Stock price prediction using sentiment Based LSTM: S&P500 vs Reddit posts

Oladapo Richard-Ojo and Hayden Wimmer [0000-0002-2811-4531]

Georgia Southern University, Statesboro GA, USA
or01131@georgiasouthern.edu, hwimmer@georgiasouthern.edu*

***Abstract*- The stock market is as volatile as it is unpredictable, the unstable nature of the stock market results in fluctuations in stock prices and invariably, the market performance of stocks. Understanding the underlying factors that contribute to the volatility of the stock market, which has its consequences on stock prices, has become important to researchers and investors alike. Some of the methods that researchers have used in the past as a gauge for understanding market performance include analyzing economic conditions, understanding company performance, following geopolitical events and market trends. To contribute to the vast research field of stock price predictions and the challenge of understanding stock price fluctuations, this study will aim to find a relationship between human sentiments on the social media platform, Reddit, and the S&P 500 stock index. In this study, we will analyze posts from five subreddits that typically discuss the stock market and stock price fluctuations. This will form the first part of our dataset. Historical stock prices for the S&P 500 index will be obtained from Yahoo Finance. This will form our final dataset. Using VADER (Valence aware dictionary and sentiment reasoner), we will extract the sentiments within the five subreddits and categorize them into positive and negative sentiments. The historical stock prices from Yahoo finance will be matched with the aggregate sentiments for each day and this data passed through the LSTM model for training. Our findings provide strong evidence of social media's impact on stock price predictions.**

Keywords—LSTM, reddit, sentiment analysis, stock market.
Acknowledgement – *This work was support in part by the National Science Foundation (USA) under Grant no. 2321939*

1. INTRODUCTION

Stock market prediction has long been a challenging and lucrative area of research and investment. In recent years, the integration of sentiment analysis and machine learning techniques has emerged as a powerful approach to enhance the accuracy and precision of stock price forecasting. Sentiment analysis is a form of Natural Language Processing (NLP) that aims to analyze user opinion, emotions or feelings towards a product, service, idea, or event. In recent times, Sentiment analysis has been used in combination with Machine learning (ML) not only to analyze user opinion but additionally predict user behavior in the future. Social platforms such as Facebook, X and Reddit are some of the most common platforms where user/public opinion is shared. With over 4 billion active users combined, they have proven to be good data sources for sentiment analysis and ML because of the diverse range of views and opinions being shared on them. Machine learning using neural network models helps create models that can predict stock prices based on the sentiments that users hold about a particular stock. For time series predictions, such as stock price predictions, LSTM (Long short-term model) has proven to be very effective as it is able to process sequences of data, are highly adept at capturing temporal dependencies in stock price data. They excel in recognizing patterns and trends in time series, making them a popular choice for short-term price prediction.

This study seeks to explore the application of sentiment analysis, natural language processing, and machine learning algorithms to predict stock market movements of the S&P (Standard and Poor's 500) index (which is made up of the best performing stocks of the 500 largest companies in the US) by analyzing posts from five relevant and highly followed subreddits (*r/investing, r/wallstreetbets, r/ethtrader, r/stocks and r/pennystocks*) on the social platform, Reddit. The findings in this study will aim to highlight the relationship between the subreddit posts and the US economy which the S&P 500 Index represents. Also, by leveraging sentiment data and advanced modeling techniques, investors and financial institutions can make more informed trading decisions, reduce uncertainty, and achieve better risk-adjusted returns. This research also adds to the wealth of research that has been done around stock market predictions, albeit this study will further advance the research field by analyzing the sentiments of several subreddits and relating these sentiments to the performance of the S&P 500 Index in the stock market. Previous research in the field have also

attributed industry performance, company news, investor confidence, gossip and public events as some of the factors affecting change in stock prices.[1, 2]

Improving the research field heightens our motivation in carrying out this study. To this end, we have highlighted two research questions to be answered at the end of this research endeavor.

RQ 1: Is there a correlation between social media sentiments and stock price movements?

RQ 2: Does combining sentiment data from social media sources with historical stock data improve the performance of neural networks for data training?

