Stock Price Fluctuation Prediction using Machine Learning and Sentiment Analysis

Juanita Gonzalez Fagua
Department of Economics
Carleton University
Ottawa ON CA
juanitagonzalezfagua@cmail.carleton.ca

Md Rezwan Hassan Khan
Department of Computer Science
Carleton University
Ottawa ON CA
mdrezwankhan@cmail.carleton.ca

ABSTRACT

Stock prices depend on a variety of factors, one of which is information. The Efficient Market Hypothesis (EMH) states that stock prices are a representation of all available information in the market [1]. Sentiment analysis through Twitter is a method of determining how individuals feel about a company while producing real time data. This study implements a sentiment analysis with Twitter data on four big technology companies namely: YouTube, Twitch, Apple, and Netflix from July 2020 to October 2020. The sentiment analysis provides a base for public emotion and is utilized in three stock prediction models using a Long Short-Term Memory (LSTM) model. The models employed are a Sentiment LSTM, Multivariate LSTM, and Price Fluctuation LSTM. The best performing model is the Multivariate LSTM resulting with a smaller loss than the Sentiment LSTM. The inaccuracy of the Sentiment LSTM proceeds this paper to conclude that the EMH holds in this study.

KEYWORDS

LSTM, Sentiment Analysis, Twitter, Stock Price

1. INTRODUCTION

The Efficient Market Hypothesis (EMH) states that stock prices trade at fair value because new information is known to all agents in the market. In theory, the stock market performs in a rational and anticipated manner where price prediction yields inaccurate results [2]. However, behavioural economics demonstrates that agents are subject to emotional responses that can violate EMH. Strictly speaking, public emotion towards a company could change that company's stock price in an unforeseen manner. This suggests that a relationship between public emotion and stock prices may exist, and if such a relationship exists could public sentiment then be used to predict future stock values. Public emotion towards a company can be measured using Twitter. Twitter provides a costless and convenient way to acquire new and real time information of public mood compared to other tedious methods such as surveys [1]. Twitter allows users to express their feelings about any topic in 280 characters or less per single Tweet. This social media platform is a very useful database that can examine groups of users through hashtags.

Collecting Tweets from Twitter and extracting public mood has been utilized greatly in past research, particularly in stock price prediction. The objective of this study is to investigate whether the EMH holds or not by implementing a sentiment analysis with Twitter data regarding four big technology companies, notably YouTube, Twitch, Apple, and Netflix. The sentiment analysis provides a base for public emotion and is utilized in three stock prediction models for each company, a Multivariate LSTM, a Sentiment LSTM and a Price Fluctuation LSTM. The prediction models utilize a Long Short Term Memory (LSTM) framework using various Python libraries. The time period of this study spans from July 2020 to October 2020. The Multivariate LSTM model utilizes only past stock values to predict future stock prices, while the Sentiment LSTM model only uses sentiment value. Lastly, the Price Fluctuation LSTM model predicts stock price fluctuation using sentiment value. This study expands upon the work conducted by Cakra et al. [2015] using recently acquired data with a longer time frame than their original research and implementing a different machine learning model.

Given the recent pandemic and worldwide lockdowns, more people have been using online platforms [3] and it is of interest of this study to examine if this increase in social media yields improved results. This paper will be organized in the following manner. Section 2 presents a review of relevant literature. Section 3 is a description of the datasets utilized in this research. Section 4 includes an explanation and description of the methods used in this study. The results and implications are presented in Section 5. Limitations are discussed in Section 6. Lastly, a brief conclusion of this study is completed in Section 7 and subsequentially, further work is explored in Section 8.

2. LITERATURE REVIEW

Social media platforms contain significant information about any subject matter. It should be of interest to any business to utilize these databases to gather vital consumer information. Various research has been conducted on how social media affects stock prices and aggregate behavior in the market. Cakra et al. [2015] further explored this concept in their paper, *Stock Price Prediction using Linear Regression based on Sentiment Analysis,* by assessing how Twitter sentiment can be a

predictive measure of future stock prices for thirteen Indonesian based companies.

The authors collected stock price data from Yahoo Finance CSV API and Tweets using the Twitter REST API over the span of seven days. The Tweets collected included the companies' official Twitter handle or discussed any of the company's products. To extract sentiment from the gathered Tweets, the authors tokenized each Tweet. This is the process of splitting each Tweet into individual words or phrases. They then formalized individual words (lexicons) using the Indonesian dictionary and accounted for sentiment shifters. Where sentiment shifters are words that include contradictions such as "don't". The authors employed five different machine learning algorithms to classify the Tweets given the lexicons and sentiment shifters. They used Support Vector Machine (SVM), Naïve Bayes, Decision Tree, Random Forest, and a single layered Neural Network. From the various machine learning models, Random Forest and Naïve Bayes resulted with the highest accuracy of 60.39% and 56.5% respectively. The authors chose these two models to then classify the entire Twitter dataset. The Tweets were classified as either positive, negative or neutral, and the percentage of positive Tweets was computed per day using the following formula [2].

