Stock Price Prediction via Sentiment Analysis and LSTM Network

Yifei Huang, Jonathan Rasmussen, Haoming Yuan
SYD 522 – Group 6
Systems Design Engineering Department
University of Waterloo, Canada
{y652huang, jrasmussen, h34yuan}@uwaterloo.ca

***Abstract*—In recent years, numerous studies have been devoted to examining the correlation between textual sentiments and stock prices; many discovered that social media sentiments provide useful information when predicting stock prices. This project examines the correlation between the sentiment of Reddit posts and opening stock prices. Sentiment analysis was performed on the titles of discussion posts from r/WallStreetBet using the Python libraries VADER and TextBlob. The Reddit sentiment is subsequently used as an input feature for the time series prediction of the open prices for the stocks of interest. The open prices are modeled with a 4-layer stacked LSTM network, where the inputs are historical market data with or without sentiment scores. The results show that the correlation between Reddit sentiment and open prices is negligible. Generally, the inclusion of Reddit sentiment introduces noise and increases error when predicting the actual open prices and their directional movements. Nevertheless, for "swing" stocks with high volatility and a high ratio of retail investors, Reddit sentiment improves the prediction accuracy of open prices significantly.**

***Index Terms*—Sentiment Analysis, LSTM, Stocks**

I. Introduction

Stock markets play a significant role in the global economy as they provide any individual the opportunity to invest in corporations while providing said businesses with investment capital. To achieve success in the stock market, an investor needs to correctly predict the price trends of stocks and make the right trading decision in a timely manner to make a profit. However, finding patterns in stock charts has been a long-standing and challenging research topic as the stock market is influenced by many factors.

One factor that directly affects the stock price is the simple economic concept of demand. The price of a stock increases when its demand is high and drops when the demand is low. With the increasing popularity of social media, more people engage in online discussions of investment strategies. This results in a greater need to measure the zeitgeist's concerning certain stocks through social media posts. Sentiment analysis is the use of natural language processing to study the affective states in texts. By applying sentiment analysis on stock-related social media posts, it is possible to determine whether investors feel positively towards a stock and hence how likely they will invest in it.

In recent years, numerous studies have been devoted to the prediction of stock prices based on textual sentiments; many studies have examined the textual sentiments in Twitter posts and economic news articles and discovered that the combination of sentiment scores and historical market data enables more accurate prediction of stock prices. This project aimed to apply sentiment analysis on Reddit discussion posts and investigated whether a correlation exists between the open price of a stock and its associated post's sentiments. This project also made sentiment-aware time series predictions of open prices and compared the results with the those made without sentiments. The goal of this project is to make viable time series predictions of opening prices of certain stocks using sentiment scores and other public market data to inform the investors of better trading decisions.

II. Background

A number of studies have been done on the prediction of stock charts; while some studies make predictions based on market data only, others also examine the correlation between market data and textual sentiments. Many studies have shown that introducing textual sentiments increases the prediction accuracy; however, results vary depending on the source of textual sentiments and the stocks of interest.

Ghosh et al. conducted a study on forecasting the directional movements of stock prices using long short-term memory network (LSTM) and random forest based on market data only. In their study, Ghosh et al. collected the open and close price of all constituent stocks of the S&P 500 from the period of January 1990 until December 2018. Ghosh et al. predicted the difference between the open and close price on the same day. Ghosh et al. achieved 69.67 % accuracy using LSTM and 65. % accuracy using random forests in predicting the direction of price changes. [1]

Li et al. also proposed the application of LSTM on financial time series forecasting; unlike Gosh et al., they also included sentiment as an input feature. Li et al. collected more than 18 million posts containing the CSI300 index from an online discussion board and classified their sentiments using Naïve Bayes classifier. The sentiment time series and CSI300 opening values were modeled by LSTM. Li et al. achieved an 87.9% accuracy in predicting the direction of next-day price change. [2]

Bollen et al. examined the correlation between Twitter mood and stock prices via OpinionFinder and Google-Profile of Mood. In the study, Bollen et al. analyzed dataset of

9,853,498 tweets posted by approximately 2.7M users. The resulting mood time series was used to predict the closing values of DJIA using Granger causality analysis and Self-Organizing Fuzzy Neural Network. Bollen et al. achieved an 87.6% accuracy in predicting the direction of DJIA's daily price change. [3]