In this paper, we will access and collect user data from Reddit using PRAW (Python Reddit API Wrapper), the data will be collected from 5 different related subreddits daily for 30 days and the S&P 500 historical stock price data will be collected from Yahoo finance for the same period. The sentiment and historical stock data will be analyzed in different instances. The sentiment data analysis will involve preprocessing and cleanup of the collected Reddit data using a Python function for stopword removal. The stock data is also pre-processed by extracting relevant features to be used for training our dataset. The processed data is then merged and trained using the LSTM (Long short-term memory) model, which is a form of recurrent neural network (RNN) that has shown very accurate results for time-series data. The final results show that combining sentiment analysis with stock price prediction returns more accurate results than when historical stock prices are used by themselves for stock predictions. Our results also show that there is a fairly positive correlation between social media sentiments and stock price movements, however, based on certain limitations such as insufficient training data, and other such factors like news and market trends, we cannot comprehensively conclude that stock prices are affected by social media sentiments.

2. LITERATURE REVIEW

The underwhelming performance realized when carrying out abstractive summarization on certain datasets as compared to extractive summarization models, is worrisome. Kim, et al. [3] posit that the subpar results gotten from carrying out abstractive summarization is because of a bias that exists in the datasets which favor extractive methods. This bias exists when the data is collected from formal documents such as news articles. They go ahead to use Reddit datasets for this research due to their casual and conversational nature and introduce a novel method called the Multi-level Memory Networks (MMN) for storing information retrieved from data sources. The Reddit dataset is preprocessed, the text is embedded and passed into the Multi-level Memory network algorithm for training. Two existing datasets in addition to Reddit TIFU are assessed: the abstractive subset of Newsroom and XSum, for comparison of the novel model based on their use case in abstractive summarization. Three abstractive summarization methods, one fundamental seq2seq model, two heuristic extractive methods, and different iterations of our model are used for comparison. The MMN model performs better than both convolutional-based and RNN-based abstractive approaches. The results demonstrate the efficacy of our multi-level memory for abstract datasets, even on formal documents with enormous vocabulary sizes. The state-of-the-art abstractive methods tested were also outperformed by their MMN model, showing that their model is efficient in carrying out abstractive summarizations [3].

Political discussions are a common thing on social media platforms. Morini, et al. [4] attempt to measure how polarized Reddit was midway through Donald Trumps' presidency. The aim is to determine and quantify the degree of a user's orientation with pro-Trump beliefs and vice versa. To do this, they made use of word embeddings, neural network parameters and the LSTM technique. They use data from Reddit spanning between 2017 and 2019. The data is gotten by using the Push shift API. Three subreddits (one pro-trump and two anti-trump) are used to get a balance in polarity of subject. The data is then processed to get rid of "noise" in the text. This is essential for feeding the data into the LSTM model for training. They also tune hyper-parameters for the LSTM model using learned and Glove embeddings. The model with Glove pre-trained embeddings and 128 LSTM units achieves the best accuracy in training and validation sets (84,6% and 83%, respectively). To determine the model's generalizability, evaluation of its performances across three other datasets (Gun control, political discussion, and minority discrimination) was carried out. Despite size variations, the model consistently achieves an accuracy of above 72% [4].

Olabanjo, et al. [5] use a Natural Language Processing (NLP) framework to get insights into Nigeria's 2023 presidential election based on public opinion using dataset gotten from Twitter. Tweepy, an open-source python library was used to scrap data from Twitter. 2 million tweets with 18 features were collected from Twitter containing public and personal tweets of the three top contestants. Analysis was performed on the pre-processed dataset using three machine learning models namely: Long Short-Term Memory (LSTM) Recurrent Neural Network (RNN),

Bidirectional Encoder Representations from Transformers (BERT) and Linear Support Vector Classifier (LSVC) models. The sentiment models gave an accuracy of 88%, 94%, 73% for LSTM, BERT, and LSVC respectively [5].