$$\frac{\sum Positive\ Tweets\ on\ day\ x}{\sum All\ Tweets\ on\ day\ x} * 100$$

The authors conducted the three following models. The price fluctuation prediction model examined if stock prices would increase or decrease based on the previous day's fluctuation. They implemented this model using price fluctuation of up to five previous days and using the same five machine learning models used in the sentiment analysis. The stock margin prediction and stock price prediction models followed. These models utilized a linear regression, where three different linear models were constructed for both the margin and price prediction models. For the margin prediction model, the first regression included the margin percentage on day x as the dependent variable and past margin percentage as the independent variable. Similarly, for the stock price prediction, the current stock price was the dependent variable, and the past stock price was the independent variable. The second regression included past percentage of positive Tweets, while the last model included an interaction term of the two regressors from the first two models. Past variables were lagged up to five days prior [2].

The machine learning model with the highest accuracy for the price fluctuation prediction model was Random Forest with an accuracy of 67.37% and a sentiment classification of Naïve Bayes. The margin price prediction resulted in an R-squared value close to zero for all three regression models, implying that the variation in the data was not explained well by the regressors chosen. The stock price prediction model resulted in a high R-squared value for all three regression models. The regression using stock price from five previous days as the only regressor yielded the highest R-squared value of 0.9989. This implies that stock prices from five-days prior provide a very good estimation for future stock prices. The regression with the interaction term of positive percentage of Tweets and stock prices from five previous days was the second highest R-squared value of 0.9983. Implying that concurrent stock prices and lagged Twitter sentiment are also good estimators for stock price prediction, given the regressors ability to explain the most of variation in the data [2].

This study provided several methods for conducting a sentiment and prediction analysis however, further explanation on why the authors chose these specific machine learning methods should have been included in the paper. Additionally, the sentiment analysis could be employed in a different manner given the accuracy in the study was not high. Lastly, a different machine learning model for a time series analysis could have increased the accuracy of the prediction analysis, such as implementing an LSTM model. Qian et al. [2019] examine how stock price predictions vary utilizing an LSTM neural network model with different stationarity conditions and compare their results from the model with an Auto Regressive Integrated Moving Average (ARIMA) model [4].

The authors discuss in their paper, *Stock Prediction Based on LSTM under Different Stability*, that linear models for stock price prediction cannot accurately utilize historical data compared to non-linear models. The authors explain how LSTM includes a time component within its network without encountering the vanishing gradient problem, affirming its usefulness for time series analysis. They implement a prediction model with three stocks that contain similar trends but differ in stationarity for an accuracy comparison. The stock price data was not described in detail aside from noting that the previous thousand days for each stock price was collected. The authors utilized the ADF statistic in Python to identify the stability of the stock prices. The data was split into a training and testing dataset with a ratio of seven to three, and standardized utilizing Min-Max. The LSTM was conducted using TensorFlow in Python and the Root Mean Squared Error (RMSE) was chosen for a performance measure. The LSTM contained 10 hidden layers and a timestep of 5. The ARIMA model was completed thereafter with the Mean Absolute Error (MAE) as the measure for performance. For both models, ten iterated experiments were implemented and the resulting range of the RMSE for the training dataset was [0.021, 0.037] while the testing data RMSE range was [0.027,0.02]. Demonstrating that LSTM is able to adapt to data with varying stability resulting in minimal accuracy loss. In a comparison model, the ARIMA model's RMSE was larger than the LSTM value. The authors conclude that LSTM's predictive ability is unaffected by data with differing stability and performatively LSTM is better than ARIMA. However, the authors note that LSTM requires a large dataset for accuracy. The authors note that the algorithm of the LSTM should be updated to increase performance [4].

Qian et al. [2019] elucidated the effectives of an LSTM model for stock price prediction in their paper, unlike Cakra et al. [2015] where multiple models were implemented without

reasoning as to why they were beneficial to the study. In light of this, this paper will update the work conducted by Cakra et al. [2015] by applying the prediction method used by Qian et al. [2019]. Additional changes include that the sentiment analysis will be employed with a different algorithm, no specific geographical location will be studied, North American based companies will be studied, and the time period will increase to sixty-five days.

3. DATASET

This research is composed of two different datasets from July 2020 to October 2020. The first dataset includes the stock prices of each of the four big technology companies. This dataset is collected from Yahoo Finance CSV API, where Yahoo Finance provides open-source historical daily data for each company [5]. The data has seven main attributes including the daily open, close, high, low, and adjusted closing price of the stock. The open and close variables represent the value of the stock at the beginning and end of each trading day. The adjusted close adjusts for stock splits, dividends, and rights offerings [6]. High and low represent the highest and lowest values on a given day.

The second dataset utilized in this study is collected from Twitter. This dataset is obtained from Kaggle collected using the Twitter API by user William Jiang. It is licensed under the General Public License version 2 (GPL2), which allows copying, distribution, and modification of the licensed product. The Tweets mention the companies of interest and any of their products through hashtags. The dataset does not limit any particular geographical location and is composed of English based Tweets only. It contains all the individual Tweets that mention the companies of interest, the date and time of the Tweet, the full text of the Tweet, the location, and the username of each user. This study only uses the content of the Tweet and the date it was posted [7].