Pagolu et al. also investigated the relationship between Twitter sentiments and the stock market . In the study, Pagolu et al. analysed the sentiment of 2,50,000 tweets relating to Microsoft with random forest, sequential minimal optimization (SMO), and logistic regression. They also compared different correlation analyzers such as support vector machine and logistic regression. Pagolu et al. achieved a 71.82% accuracy in predicting the direction of price change using random forest for sentiment analysis and support vector machine for correlation analysis. [4]

In addition, Neme and Kiss conducted a study to examine the correlation between stock value changes and news headline sentiments. Neme and Kiss collected a dataset of 400 economic news headlines mentioning the stock AMD and compared their sentiments with AMD's daily prices. For sentiment analysis, Neme and Kiss compared TextBlob, NLTK-VADER Lexicon, recurrent neural network (RNN), and bidirectional encoder representations from transformers (BERT). Neme and Kiss discovered that the compound sentiment scores calculated by NLTK-VADER Lexicon and TextBlob related positively to the daily high price of AMD, with correlation scores 0.96 and 0.89. [5]

III. Dataset

This project aims to make time series predictions of stock prices based on Reddit sentiment. The data used for sentiment analysis were posts from r/WallStreetBets, a discussion board focusing on stock market and investment strategies. The dataset is available on Kaggle; it consists of 1,118,863 discussion posts from r/WallStreetBets posted from April 11th, 2012 to February 16th, 2021 [6]. The dataset does not include the body text or comments, but contains the number of comments, number of awards, score (number of upvotes), and the title for each post.

The dataset includes all discussion posts in r/WallStreetBets indiscriminately. This means that while some posts might be a good representation of the discussion board's collective mood, others might be outliers and introduce noise. To obtain accurate sentiment scores and reduce noise from outliers, only posts with high traffic were used for sentiment analysis. For this project, a threshold of score greater than 100, number of comments greater than 50, or the total awards being greater than 10 is applied on the dataset. The resulting data size is 59,544 posts.

A stock ticker search was applied to each post title using the Natural Language Toolkit (NLTK) phrase matcher to determine whether a post is relevant to a stock. The ticker list used in the search was made available by the United States Securities and Exchange Commission [7]. A post is considered relevant to a stock if a ticker match is found in its title. Stocks with tickers that coincide with common words (such as "A" and "AND") were not considered for this project as ticker matches would be found in many irrelevant titles. In addition, some titles contain multiple tickers; however, without body texts and comments, it is difficult to determine the degree of relevance of the posts to each stock mentioned in the title. Thus, for this project, a post is considered relevant to a stock if the ticker appears in its title, regardless of the presence of other tickers.

The sentiment time series produced from the discussion posts are averaged by day and the corresponding market data for each date is obtained from the Yahoo Finance API which is then used to construct a time series model [8]. The model aims to predict the next-day open price value of a stock based on the remaining market data and sentiment scores. However, market data is not available on weekends and holidays, and there are days where no sentiment scores are available. To fill in the null dates, two methods were tested. The first method was to drop the days with null market data and replace null sentiments with neutral scores (0). The second method was to use imputation, where each null value was replaced by the average between the last valid chronological value and the next valid value [9]. It is discovered that the two methods yield similar results; therefore, the first method was used for simplicity.

In addition, with the different scale of values between the market data and the sentiment scores, a min-max scaler was applied to the data to scale all values to be between 0 and 1 to ensure the features are of equal significance.

IV. Proposed Scheme and Algorithms

The proposed scheme for this project can be divided into two parts: 1) sentiment analysis on the Reddit post titles, and 2) time series prediction of stock prices using reddit sentiment and market data. The algorithms were implemented in Python and the VADER and TextBlob libraries were used for sentiment analysis. The correlations between the sentiment scores and market data were examined, and a long short-term memory network (LSTM) was implemented using the Keras library for time series prediction.

As mentioned in the previous section, to ensure a good representation of the collective mood and to reduce noise from outlying posts, only high traffic posts were kept. As there are hundreds of different stocks mentioned in the posts, only the five most mentioned stocks, SPY, GME, TSLA, AMD, and AMC, were used for subsequent analysis.

A. VADER

The Valence Aware Dictionary for Sentiment Reasoning (VADER) is a lexicon and rule-based sentiment analysis model [10]. VADER views input text in the form of "bag of words" and performs sentiment analysis based on a set of manually crafted standards, including a dictionary containing common adjectives, emoticons, and slangs with human assigned sentiment scores. In addition, VADER recognizes the effect of punctuation and capitalization. These features make VADER

especially powerful in analyzing social media texts and well fitted for the r/WallStreetBets dataset. VADER can calculate both the polarity (positive, negative, or neutral) and intensity (compound score between -1 and 1) of textual sentiments. For this project, the intensity is used to create a continuous time series for regression analysis.

