predict future market prices. Real world performance is evaluated with a trading algorithm that follows simple principles in order to assess model predictions. Manipulation on Recurrent network models aim to determine what extent LSTM can handle long term tendency issues and manipulation on the Regression model determined which external factors affect the accuracy of predictions.

Keywords: Stock Market Prediction; Recurrent Network; Regression Network; LSTM

Introduction

Stock Market Frenzy

Changes within the stock market are dependent on numerous economic factors Asset prices in particular are commonly believed to react sensitively to economic news (Chen, Roll, & Ross 1986). Following the COVID-19 pandemic and the market crash in March 2020, an estimated 10 million new brokerage accounts were opened by individuals in 2020 (Tompor 2021). According to The Wall Street Journal, individual investors made up an estimated 19.5% of the U.S. equity trading volume in 2020, compared to 4% in 2019 (Osipovich 2020). With an influx of new investors, market movement has become increasingly harder to predict as individuals now have the ability to trade through easy-to-use brokerage apps like Robinhood. This can be seen when the video game seller GameStop caught the attention of reddit subreddit /r/wallstreetbets which artificially drove the price of GameStop, AMC Entertainment Holdings, Blackberry, and Nokia to all-time highs after the conjoined efforts of individual small-scale investors initiated a short squeeze of institutional investors (Lyócsa et al 2021). With the increased interests of individual investors in the stock market, prediction of stock market movement has not only caught the attention of financial institutions, but also of the common trader. Previously, investments decisions were made through stringent Fundamental Analysis that examines the underlying forces that affect the well-being of the economy to calculate intrinsic value of a company. Additionally, Technical Analysis was used as supplementary analysis to access the market indexes and calculate supply and demand (A.S 2013).

Using Artificial Intelligence

Now, more than ever, is there a desire to be a part of this lucrative trading game. However, there are still many factors to take into consideration when investing due to the volatility of the market and the inherent risk that follows in asset trading (Guthrie 2006). The use of machine learning can help the common investor reduce risk and increase their chances at profiting off their initial investments. Due to the many factors that exist that influence the market, it would be extremely difficult for a single person to take into consideration all the factors on a given day and predict the prices of the stock the following day.

In this paper, we assess the performance of Recurrent Neural Network predictions (RNN) using Long Short Term Memory (LSTM) as well as investigate the properties that influence asset value through a Regression Neural Network. The model is trained on historic Amazon (AMZN) stock prices from 05/10/2017 to 04/01/2020 and the prediction performance is evaluated from the training end date to the end year (12/31/2020).

Methods

Recurrent Neural Networks

The applications of Recurrent Neural Networks are numerous; this neural network can solve a variety of problems such as speech recognition, language modeling, translation, image captioning, and so forth. What differentiates a RNN from a traditional neural network is its ability to handle sequential, time-sensitive data, due to the basic structure of RNNs having a loop that allows time-series to be passed from one step of the network to the next. With the additional use of Long Short Term Memory within our RNN, we are able to solve some of the performance issues that RNN models have on long-term dependency. LSTM was introduced by Hochreiter & Schmidhuber (1997). We use LSTM because it can remove or add information to the cell state and regulated by structures called gates. While input data travels through the cells within LSTM, forget gates determine what information to be forgotten and the cell state is updated with the hidden neurons to create a new cell state that will combine with sequential data to create the final output layer. The purpose

[1] Economic Forces and the Stock Market (Chen, Roll, & Ross 1986)

of the forget gates is to regulate overfitting of the training data as the RNN in total is trying to recognize the pattern in which the historic stock prices are moving. Because the stock market moves unpredictably, we the forget gates tries to prevent the network from learning incorrect patterns and using it to make future predictions as there is no guarantee that the stock market will perform in the same pattern as it did in the past.

Data Preparation Historic High, Low, Open, and Close stock data on AMZN prices from the start date of 05/10/2017 to the training end date of 04/01/2020 were used as the training data set for RNN. Each sequence that is passed in for training consists of 20 days of historic data with the forementioned 4 features. The model will predict the 21st day's values based on the patterns of the previous 20 days. Table 1.1 shows the parameters that the model is using to train on with 4 hidden LSTM layers and 4 interwoven dropout layers, the final output is a dense layer with 4 units corresponding with the desired features. Prior to training, all features are scaled using MinMaxScaler with feature ranges (-1,1).