The Twitter dataset is composed of ten technology companies in total, the four companies in this study are chosen due to their equal majority from the entire dataset as seen in Figure 1 [7]. It should be noted that two companies studied in this paper are subsidiaries of larger companies, YouTube and Twitch are owned by Google and Amazon, respectively. These two companies do not have individual stock prices as their value is included within their holding company stock price. This could impair the accuracy of YouTube and Twitch's prediction models. For example, public Twitter sentiment regarding YouTube could only partially affect Google's total stock value or may leave it unchanged.

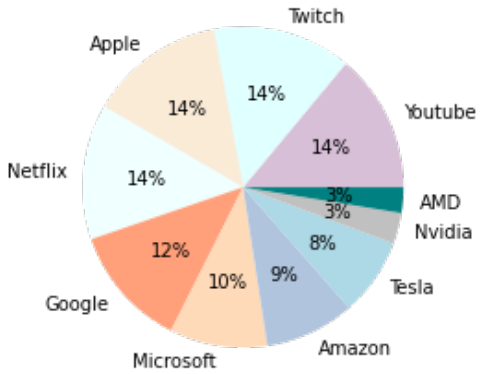


Figure 1: Distribution of Twitter Dataset

4. METHODOLOGY

This study is implemented in the following three stages: Stage One includes data preprocessing and the execution of the sentiment analysis. Once the sentiment value of a given day is obtained, each Tweet is filtered based on the company it is concerning. Thus, four different data frames are created. The data frames are then merged with the appropriate company's stock price dataset. Stage Two conducts a correlation analysis between stock price and sentiment value in order to identify if a relationship exists between these variables. Stage Three employs the three LSTM prediction models.

4.1 Preprocessing and Sentiment Analysis

The python library TextBlob is used to pre-process the raw Twitter data. This involves tokenizing each Tweet, as done in Cakra et al. [2015]. Additionally, each word is stemmed, this is process of removing the ending of a word to obtain to root of a word, such as stemming the word "stopping" to "stop". Stop words are removed as well, these are words that do not provide any sentiment value or meaning, such as the word "the". Symbols such as hashtags '#', mentions '@', and hyperlinks are also removed. Visualization of the cleaned text data is conducted for each company. Figure 2 illustrates Apple's Word Cloud with the most frequently Tweeted words related to Apple.

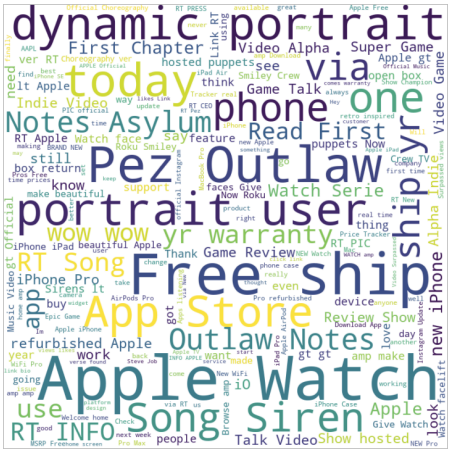


Figure 2: Apple Word Cloud

The text data is then given a value for its subjectivity and polarity level, and is classified as positive, negative, or neutral. The percentage of positive Tweets are calculated using percentage positive equation from Cakra et al. [2015], where

sentiment value is now proxied by this formula. After obtaining the percentage of positive Tweets for each day per company, the values are then merged with the Yahoo Finance stock price dataset [5] to implement the LSTM prediction models. Figure 3 presents how the final data frame for each company appears. Additionally, the first differences of close price are calculated to obtain stock fluctuation for the last prediction model.

month	day	postitive_tweets	negative_tweets	positive%	DayPolarity	Close	Up=1/Down=0
7	13	119	23	83.802817	Postive	3104.000000	0.0
7	14	213	52	80.377358	Postive	3084.000000	0.0
7	15	683	132	83.803681	Postive	3008.870117	0.0
7	16	530	88	85.760518	Postive	2999.899902	0.0
7	17	944	186	83.539823	Postive	2961.969971	1.0
...	...	...	...	...	...	...	...

Figure 3: Final Dataset for Prediction Models

4.2 Correlations Analysis

The objective of this study is to identify whether sentiment from Twitter about a given company is predictive of that company's stock price. Before testing this idea, the relationship between sentiment value and stock price is examined. The Pearson Correlation is used to find the relationship between these two variables. If the correlation value is positive, then sentiment will positively impact stock price. If the correlation is negative, then sentiment will negatively impact stock price. The following formula is the Pearson Correlation, where y is stock price, x is sentiment value, and any variable with a bar represents the mean value.

$$Corr\ (x,y)\ = \frac{\sum(x-\bar{x})(y-\bar{y})}{\sqrt{\sum(x-\bar{x})^2(y-\bar{y})^2}}$$

The co-variance was not utilized in this study as the Pearson Correlation provides both direction of relationship and the strength of the correlations as shown in the equation.