B. TextBlob

TextBlob is a Python library for processing textual data built on the NLTK and pattern library [11], [12]; it also uses the lexicon approach and is rule-based. TextBlob uses the word bank with assigned sentiment values from NLTK to determine textual sentiment. However, unlike VADER, TextBlob is not tuned for a specific type of text. Since it is one of the most popular and practical tools for sentiment analysis in Python, TextBlob was also used for this project. Similar to VADER, the polarity score ranging between -1 and 1 produced by TextBlob is used to create sentiment time series for regression analysis.

C. Correlation Analysis

The daily high, low, open, and close price as well as trading volume of the most mentioned stocks are retrieved from Yahoo Finance. The Pearson correlation coefficients between the sentiment scores and each market data are calculated using the below formula [11].

$$r = \frac{\Sigma(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{(x_i - \bar{x})^2 (y_i - \bar{y})^2}} \tag{1}$$

Fig. 1
Pearson correlation coefficient

D. Long Short-Term Memory Network (LSTM)

The long short-term memory network (LSTM) is a type of recurrent neural network (RNN) that has been proven powerful in time series prediction [13]. It is ideal for time series like stock prices where the current value may be informed by previous values. Unlike basic RNNs, LSTM have more complicated repeat modules and hence more controllability over hidden states; it is more viable in handling long-term dependencies.

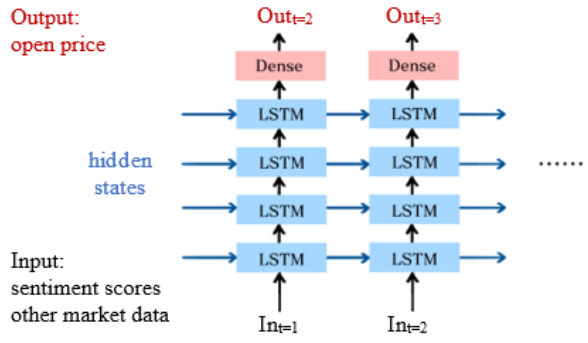


Fig. 2
LSTM Structure

This project models the sentiment scores and market data using a 4-layer stacked LSTM model shown in Figure 2. The input features are the daily sentiment score, close price, high price, low price, and trading volume of a stock; the output is the stock's next-day open price. Since the input features may affect the output with different time lags, stacked architecture was used to allow the operation of hidden states at different timescales [13]. The optimal number of epochs and batch size were determined through randomized search. The number of epochs was chosen from the range of 0 and 1000 and the batch size was selected from the list of 16, 32, 64, 128, and 256.

V. Results & Analysis

Table I shows the Pearson correlation coefficients between the sentiment score and the open price for the five chosen stocks. The results show that the relationship between open price and sentiment score is negligible for all five stocks. One possible reason is that there are many zero-filled sentiment scores on dates where no sentiments are found, which may introduce noise to the dataset. In addition, disputes within each post are not reflected in the post titles and can result in further noise. Another possible reason is that the users of r/WallStreetBets do not provide enough insight to influence the stock market and cannot represent the entire investor body's sentiment.

TABLE I
Pearson correlation between sentiment score and open price

Stock	VADER	TextBlob
SPY	0.03	0.01
GME	0.01	0.01
TSLA	0.05	0.02
AMD	0.02	-0.03
AMC	-0.13	-0.06

To analyze the effects of the sentiment scores on the stock market, regression analysis was conducted using the LSTM model. The input features consist of either the VADER sentiment scores, TextBlob sentiment scores, or no sentiment scores along with the market data excluding the open price. Table II summarizes the root mean squared error loss of the predicted open prices between 2019-01-01 to 2021-02-16 with the presence of different sentiment scores. The losses are in the unit of US dollars and represent the absolute difference between predicted and actual open price. Overall, introducing sentiment scores improved the performance of the LSTM network; however, results vary between stocks. As shown in Table II, the model performs well on SPY, AMD, and AMC, where the largest loss between the three stocks was only $4.00. It was also observed that the presence of sentiment scores in the input for the three stocks resulted in higher losses as more noise was introduced to the model. On the other hand, the model performs more poorly on GME and TSLA with losses ranging from $16.18 to $41.76. However, the addition of sentiment scores improves the prediction of GME and TSLA significantly.