YouTube is the most active and popular social media platform in Indonesia. Wisnubroto, et al. [6] aimed to mine the opinion of YT users about the 2024 Indonesian Presidential elections. The YT data was collected by crawling and coded using python language. The data was then preprocessed to remove stop words making it easier for the ML algorithm to process. Sentiment analysis was then carried out to analyze the data and determine user opinions. Their findings suggest that there are more negative opinions than positive as they relate to the Presidential candidate [6].

Sarkar, et al. [7] set out to combine sentiment of social media data (Twitter and Reddit) with old stock data and study its effect on closing prices over a period. Apple and Tesla stocks were used as a case study. Twitter data was collected using snscrape and Reddit data was collected using pushshift API. Data preprocessing was done to clean up the text and Vader was used alongside Finbert for the sentiment analysis and classification. The study showed that social media data from both Twitter and Reddit have a deep influence on close price movements. It was also determined that Sentiment of tweets by executives have a deeper influence on the prediction of close price, due to their impact on society and the faith the masses have in them [7]. Trawinski, et al. [8] propose a novel method for predicting stock prices based on the sentiment analysis of tweets from the US Congress and the general public. They use a long short-term memory (LSTM) network to capture the temporal dependencies of the tweets and their impact on the stock market. They compare their approach with several baselines and show that it outperforms them in terms of accuracy and profitability. The paper also analyzes the differences between the sentiments of the Congress and the public, and how they affect the stock prices differently [8].

Guo [9] investigates the impact of news sentiment analysis on stock price prediction using a long short-term memory (LSTM) neural network. The author collects daily news articles and stock prices of 10 companies from the S&P 500 index and applies a sentiment analysis tool to extract the polarity and subjectivity of the news. Then, they train an LSTM model with both news sentiment features and historical stock prices as inputs and compare its performance with a baseline LSTM model that only uses historical stock prices. The results show that the news sentiment analysis can improve the accuracy and stability of the stock price prediction, and that the polarity feature has a stronger effect than the subjectivity feature [9]. Sarkar, et al. [10] propose a novel approach for stock market prediction using long short-term memory (LSTM) and sentiment analysis. They use LSTM to model the temporal dependencies of stock prices and sentiment analysis to capture the effects of news articles on investors' sentiments. The proposed method, LSTMSA, combines both features to generate more accurate and robust predictions. The paper evaluates LSTMSA on two datasets, NSE and BSE, and compares it with several baselines. The results show that LSTMSA outperforms the baselines in terms of mean absolute error, root mean square error, and directional accuracy [10].

Lim and Yeo [11] present a machine learning approach to predict the movement of the New York Stock Exchange Composite (NYA) based on technical features and content features from Twitter accounts. The authors use probabilistic sentiment analysis of Twitter news and apply it to a simple recurrent neural network with gated recurrent units. They show that their method improves the prediction performance significantly compared to using only technical features or only content features [11].

Weng, et al. [12] propose a novel method for stock price prediction based on Long Short-Term Memory (LSTM) and Bidirectional Encoder Representations from Transformers (BERT). They first use LSTM to capture the temporal dependencies of historical stock prices, and then use BERT to extract the semantic features of financial news. The two types of features are concatenated and fed into a fully connected layer to generate the final prediction. The paper evaluates the proposed method on two real-world datasets and compares it with several baseline methods. The experimental results show that the proposed method achieves superior performance in terms of accuracy, precision, recall and F1-score [12].

Karlemstrand and Leckström [13] propose a neural network model that incorporates historical stock values, technical indicators and Twitter attributes such as sentiment score, favorites, followers, retweets and verified status. They claim that adding more Twitter attributes improves the prediction accuracy by 3% and that using technical indicators reduces the mean squared error by 11%. The limitations and challenges of using Twitter data for stock price prediction are discussed, such as the difference in time zones, the popularity of the stock and the noise in the data [13].