Prior to conducting the correlation, the variables are normalized by subtracting the mean and dividing the standard deviation. The correlation results demonstrate that the correlation coefficient between sentiment and stock price for all companies is very low. This indicates that there is little to no relationship between our variables of interest. A correlation analysis between the four stock price variables and closing price is also implemented for the Multivariate LSTM model. Where open, adjusted close, high, low values are highly correlated with close price. The correlation values will be further discussed in Section 5.

4.3 Three LSTM Prediction/Forecasting Models

The prediction and forecasting models are implemented using an LSTM model. Where LSTM is a unique version of a Recurrent Neural Network (RNN) that works well with sequence data, which is useful for prediction problems [8]. LSTM is chosen over other traditional regression models because of its ability to remember past information. Time series analysis works best given LSTM's ability to observe patters and relationships with a longer memory than other machine learning methods.

LSTM's cell structure is composed of three main gates, forget gate, input gate and output gate [9][10]. The forget gate decides which information the model should remember, and it decides whether to discard information. It is composed of a sigmoid function for its input. Thus, the output of forget gate is either one or zero. If the output is one, then no information is lost or forgotten, and it continued through the cell state. The input gate decides what new information LSTM will save in the cell state. It is composed of both a sigmoid and tanh function. Where, the sigmoid layers decide which data is stored and the tanh layers distribute a weight to each output value that was passed through the sigmoid layer of input function. The output gate decides what to output to the following cell and is composed of a sigmoid function and tanh function. Where the tanh function gives a weight to the output in the range between negative one to one [11]. Figure 4 depicts the gates within an LSTM cell, obtained from Christopher Olah's [2015] GitHub [9].

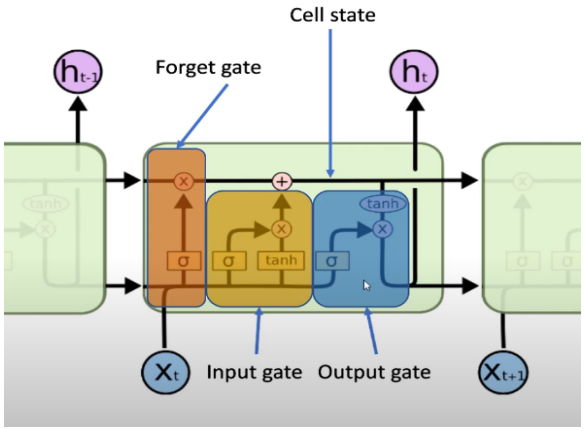


Figure 4: LSTM Cell with Three Gates [9]

To implement the LSTM model in Python, the work conducted by Krish Naik [12] and Sreenivas Bhattiprolu [13] using the TensorFlow and Keras libraries in GitHub is adapted. A sequential API is used to create the LSTM models using Keras. The LSTM model is layered with 50 units. Given that this model used more than one LSTM layer, the return sequence is set to "True" allowing for a three-layer LSTM. The timestep is set to four and a dropout layer is included to reduce overfitting. Lastly "Adam" is used as an optimizer and the Root Mean Squared Error (RMSE) is used to compute the loss in the models. The epoch is set to 100 with a batch size of 64 and the training and validation loss are plotted. From the Keras library,

the function model.predict() is used to evaluate how each model works for our datasets [12] [13]. Each model split the entire dataset into a training and testing dataset, where the training dataset consists of the dates between July 7th to September 9th and the testing dataset is from September 10th to October 9th. The three models created in this study are the described below, where all values are normalized prior to implementation in the LSTM models. Normalization method is MinMaxScaler in Python where the results are returned to their original ranges using the inverse transform function.

Model 1: Sentiment LSTM

This is a single variable model that predicts closing stock price based given sentiment value for each company. Despite the low correlation results, this model is still implemented with the goal of being improved in future work. The input variable is Twitter sentiment, and the output variable is predicted closing stock price.

Model 2: Multivariate LSTM

This model utilizes five features from the Yahoo Finance dataset, it includes open, close, low, high, and adjusted close price. These five variables are used as input and the output is predicted closing price.

Model 3: Price Fluctuation LSTM

This model only utilizes sentiment value as an input and outputs the closing stock fluctuation for each company.

Figure 5 shows the training and validation loss from the Multivariate LSTM for Twitch.