Table 1.1
Recurrent Neural Network Model Parameters

Layer (type)	Output Shape	Param #
lstm (LSTM)	(None, 20, 80)	27200
dropout (Dropout)	(None, 20, 80)	0
lstm_1 (LSTM)	(None, 20, 40)	19360
dropout_1 (Dropout)	(None, 20, 40)	0
lstm_2 (LSTM)	(None, 20, 40)	12960
dropout_2 (Dropout)	(None, 20, 40)	0
lstm_3 (LSTM)	(None, 40)	12960
dropout_3 (Dropout)	(None, 40)	0
dense (Dense)	(None, 4)	164

Total params: 72,644
Trainable params: 72,644
Non-trainable params: 0

RNN Manipulation To fully understand how the RNN learns its patterns, we manipulated the input sequences into the model and evaluated its performance.
Predicting the period in between two training sets The model is trained on the data from 11/28/2017 – 09/01/2018 and 04/01/2020 – 12/31/2020 and tested on the middle (untrained) period from 09/01/2018 – 04/01/2020.
Multiplying by Scaler Multiply the input sequences by 0.5
Introducing noise and ambiguity Instead of feeding in 20 consecutive days we instead feed in every other day over a 40-day period.
Feed Backwards To test whether LSTM can truly handle Long Term Dependency issues, the input sequences were reversed.
Rotate Input Sequences On the same note of testing the limits of long-term dependency, we tested the effects of rotating the first input (distance = 1) to the last index and shifting elements to the left. This is repeated for distances of 5 and 10 by rotating the first 5 inputs to the end and the first 10 inputs to the end.
Reverse Early Sequences The reversal of sequences was done by reversing the first 5, 10, and 15 inputs within the 20-day sequence and keep the latter unchanged sequences in their respective indexes.

Regression Neural Networks

While we can hope that investors make rational decisions within the market to maximize profits[2], it has been shown that even professional investors make irrational decisions[3] . According to Ritter, the irrationality stems from people being "bad Bayesians" and thus even if the RNN were to learn the patterns perfectly, there is no guarantee that those patterns will pronounce themselves in the future. Since the pattern of past prices is not sufficient in generating future predictions, we incorporated a Regression Neural Network to act as the supplementary analysis to aid our prediction. While the RNN uses blocks of 20 past days to predict, the Regression model adds in external factors that may affect the stock price on any given day. The Regression features are available on the 20th day to promote the model prediction closer to the actual prices of the next day. Table 1.2 shows the parameters used in the base Regression Model.

Table 1.2
Regression Neural Network Model Parameters

Model: "sequential_36"

Layer (type)	Output Shape	Param #
dense_176 (Dense)	(None, 32)	576
dense_177 (Dense)	(None, 64)	2112
dense_178 (Dense)	(None, 64)	4160
dense_179 (Dense)	(None, 124)	8060
dense_180 (Dense)	(None, 4)	500

Total params: 15,408
Trainable params: 15,408
Non-trainable params: 0

Features The external factors that we added to improve our predictions can be categorized into groups: Market Indicators, Trends, Sentiment Analysis, Relevant Tickers, and Economic Indicators. Again, all features are trained after they have been scaled to the range (-1,1).
The Market Indicators include Upper/Lower Volatility, Short/Long Resistance, and Short/Long Support. Volatility is defined as 3% of a given day's moving average added to

[2] Efficient Market Hypothesis (Malkiel 1989)
[3] Behavioral Finance (Ritter 2003)

itself (+3% for upper, - 3% for lower). Resistance is calculated by taking the maximum 'High' value within a period of time (past 3 days for short resistance, past 7 days for long resistance). Conversely, support is calculated by taking the minimum 'Low' value (3 days for short support, 7 days for long support).

Trends were taken from Google stock trends API.

News sentiment was done through the NLTK sentiment analyzer python library and applied to news articles taken from the Finviz stock news database.

Relevant Tickers are the 10 most highly correlated stocks with AMZN

Economic Indicators include historic data from S&P 500 and Dow Jones indexes.

Regression Manipulation As the Regression model attempts to consider the external factors, manipulation is done through removing or adding features for the model to regress on. As not all information is necessary and not all features have the same importance on its effect on the price, trials were run on the regression model with the removal of certain features by category.