Sen [14] presents a novel approach to forecast the future movement of stock prices using a combination of machine learning, deep learning, and sentiment analysis techniques. They propose several predictive models based on convolutional neural networks (CNNs) and long short-term memory networks (LSTMs) that use historical data of the NIFTY 50 index listed in the National Stock Exchange of India as the input. The paper also incorporates a sentiment analysis module that analyzes public opinion on Twitter on the NIFTY 50 stocks and uses it as an additional input to the predictive models. They evaluate the performance of the proposed models using various metrics such as mean absolute error, root mean squared error, mean absolute percentage error, and directional accuracy. They posit that the proposed models can achieve high accuracy and reliability in predicting the future values of stock prices and outperform the existing methods in the literature [14].

Bollen, et al. [15] present an innovative approach to predict the movements of the stock market based on the collective mood of Twitter users. The authors use a text analysis tool called OpinionFinder to measure the positive and negative sentiment of tweets, and a machine learning algorithm called Google-Profile of Mood States (GPOMS) to classify tweets into six mood dimensions: calm, alert, sure, vital, kind, and happy. They then correlate these mood indicators with the Dow Jones Industrial Average (DJIA) values and find that some of them can anticipate the changes in the market up to four days in advance [15]. Sidi [16] explores the idea of using related stocks as additional features for forecasting algorithms. The paper claims that most of the existing algorithms train only on data collected on a particular stock, while a professional trader would also consider the performance of similar stocks in the same industry or market. The paper proposes to use five different similarity functions to measure the relatedness of stocks based on their time series data, and to use co-integration similarity as the best one for improving the prediction model. The paper evaluates the models on seven S&P stocks from various industries over a five-year period and compares them with a state-of-the-art model that does not use similar stocks. The paper reports that the prediction model that uses similar stocks has significantly better results in terms of accuracy and profit [16].

Mohanty, et al. [17] present a novel approach to forecast the future prices of stocks using long short-term memory (LSTM) networks. They claim that the proposed method can outperform existing techniques based on linear regression, support vector machines and artificial neural networks. The paper also provides a detailed analysis of the performance of the Stockbot model on various datasets, such as the S&P 500, NASDAQ and NIFTY 50 [17]. Kalyani, et al. [18] propose a method to predict stock trends using news sentiment analysis. The authors assume that news articles have an impact on the stock market and try to classify them as positive or negative. They use three different classification models: Naive Bayes, Random Forest, and Support Vector Machine. They evaluate their models on a dataset of news articles and stock prices of four companies. They report that their models achieve more than 80% accuracy in predicting the stock trends and outperform random labeling by 30% [18].

3. METHODS

This section will discuss the methods used for this implementation. The choice of methods used in this paper were considered for their usefulness in analyzing stock price data and sentiment analysis of Reddit posts.

3.1 Time Series Data

Stock price data is considered as time series data, this is mainly due to the fact that the value of stocks is quoted on a daily, weekly, monthly, and yearly basis. In this paper, we utilized historical stock data for the S&P 500 index which gauges the performance of the 500 largest companies listed on the US stock exchange. This stock formed the basis of our investigation and experiment as we sought to find the relationship between it and social media, Reddit sentiments.

Sequential Data. We analyzed stock prices of the S&P 500 index for this experiment; the stock data as earlier stated is a time series data which makes it particularly useful for our LSTM model. The concept behind the LSTM model involves passing data/information into the LSTM network as input and producing an output which is used as input in another sequence. It does this repeatedly in a loop or in sequence to accurately solve prediction or forecasting problems.

Previous work shows that to improve stock prediction models, related S&P 500 stocks from different sectors are analyzed rather than a single one, using Random Forest and Gradient boosting trees algorithms, the results reveal that

training done on similar stocks had significantly better results with 0.55 mean accuracy, and 19.782 profit as against the state-of-the-art model with an accuracy of 0.52 and profit of 6.6 [16].

3.2 Development environment and programming language

We developed the Machine Learning model for this project using the Jupyter notebook integrated development environment (IDE). Using this IDE, the data was preprocessed and cleaned using Python scripts and libraries such as pandas, NumPy, and NLTK. For collection of posts from the relevant subreddits, we used the Python Reddit API Wrapper (PRAW) library which provides authentication and access to the Reddit platform. The historical stock data (S&P 500) is collected using the yfinance API, this aided easy retrieval of the data from Yahoo Finance.