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
lstm (LSTM)                  (None, 4, 50)             11200
_________________________________________________________________
lstm_1 (LSTM)                (None, 4, 50)             20200
_________________________________________________________________
lstm_2 (LSTM)                (None, 50)                20200
_________________________________________________________________
dense (Dense)                (None, 1)                 51
=================================================================
Total params: 51,651
Trainable params: 51,651
Non-trainable params: 0
```

Figure 5: Model Summary for Twitch's Multivariate LSTM Model

5 RESULTS

For all four companies a separate correlation analysis is implemented. The RMSE values are also calculated for each company per LSTM model. Given the limited space in this paper, Twitch will be exemplified in the prediction graphs while the remaining graphs of the other companies will be included in the appendix.

5.1 Correlation Results

The correlation coefficients between sentiment and stock price are very low for all companies, indicating that there is little to no relationship between the variables as shown in Table 1.

YouTube	Twitch	Netflix	Apple
0.18	-0.19	0.025	-0.031

Table 1: Correlation Coefficients between Sentiment and Closing Stock Price

This low correlation could be due to various factors such as the quality of Tweets obtained, or the short time of study. Although the daily correlation between these variables is low, a higher correlation could be obtained with a lagged sentiment value and daily close price. It could take Twitter sentiment time to affect stock prices, this premise will be explored further Section 7.

5.2 Model Evaluation

The training dataset comprised sixty-five percent of the entire dataset, while the remaining thirty-five is the testing dataset. Figure 7 shows the plotted training and testing loss for Twitch in the Multivariate LSTM, while Figure 8 shows Twitch's loss in the Sentiment LSTM.

In the Multivariate LSTM, the testing loss curve is slightly below the training loss curve which implies that the model may

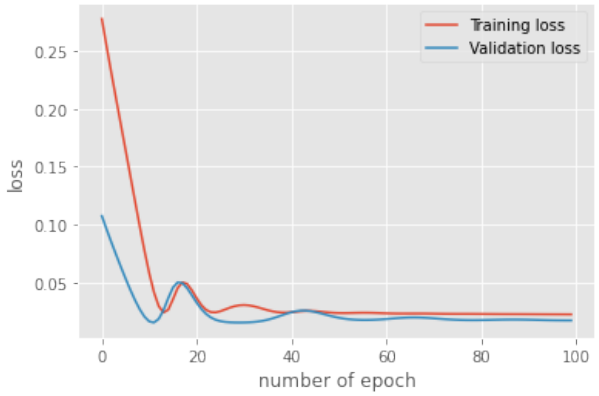


Figure 7: Twitch's Training and Validation loss in Multivariate LSTM

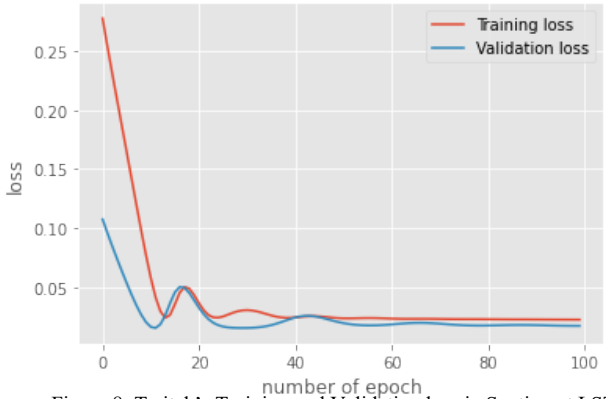


Figure 8: Twitch's Training and Validation loss in Sentiment LSTM

be underfitting the data. A similar argument can be made for the Sentiment LSTM in Figure 8 however, the difference between the two lines is wider indicating that the Sentiment LSTM underfitting problem is worse.

5.2A Sentiment LSTM Model Evaluation

As expected, this model resulted with a very low accuracy and could not forecast future stock prices in correct manner. This is not surprising given the low correlation value found earlier. Figure 9 displays the testing prediction results for Twitch. The blue line is the prediction value, and it does not follow the true stock price movements well. The prediction curve is smooth with little variation compared to the sharp changes in the true value. Additionally, it overpredicts the values of the stock prices and this could be due to the fact that the training dataset was trained when the stock prices were on an upward trend while the testing dataset contains a decreasing stock price trend.

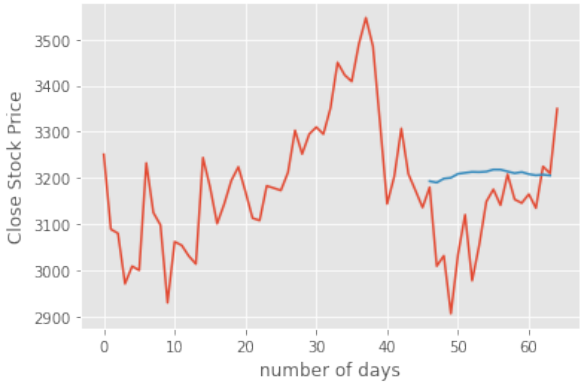


Figure 9: Twitch's Testing Prediction in Sentiment LSTM

5.2B Multivariate LSTM Evaluation

Figure 10 displays the Multivariate LSTM prediction results for Twitch. As shown, the prediction line captures the trend of the stock price movements relatively well and to a better extent than the Sentiment LSTM. The predicted curve contains significantly more variation than the Sentiment prediction and follows the price trend to a greater extent. This was expected given the lower error loss in the Multivariate model. This model was the best performing model out the three executed.

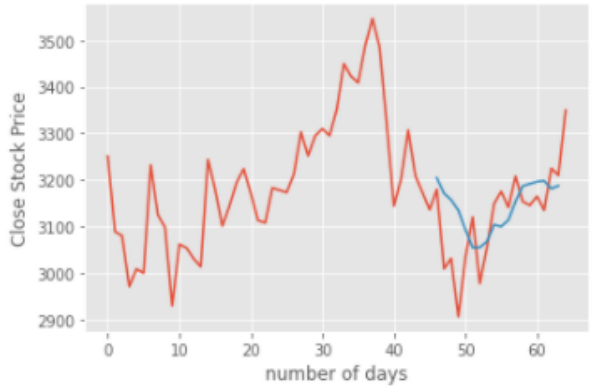


Figure 10: Twitch's Testing Prediction in Multivariate LSTM

5.2C Price Fluctuation LSTM Evaluation