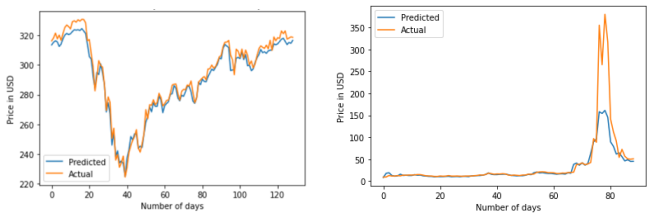
TABLE II
Loss (RMSE) with VADER, TextBlob, and no sentiment

Stock	VADER	TextBlob	No Sentiment
SPY	4.00	2.87	3.14
GME	27.91	27.15	41.76
TSLA	17.42	16.18	29.33
AMD	1.69	2.86	1.45
AMC	0.59	0.45	0.41
Total loss	51.62	49.52	76.08

TABLE III
Directional accuracy with VADER, TextBlob, and no sentiment

Stock	VADER	TextBlob	No Sentiment
SPY	0.78	0.76	0.80
GME	0.63	0.59	0.58
TSLA	0.72	0.70	0.72
AMD	0.78	0.76	0.75
AMC	0.82	0.80	0.76
Avg. acc	0.74	0.72	0.72

The results of SPY, AMD, and AMC are consistent with the previous result that the correlation between sentiment scores and open prices is negligible. TSLA, on the other hand, is a popular stock among retail investors, which may explain the reason why reddit sentiments are more helpful in its case. However, TSLA is also a classic swing stock with high volatility, which makes it noisy and results in larger loss. GME is also popular among retail investors and has experienced a drastic price spike. Figure 3a and Figure 3b show how the model performs on a typical stock (SPY) and an outlier (GME). As Figure 3a shows, the model predicts SPY consistently with small error. On Figure 3b, the model also predicts GME consistently before the sudden spike; at the spike, the model picks up the trend but fails to predict the actual price.



(a) SPY (b) GME

Fig. 3
Predicted stock price with VADER

To determine the model's accuracy in predicting the directional movements of open prices, the regression problem is transformed into a classification problem by categorizing the difference between current and previous stock prices as positive or negative. The classification conversion is applied to the actual and predicted stock prices and the values are compared. Table III summarizes the directional accuracies for each stock with different sentiment inputs. As Table III shows, the addition of sentiment scores only marginally improves (if not corrupts) the model's performance. In addition, the model still performs better on typical stocks than on swing stocks. GME has the worst accuracies again due to its sudden price change.

VI. Conclusions

Based on the results, it is possible to conclude that the inclusion of sentiment from Reddit posts generally introduces noise and increases error when predicting the open price values and directional movements of typical stocks using a LSTM network. The model is not sensitive to sudden, drastic change of prices and cannot make viable predictions for outliers such as GME. However, for stocks with high volatility, sentiment scores bring in useful information and improves the prediction on open values significantly. Thus, the model is noteworthy for swing stocks that are particularly influenced by retail investors.

For further development, a larger dataset with longer timeframe should be used, and more training should be done on high-volatility stocks. In addition, dataset that contains body texts and comments should be considered to enable more accurate sentiment analysis.

References

[1] Pushpendu Ghosh, Ariel Neufeld, and Jajati Keshari Sahoo. Forecasting directional movements of stock prices for intraday trading using lstm and random forests, Apr 2020.
[2] Jiahong Li, Hui Bu, and Junjie Wu. Sentiment-aware stock market prediction: A deep learning method. *2017 International Conference on Service Systems and Service Management*, 2017.
[3] Johan Bollen and Huina Mao. Twitter mood as a stock market predictor. *Journal of Computational Science*, 44(10):91–94, 2011.
[4] Venkata Sasank Pagolu, Kamal Nayan Reddy, Ganapati Panda, and Babita Majhi. Sentiment analysis of twitter data for predicting stock market movements. *2016 International Conference on Signal Processing, Communication, Power and Embedded System (SCOPES)*, 2016.
[5] László Nemes and Attila Kiss. Prediction of stock values changes using sentiment analysis of stock news headlines. *Journal of Information and Telecommunication*, page 1–20, 2021.
[6] Raphael Fontes. Reddit - r/wallstreetbets, Feb 2021.
[7] Company tickers.
[8] R Arossi. Yfinance 0.1.59.
[9] Anshul Mittal and Arpit Goel. Stock prediction using twitter sentiment analysis.
[10] E.E. Hutto, C.J. & Gilbert. Vader: A parsimonious rule-based model for sentiment analysis of social media text. *Eighth International Conference on Weblogs and Social Media (ICWSM-14)*, Jun 2014.
[11] Textblob: Simplified text processing.
[12] Tom De Smedt and Walter Daelemans, Jun 2012.
[13] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997.