3.3 Data Collection

Stock data collection. The historical daily stock data in this paper are obtained from Yahoo Finance using the yfinance library on Python. As shown in fig 1.0, the features of this data include open, high, low, close, adj close and volume.

Reddit data collection. Our overall aim is to find a correlation between Reddit sentiments and stock price gains or losses (stock price movements). We initially tried to collect historical Reddit data using the API PushShift, however, Reddit had restricted access to this API to only moderators on their platform. As a work around, we opted to collect daily subreddit posts from Reddit using the PRAW API. This API enabled us access to the required posts except that it only allowed real time data acquisition. To obtain access via the API, a *client_id*, *client_secret* and *redirect URL* address were required to be parsed to the API. The data (subreddit posts) was collected daily over a 30-day period between October 2023 and November 2023.

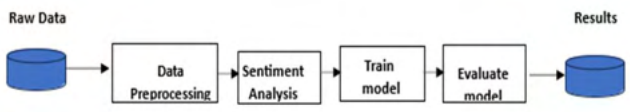


Fig. 1. Implementation flow for Stock price prediction using Reddit sentiments.

3.4 Data Pre-processing

This process involves cleaning and noise removal from the collected subreddit post data. To extract patterns for training the data, neural networks require the data to be free of certain encumbrances like stopwords, spaces, hyperlinks, flags, symbols, and emoticons. To achieve this, we first convert the text into lower case to overcome same words with different capitalization. This could result in inaccurate sentiment analysis of the text if this is not properly done. Using the remove emoji and cleantext functions in Python, the text would be in a clean state for sentiment analysis to be carried out.

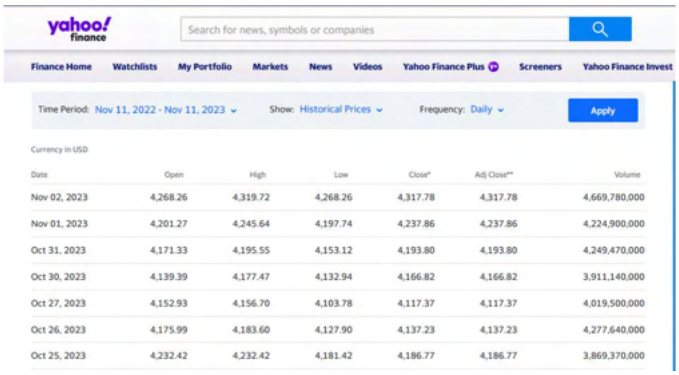


Fig. 2. Historical stock prices from Yahoo Finance

3.5 Sentiment Analysis

This process involves analyzing the text corpus and splitting it into positive, neutral, and negative sentiments. The aim is to get the feeling or opinion of the post that is being analyzed. Being able to extract the sentiments of our subreddit

posts will allow us to identify the relationship between the posts and the stock price movements of the S&P 500 index. VADER (Valence Aware Dictionary and Sentiment Reasoner) a natural language processing (NLP) technique was used to carry out sentiment analysis in this paper. VADER was selected based on its rule based and its effectiveness in matching sentiment scores to features of words and sentences. The NLTK library in Python is a toolkit that processes human language data or text for natural language processing. It is used in this paper for tokenization, classification and tagging of the subreddit posts that were scrapped from Reddit. For the purpose of text analysis, using the NLTK library, we retrieved positive, neutral, negative, and compound scores for all the processed text, however, we only utilize the compound scores (which is a score between -1 and +1) in this paper.