Lastly, Figure 11 displays the Price Fluctuation LSTM for Twitch, where the model performs decently. This model was unable to capture the large fluctuations that occurred in the true dataset. Although the performance of this model was subpar, it was the second-best performing model from the three completed.

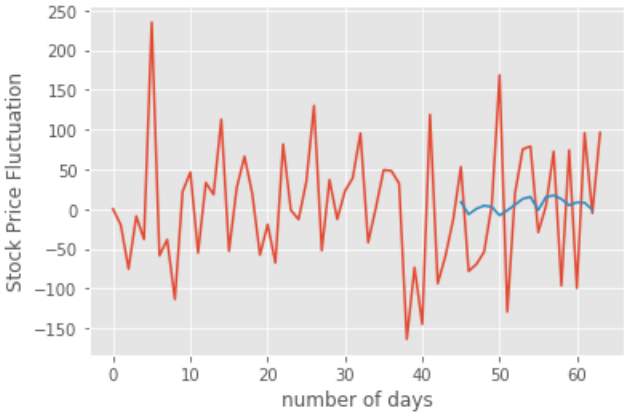


Figure 11: Twitch's Testing Prediction in Price Fluctuation LSTM

In addition to visual representations of the three model's performance, the root mean squared error (RMSE) values were calculated to determine accuracy, following Qian et al. [2019]. The RMSE evaluates the amount of error in a model and whether the value is small or large depends on the units of the dependant variable, y, in the model. In the first two models, the dependent variable is closing stock price. In the last model the dependent variable is a daily change value of closing price. A low RMSE is desired for accuracy. The formula for the RMSE is as follows, where $\hat{y}$ is the predicted value, y is the actual value, and n is the number of observations:

$$RMSE = \sqrt{\sum_{i=1}^{n} \frac{(\hat{y} - y)^2}{n}}$$

The RMSE values are calculated for both training and testing datasets for each company. Table 2 displays the RMSE values of the training and testing datasets for the three LSTM models. Table 3 displays the Price Fluctuation RMSE values of the training and testing dataset for each company.

YouTube	Twitch	Apple	Netflix
Sentiment LSTM			
1551.26	3140.84	118.79	498.15
Multivariate LSTM			
1490.49	3140.85	114.35	492.67

Table 2: Testing RMSE for Sentiment and Multivariate LSTM

Price Fluctuation LSTM				
	YouTube	Twitch	Apple	Netflix
Train	3.54	10.17	0.56	2.03
Test	4.741	8.86	0.47	1.85

Table 3: Price Fluctuation RMSE Values

It is important to note that comparison among the companies RMSE values would be imprecise given that each company has varying ranges for their stock price values. However, a comparison between the Sentiment LSTM and the Multivariate LSTM is possible for each company. YouTube, Apple, and Netflix all have improved RMSE values in their Multivariate LSTM. This is consistent with the more accurate prediction curve in the LSTM graphs. Both of Twitch's RMSE values are extremely close and only differ by 0.01. Given the general decrease in RMSE values in the Multivariate LSTM, these results are concurrent with the arguments above that this model performed the best. Although the stock price prediction is not completely accurate, the model captured the overall trend of prices very well as seen in Figure 10.

In the Price Fluctuation LSTM model, the dependent variable is the daily change in stock price for each company. Figure 3 compares the training and testing RMSE values for each company. In general, most of the company's RMSE values decrease in the testing model, with the exception of Twitch. This low RMSE in the testing dataset implies that this model is subject to underfitting and should be trained with more data. This is consistent with the training and validation loss figures. Given that the Sentiment LSTM model did not fare well with the stock price prediction model, other machine learning methods may yield more accurate results.

6 LIMITATIONS

Given that the Twitter dataset was not collected for the purpose of this study, this study may have been subject to bias given the context of the Tweets. As Tweets are only one dimension of public mood and may not accurately represent the population. The Twitter dataset only collected information about each company from hashtags and did not include other measures that account for the value or worth of these companies. If these additional measures were included in the collection of this dataset then, this could have altered the sentiment value and thus the prediction results in perhaps a more accurate manner. Additionally, the Twitter dataset depends on individuals Tweeting about these companies daily and there is a possibility that one or more companies have no Tweets on a given day or that the distribution of Tweets differs per company. This could explain why some companies fared better than others in the models. Additionally, this study is limited given the time frame of only sixty-five days. A longer time period may have improved accuracy in the LSTM models, as stated by Qian et al [2019]. As this model relies only on stock prices and Twitter sentiment, it should be noted this is a generalization and more research should be completed on the causal relationship between these variables.