Testing Accuracy and Performance Evaluation

After training the model, the predicted results are tested for accuracy on the actual stock prices from 04/01/2020 to 12/31/2020 (unless specified otherwise). As both model output 4 features 'High', 'Low', 'Open', and 'Close', we assess the accuracy based on how close the predicted 'Close' prices match to the actual 'Close' prices on a particular day. The mean squared error is calculated by taking the scaled (-1,1) predicted 'Close' values and comparing it to the scaled actual 'Close' values of that day. Additionally, real world performance evaluation is done by implementing a simple trading strategy that operates on 3 rules: (1) is the predicted value for the next day is higher than today's price, then buy 1 share. (2) Otherwise, sell 1 share since prices are predicted to depreciate. (3) At the end of the trading period, sell all shared and calculate % return on initial investment. This algorithm is supplied to the predicted 'Close' values that the model outputs. You can supply the algorithm with the starting initial balance and the number of stocks to buy if the prediction is higher or lower (in our case quantity = 1 share). The actual closing prices within the trading period is run through this algorithm to provide a baseline comparison of how well this trading policy performs in terms of % return on investment given the real prices. Both the RNN model predictions and Regression Model are run through the same algorithm for comparisons.

Results

Performance Evaluation

With the simple trading algorithm, the expected percent return on initial investment with the starting balance of $10,000 and the quantity specified to buy/sell to 1, with the information of the real stock prices is 100.99%. Taking the predicted values from the RNN prediction and comparing it to the Regression model, the Regression model performed 2.4 times better than the RNN model. The Regression model's testing MSE was 10.9 times more accurate (0.0408) than the testing MSE from RNN's prediction (0.448). Table 2 summarizes the performance of the base models prior to any input manipulation.

Table 2

Base Models Performance Summary

Model	% Return	Train MSE	Test MSE
Actual Price	100.99%	N/A	N/A
RNN	25.52%	0.0040	0.448
Regression	62.87%	0.00086	0.0408

Figure 1.1 Shows the RNN predicted values compared to the actual values within the testing period. The bolded lines indicate the 'Close' values. Figure 1.2 Shows the Regression predicted 'Close values compared to the actual 'Close' values.

Figure 1.1: RNN Predicted vs Actual

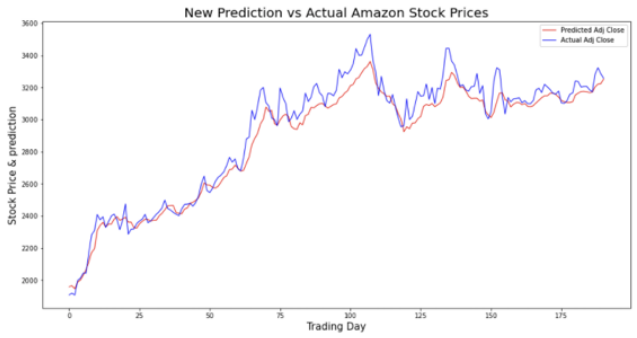

Figure 1.2 Regression Predicted Close vs Actual Close

Figure 2.1 Shows the Portfolio Performance of the trading algorithm using the actual 'close' values. Figure 2.2 shows

the RNN Stock Portfolio Performance and Figure 2.3 shows the Regression Model's Portfolio Performance.