3.6 Polarity Score of Reddit Posts

As earlier stated, the compound score is the output from using the NLTK library in Python for sentiment analysis. The compound score describes the polarity of the sentiments of each post, a score of -1 connotes a negative sentiment while a score of +1 connotes a positive sentiment. As shown in table 1.0, the subreddit posts are analyzed using the NLTK library and polarity scores are calculated for each post.

	neg	neu	pos	compound	Date
does hims have chance against amzn	0.0	0.714	0.286	0.25	10/10/2023 23:17
bank of england warns u s tech stock valuations may be out of whack	0.097	0.903	0.0	-0.1027	10/10/2023 21:48
ast spacemobile nasdaq asts	0.0	1.0	0.0	0.0	10/10/2023 21:12
grandpa s and grandma s stocks please help	0.0	0.545	0.455	0.6124	10/10/2023 20:11
why would you not buy rtx right now	0.0	1.0	0.0	0.0	10/10/2023 19:55

Table. 1. Sentiment classification of subreddit posts showing polarity scores

3.7 Neural Network Model (LSTM)

In this paper, we made use of the LSTM (Long Short-Term Memory) model to train our dataset. The LSTM is a form of RNN (Recurrent Neural Networks) that is widely used for time series forecasting such as stock price predictions.

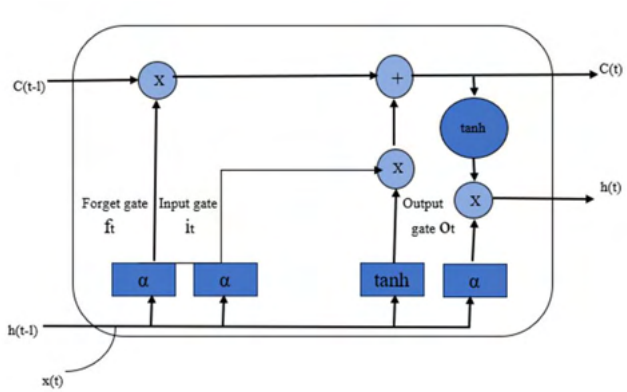


Fig. 3. LSTM Structure

The decision to use LSTM as against a form of CNN is predicated on their common use for analyzing sequential data such as text, or in this case subreddit posts scrapped from Reddit. However, previous studies have been done using other machine learning models in comparison with LSTM. Using LSTM for cryptocurrency price prediction, [19] focuses on three different coins: Bitcoin, Ethereum, and Litecoin. They compare LSTM and Gated Recurrent Unit (GRU) algorithms. Their results show that while GRU is advantageous for downward stabilization trends in BTC and ETH, LSTM is suitable for upward stabilization trends.

Before we feed our data into our LSTM model, there are a few data manipulation techniques that were required. Firstly, we merged our historical stock data with the compound scores from the sentiment analysis done on the subreddit posts. As can be seen in fig 9, the values for compound and stock (open, close, high, low) are in different scales. Using the *MinMaxScaler* function in Python, we scaled the data to make it uniform and for the model to easily understand and train the data. In this paper, we used only the close prices for our study alongside the compound scores which represent the sentiment of the subreddit posts on a given day.

$$x(scaled) = x - \min(x) \over \max(x) - \min(x)$$

	Date	Open	High	Low	Close	Adj Close	Volume	compound
0	2023-10-11	4367	4379	4345	4377	4377	3601660000	0.144645
1	2023-10-12	4381	4386	4325	4350	4350	3713140000	0.025443
2	2023-10-13	4360	4377	4312	4328	4328	0	0.031191
3	2023-10-16	4342	4383	4342	4374	4374	3409960000	0.130307
4	2023-10-17	4345	4394	4338	4373	4373	3794850000	0.033471
5	2023-10-18	4357	4364	4304	4315	4315	3686030000	0.109508
6	2023-10-19	4321	4340	4270	4278	4278	3969730000	0.136607
7	2023-10-20	4274	4277	4223	4224	4224	4004030000	0.062067
8	2023-10-23	4210	4256	4189	4217	4217	3776100000	0.180923

Fig. 9. Combined dataframe with stock and sentiment data

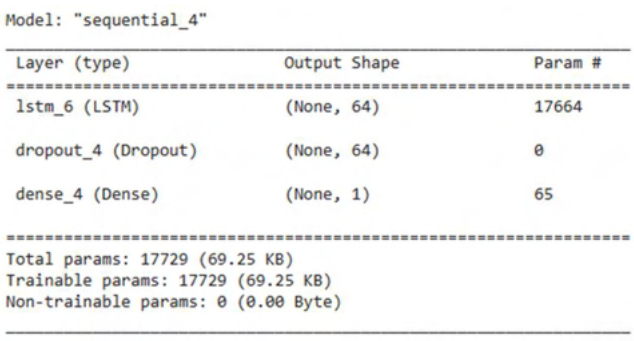
The *MinMaxScaler* function is shown in the equation below, where x(scaled) is the scaled value, x is the original cell value, min (x) is the minimum value and max(x) is the maximum value of the columns.