7 CONCLUSION

The Sentiment LSTM provided inaccurate prediction results and performed poorly. This could be due to the lack of adequate data required for a high accuracy LSTM model or could be due to the substance of the Twitter data. The Sentiment LSTM should be further investigated with appropriate changes outlined in Section 8. The Multivariate LSTM outperformed the Sentiment LSTM given its ability to properly capture the trend in price movements. Of the three models implemented in this study, the Multivariate LSTM was the most accurate, while the Sentiment LSTM was the least. To address the underfitting issue in both the Multivariate and Sentiment LSTM more data should be collected for an LSTM model. Overall, this study concludes that the EMH theory holds where agents are not subject to emotional responses that affect their market behavior.

8 FURTHER WORK

This paper recommends the following to improve future work:

- Utilize a different machine learning model
- Modification of the sentiment analysis, perhaps filtering for words related to the values of the companies
- Collection of Twitter data that directly discusses or relates to the value or stock of a company
- Lengthen the time frame to a range of six months to a year for improved LSTM accuracy
- Further explore the relationship between stock price and Twitter sentiment through correlation analysis or in a controlled experiment

The last recommendation is further explored in this paper. A correlation analysis with stock price and lagged sentiment value is completed for all companies. Given the low correlation found between current stock price and current Twitter sentiment, a relationship could still exist with lagged sentiment value. The objective is to identify if sentiment value from a

couple of days prior is related to current stock prices. Table 4 presents a comparison between the old correlation values computed and the new correlation. The new correlation utilizes a normalized value of stock price and a normalized value of sentiment from two days prior.

Company	Old Correlation → New Correlation
YouTube	0.18 →-0.059
Netflix	0.025 → 0.12
Apple	-0.031 →0.62
Twitch	-0.19 → 0.021

Table 4: New vs. Old Correlation Values

The strength of the correlation increased for Netflix and Apple in the new correlation. However, Twitch and YouTube's new correlations decreased in magnitude and switched signs. This could be due to the fact that they are subsidiaries and a past sentiment value will not considerably affect their parent company's stock price. Additionally, the new correlations of Twitch and Apple are now positive, this intuitively makes more sense than the old negative correlation. As sentiment is proxied by percentage of positive Tweets, and thus a past positive sentiment of a company should have a positive correlation with that company's stock price. YouTube surprisingly switched signs but this again could be to lack of precision of the relationship between YouTube's sentiment and Google's stock price.

In addition to the new correlation, correlations with sentiment lags were performed with up to five previous days. This is to assess which lag would yield optimal performance in future studies, as seen in Table 5**:**

2	3	4	5
YouTube			
-0.05	-0.04	-0.16	**-0.19**
Netflix			
0.059	0.065	-0.04	**-0.0935**
Apple			
0.62	0.659	0.67	**0.71**
Twitch			
0.021	-0.11	**-0.23**	-0.22

Table 5: Correlation Values Given Lagged Sentiment Value