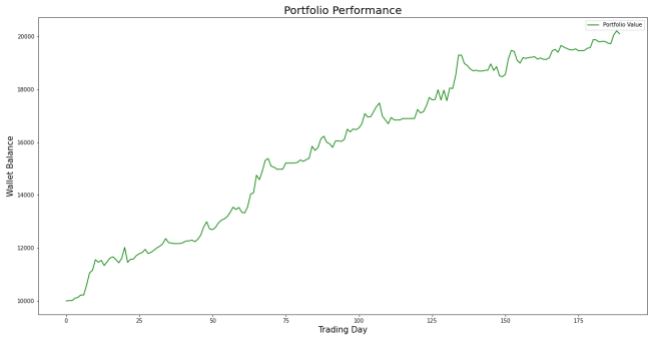

Figure 2.1: Actual Prices Performance

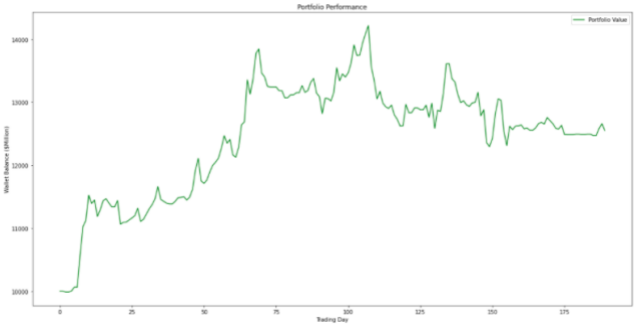

Figure 2.2: RNN Performance

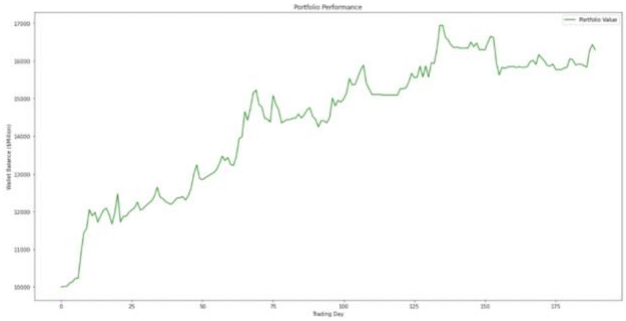

Figure 2.3: Regression Performance

RNN Manipulation Outcomes

Predicting the period in between two training sets When we deliberately fed in training data that consists of the rapidly increasing and decreasing periods and asked our model to predict periods that are of similar patterns but on a smaller scaler, our model actually predicts the middle period pretty well (Testing MSE = 0.0041).

Multiplying by Scaler When multiply the 20 input sequences by a scaler of 0.5, the slope of the line connecting day 20 and the prediction target becomes larger, and our model now learns more rapidly increasing patterns than the original ones. The prediction gives higher values than initially predicted and is more accurate (Testing MSE = 0.1227)

Feeding Every other Day When introducing ambiguity and noise to the input and feeding every other day of the input sequence over 40 days instead of 20 consecutive days, the predicted values from only the even days differ greatly from the predicted values of the odd days.

Feed Backwards When the sequence is reversed, the model can still construct the training set, however, the predictions on the testing set are way off.

Rotate Input Sequences When only day 1 is rotated to the end, the overall prediction was the best compared to when the distance is increased to 5. At distance = 5, the prediction is flatter and misses the peaks. When the first 10 inputs in the sequence are rotated to the end, the model breaks and the trend is not captured at all.

Reverse Early Sequences When only the first 5, 10, or 15 sequences are reversed the model still captures the overall trend but worsening as more early sequences are reversed. This performs better than when entire sequences are reversed, and we conclude that the last 5 days of input are the most important in generating predictions.

Regression Manipulation Outcomes

Table 3 summarizes the effect of removing certain Regression features and the relative performance compared to the base model with all initially considered features.

Economic Indicators The removal of Economic Indicators like S&P 500 and Dow Jones slightly improved the predictions (2%). The reason for the increase in accuracy is because the overall performance of the economy captured by these indices may not be entirely representative of AMZN stock specifically.

Relevant Tickers If we removed AMZN's 10 most correlated stocks from our regression list, then the prediction is worse. Without the relevant tickers, the peaks are not as well predicted and the predicted prices are not as precise, making the overall predicted values 'smoother' across the testing period.

Sentiment Analysis Sentiment related features such as Trends and News give varying results when removed. Further testing over more simulated runs must be done to make a conclusion on why it sometimes improves the predictions and other times worsens the predictions. However, we propose that the trends were not useful in prediction because Google's API takes trend hit from a period of 50 days while our model is not within the same window frame.

Most Important Features Without Market Indicators, the regression model breaks, and the predictions are useless (MSE = 3.59). This emphasizes the importance of features such as resistance, support, and volatility previously

unconsidered within the RNN model. The best results were achieved when Trends and Economic indicators were removed together (MSE = 0.009).

Table 3
Effects of Removing Regression Features

Feature Removed	Test MSE	Relative Performance (from base)
Economic Indicators	0.040	2% Better
Relevant Stocks	0.052	27% Worse
News Sentiment	0.045	10% Worse
Trends	0.048	17% Worse
Market Indicators	3.59	8800% Worse
Trends & Economic Indicators	0.009	80% Better