After normalizing our data, we store our input features in 'trainX' (i.e. past observations) and store the future observations in 'trainY'. We consider the last 3 days which is the 'n_past' days, and 'n_future' is the number of days to be predicted, which is 1 in this case. To prepare the training data for our machine learning model, we extract a sub-array of the scaled training data by considering the rows and the 2nd column to the last column in our dataframe. We then extract the target variable for the future time steps from the first column to get the last value in the predicted sequence. To feed our data into the machine learning model, we need to convert 'trainX' and 'trainY' into NumPy arrays. The next thing is to build our LSTM model using Keras. We use 64 LSTM cells in the layer, the input shape is what the layer expects as it agrees with the number of time steps, due to the size of our data, we only use one LSTM layer, and our model is set to return only the output of the last time step and not the full sequence. Making use of 80% of our data for training and the remaining 20% for validating the model, we train the model 15 times on the entire training dataset and also use a batch size of 1 which means the model will be updated after processing each sample.

Data Split and parameter setting. We split the data into training and testing sets using the ratio 80:20, this means that 80% of the data will be used for training the model, while 20% will be used to test the model. During model fitting, we tuned the hyperparameters to obtain the most accurate results. The paper uses parameters including epochs, batch size and validation split. We ran the training set through the LSTM model using 10 epochs, and a batch size which represents the number of samples used in each iteration for updating the model's weights, was set to 10 to achieve a balance of speed and model fitting. As previously stated, the validation split was done in a 80:20 manner.

3.8 Model Construction and Training Analysis

As required for LSTM networks, we reshape our input data into - n_samples x timesteps x n_features. In this paper, the value for n_features is 2, which represents the number of features or columns to be analyzed. We made use of 5 timesteps in constructing our model (timesteps refers to past days data used for training) and lastly n_samples indicate the number of values in both columns to be trained. The next step is to convert the dataset into a NumPy array because Machine Learning frameworks, including TensorFlow and PyTorch, are optimized to work with NumPy arrays. Converting the dataset to a NumPy array also made it easy to integrate it into the LSTM model for training and evaluation.



```
Model: "sequential_4"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 lstm_6 (LSTM)               (None, 64)                17664

 dropout_4 (Dropout)         (None, 64)                0

 dense_4 (Dense)             (None, 1)                 65

=================================================================
Total params: 17729 (69.25 KB)
Trainable params: 17729 (69.25 KB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
```

Fig. 4. Neural Network Model (LSTM)

To create training samples and target for the model, we looped over time indices and imported the necessary libraries, including TensorFlow or Keras, which provide tools for building neural networks. We created an instance of the Sequential model, which allows you to add layers one by one. 64 and 32 are the number of units (neurons) in each LSTM layer. These were adjusted based on the model complexity. We also used the activation function 'relu' in our model. The input_shape = (n_past, num_features) specifies the input shape, where n_past is the number of time steps in each input sequence, and num_features is the number of features in each time step. A dropout of (0.2) means randomly setting 20% of input units to 0 at each update during training, which helps prevent overfitting. The model was compiled by specifying the optimizer 'adam', loss function, and metrics. 'Adam' is an optimizer commonly used for gradient-based optimization.

3.9 Model Evaluation

For model evaluation, we made use of the mean squared error (MSE) to measure the performance of the machine learning model. The ML model is evaluated by comparing the predicted price and the real close price and computing the mean squared error.

$$Mean\ Square\ Error\ (MSE) = \frac{1}{n}\sum_{t=1}^{n} e^2$$

In the equation above, e denotes the error between the ground truth and the predicted value. The MSE is the average of accumulated error across the whole validation and test data set within every epoch to measure the actual performance of the neural network model.