The bolded values demonstrate the strongest correlations in absolute terms given a lagged sentiment value. As shown, the strongest correlations per sentiment lag differ for each company. YouTube continues to advance further into a negative correlation given past sentiment values, with the largest value at a lag of five days. A similar trend occurs for Twitch and Netflix, where the correlation switches from positive to negative as the number of lags increase. For Twitch, YouTube, and Netflix the strongest correlation is a negative value with lags around four or five days. Apple is the company that shows the greatest promise, given its high correlation values that increase with sentiment lags. Apple is the company with the strongest correlation out of all of the companies in terms of magnitude and is the only company with positive correlation values given lags.

The correlations with stock price and sentiment lags were mostly inconclusive and provided little to no further knowledge the relationship between these variables. Further work with this method should only be conducted with Apple given its high correlation values compared to the other companies.

REFERENCES

[1] Johan Bollen, Huina Mao, and Xiaojun Zeng. 2011. Twitter mood predicts the stock market. Journal of Computational Science, 2(1), (2011), 1–8. https://doi.org/10.1016/j.jocs.2010.12.007

[2] Yahya Eru Cakra, and Bayu Distiawan Trisedya. (2015). Stock price prediction using linear regression based on sentiment analysis. 2015 International Conference on Advanced Computer Science and Information Systems (ICACSIS), (2015), 147–154. https://doi.org/10.1109/ICACSIS.2015.7415179

[3] Shweta Singh, Ayushi Dixit, and Gunjan Joshi. (2020). Is compulsive social media use amid COVID-19 pandemic addictive behavior or coping mechanism? Asian Journal of Psychiatry, (2020), 54, 102290–102290. https://doi.org/10.1016/j.ajp.2020.102290

[4] Fei Qian and Xianfu Chen, "Stock Prediction Based on LSTM under Different Stability," 2019 IEEE 4th International Conference on Cloud Computing and Big Data Analysis (ICCCBDA), Chengdu, China, 2019, pp. 483-486, doi: 10.1109/ICCCBDA.2019.8725709.

[5] Yahoo Finance. 2021. Yahoo Finance. Retrieved from https://ca.finance.yahoo.com/

[6] Ganti, Akhilesh. 2020. Adjusted Closing Price Definition. (December 28). Retrieved March 20, 2021 from https://www.investopedia.com/terms/a/adjusted_closing_price.asp

[7] William Jiang. 2020. Big Tech Companies-Tweet Sentiment. (December 2020). Retrieved January 30, 2021 from: https://www.kaggle.com/wjia26/big-tech-companies-tweet-sentiment/activity

[8] Jason Brownlee. 2017. A Gentle Introduction to Long Short-Term Memory Networks by the Experts. (May 24, 2017). Retrieved March 20, 2020 from https://machinelearningmastery.com/gentle-introduction-long-short-term-memory-networks-experts/

[9] Christopher Olah. 2015. Understanding LSTM Networks. Understanding LSTM Networks-Colah's Blog. (August 2015). Retrieved March 20, 2021 from https://colah.github.io/posts/2015-08-Understanding-LSTMs/.

[10] Sreenivas Bhattiprolu. 2020. 165-An introduction to RNN and LSTM. Video. (October 8, 2020). Retrieved March 20, 2020 from https://www.youtube.com/watch?v=Mdp5pAKNNW4

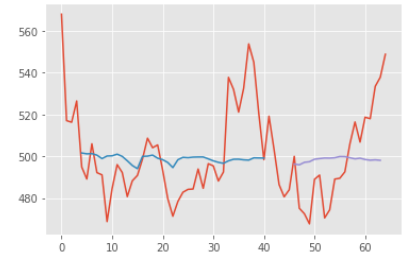
[11] Eugine Kang. 2017. Long Short-Term Memory (LSTM): Concept. (September 2017). Retrieved March 20, 2021 from https://medium.com/@kangeugine/long-short-term-memory-lstm-concept-cb3283934359#:~:text=LSTM%20is%20a%20recurrent%20neural,time%20lags%20of%20unknown%20duration.&text=RNN%20and%20HMM%20rely%20on%20the%20hidden%20state%20before%20emission%20%2F%20sequence.

[12] Krish C Naik. 2020. Stock Market Prediction And Forecasting Using Stacked LSTM. (May 26, 2020). Retrieved March 20, 2020 from https://github.com/krishnaik06/Stock-MArket-Forecasting/blob/master/Untitled.ipynbf

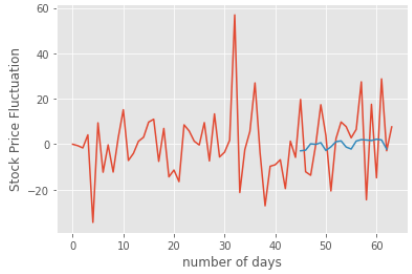
[13] Sreenivas Bhattiprolu. 2020. Python_for_microscopists/166a-Intro_to_time_series_Forecasting_using_LSTM.py. (October 13, 2020). Retrieved March 20, 2020 from https://github.com/bnsreenu/python_for_microscopists/blob/master/166a-Intro_to_time_series_Forecasting_using_LSTM.py

APPENDIX

Netflix Multivariant LSTM



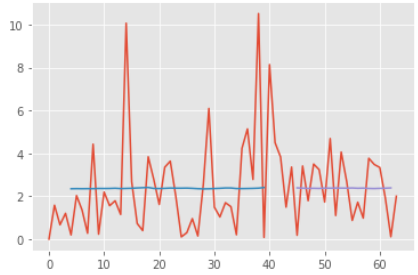
Netflix Sentiment LSTM



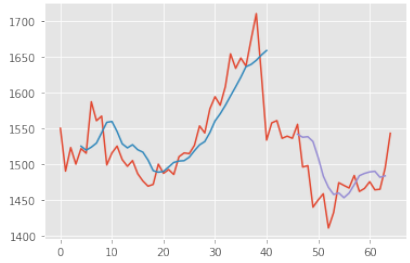
Netflix Price Fluctuation LSTM

Apple Multivariant LSTM

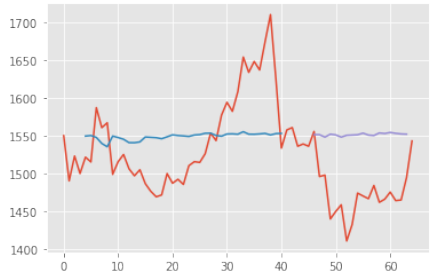
Apple Sentiment LSTM



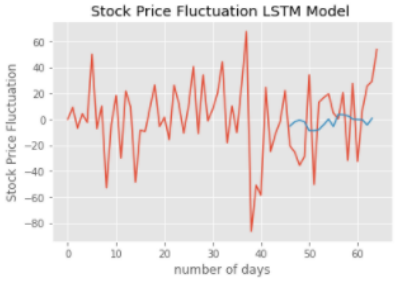
Apple Price Fluctuation LSTM



YouTube Multivariate LSTM



YouTube Sentiment LSTM



YouTube Price Fluctuation LSTM