Discussion

Conclusion

Recurrent Neural Networks using LSTM can solve long-term dependency issues, but only to a certain extend as it has been shown that when reversing the entire sequence, the RNN model breaks but up until reversing the first 15 sequences, the overall pattern is relatively captured.
Regression Models are heavily dependent on features that relevant and features that are not as important may decrease prediction accuracy.
Due to dropout layers preventing the overfitting of RNN models, predictions from each iteration of running the program gives slightly variable results. Dropout layers are also essential for LSTM models as they prevent the model from learning incorrect patterns caused by irrational traders. Many dense layers are required for Regression Models to converge on the correct answer.
Despite our efforts, since results are variable, it is hard to put 100% trust into any model prediction. However, from this analysis, we can conclude that while historical data analysis may not predict entirely correct based off pattern alone, the order in which the sequences are presented are highly important. Since there already exist innate noise from stock data, introducing more noise throws the entire prediction off. Additionally, the ability for the RNN model to predict somewhat accurately with only 4 features tells us that it's not impossible to predict the future prices. With the help of additional regression features and sufficient market knowledge, a great deal of money can be made from successful prediction of the market.

Future Directions

In this analysis we only generated predictions for the later three quarters of the 2020 year, in the future, we would like to implement this model to be able to update and capture data leading up to the present day in order to predict a value for tomorrow. The optimization of the layers and neurons within each model should also be examined as more efficient parameters significantly increases training time required by the model. Initially when the sequence size fed into the RNN model was past 50 days instead of 20, the model takes hours to train.
Potential future question to investigate include analyzing the difference between Wall Street stocks and Blockchain Cryptocurrency as there are not the same rules and regulations in crypto as there are in an official government sanction market. Could the same psychological patterns and irrational trading behaviors exist in different contexts? Could two networks performing the same job communicate with each other to enhance performance? If a neural network with real capital could actively contribute to the market, could it learn over time the best trading strategy through training?

Acknowledgments

Special thanks to Professor Zili Liu and TA Mac Xing at UCLA for facilitating this analysis.

References Instructions

Follow the APA Publication Manual for citation format, both within the text and in the reference list, with the following exceptions: (a) do not cite the page numbers of any book, including chapters in edited volumes; (b) use the same format for unpublished references as for published ones. Alphabetize references by the surnames of the authors, with single author entries preceding multiple author entries. Order references by the same authors by the year of publication, with the earliest first.

Use a first level section heading, "**References**", as shown below. Use a hanging indent style, with the first line of the reference flush against the left margin and subsequent lines indented by 1/8 inch. Below are example references for a conference paper, book chapter, journal article, dissertation, book, technical report, and edited volume, respectively.

References

A.S, S. (2013). A Study on Fundamental and Technical Analysis . *International Journal of Marketing*.

Chen, J. (2020, December 31). *2020 Was a Big Year for Individual Investors*. Investopedia.

https://www.investopedia.com/2020-was-a-big-year-for-individual-investors-5094063.

Chen, N.-F., Roll, R., & Ross, S. (1986). Economic Forces and the Stock Market. *Journal of Business*, *59*(3).

Guthrie, G. (0AD). *Regulating Infrastructure: The Impact on Risk and Investment*. Journal of Economic Literature. https://www.aeaweb.org/articles?id=10.1257%2Fjel.44.4.925.

Lyócsa, Š., Baumöhl, E., & Tomáš, V. (2021). *YOLO trading: Riding with the herd during the GameStop episode*. https://www.econstor.eu/bitstream/10419/230679/1/YOLO-Trading-Riding-with-the-herd-during-the-Gamestop-episode.pdf.

Malkiel B.G. (1989) Efficient Market Hypothesis. In: Eatwell J., Milgate M., Newman P. (eds) Finance. The New Palgrave. Palgrave Macmillan, London. https://doi.org/10.1007/978-1-349-20213-3_13

Osipovich, A. (2020, August 31). *Individual-Investor Boom Reshapes U.S. Stock Market*. The Wall Street Journal. https://www.wsj.com/articles/individual-investor-boom-reshapes-u-s-stock-market-11598866200.

Ritter, J. R. (2003, July 30). *Behavioral finance*. Pacific-Basin Finance Journal. https://www.sciencedirect.com/science/article/pii/S0927538X03000489?casa_token=XvJuBW6EZp4AAAAA%3AzmBAf5zG84II-CNW0VrVwHanQybfJ-kYkq37UhFJrU4f07dQRuKIcM8MN7yAaoS_ts_-tidI1Q.

Tompor, S. (2021, February 6). *Why new investors bought stock during the COVID-19 pandemic*. Detroit Free Press. https://www.freep.com/story/money/personal-finance/susan-tompor/2021/02/05/how-invest-stock-market/4360276001/.