4. RESULTS

In this section, we will present the results of the machine learning model and its suitability for predicting the prices of the S&P 500 index with and without text data. The model evaluation done on the forecast data using the Mean Squared Error (MSE), Mean Absolute Error (MAE) and the Mean Absolute Percentage Error (MAPE) provided results as shown below.

Table 2. Accuracy results for our data

Measure	With sentiment data	Without sentiment data
MAE	47.58034	58.01488
MAPE	1.119285	1.361079
MSE	3531.11	5237.70

As evidenced from the above results, our research questions 1 (RQ1) and RQ 2 have been answered, from table 2, we can see that our model returned better values using the sentiment data than without the sentiment data. This indicates that the model will better train and predict accurately the dataset with the sentiment data than the dataset without it. Fig 6 shows the training and validation loss, which are both measures of how well the model is fitting the training and test data.

Fig. 5. Actual stock price vs predicted stock price stock price without sentiment.

Comparing the results gotten from training our data using sentiment data and those gotten from not using it, we immediately find that the stock predictions (forecasted values) are closer to the actual stock price values when the Reddit data is combined with the historical stock data for model training. Although both scenarios provide positive results, the graph that is plot in fig 7 is closer in terms of the general trend of the actual stock price as opposed to the graph in fig 5.

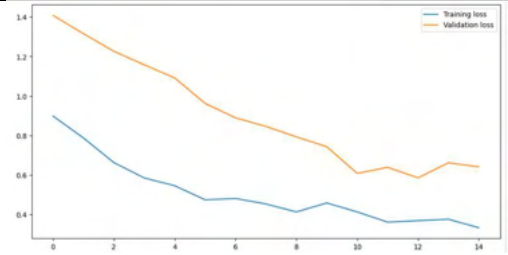


Fig. 6. Validation loss vs Training loss

Fig. 7. Actual stock price vs predicted stock price stock price with sentiment.

5. DISCUSSION AND CONCLUSION

By combining sentiment scores with historical stock prices, the mean squared error (MSE), MAPE and MAE were reduced by 32.5%, 21% and 21.9% respectively. We collected our sentiment data using the Reddit API, for a period of 30 calendar days, between October and November 2023. The small data size came with its own limitations as our predicted values did not exactly jive with the actual values, although to a large extent we were able to extract some information about how the sentiment (positive or negative) of certain social media platforms can affect stock prices. Similarly, stock prices can be affected by macro-economic factors such as GDP growth. [20] posits that a Vietnamese stock, VCB, is positively impacted by a growth in the GDP. The results further prove that the factors affecting stock market prices could vary from market economic rules, political leanings, social media sentiments and so on.

From RQ 1 we found the trend from our analysis showed that when there was a positive sentiment in the subreddit groups, the following day typically had an upward movement in the stock price and vice versa. This trend was recorded for more than 60% out of the 30 days. Therefore, we can conclude based on our results that there is some correlation between social media sentiments and stock price movements. From RQ 2 we found from comparing the predicted values with and without sentiment analysis and calculating each mean squared error, it is evident that the data with the sentiment scores predicted the actual values more accurately.

In conclusion, this study provided some evidence to the importance of social media sentiment for stock price predictions. The results using different measures of accuracy showed that the model performed better when combined with sentiments from social media. Further research can be done in this field whereby social media sentiments and stock news corpus can be combined to train neural network models for stock price predictions. The stock market however cannot be accurately predicted solely by combining sentiments with historic stock prices, several other factors like

market forces, economic policies etc., play a part in stock price movements, however this study moves the needle closer to making stock price prediction a possibility.

7. REFERENCES

[1] M. Kesavan, J. Karthiraman, R. T. Ebenezer, and S. Adhithyan, "Stock market prediction with historical time series data and sentimental analysis of social media data," in *2020 4th International Conference on Intelligent Computing and Control Systems (ICICCS)*, 2020: IEEE, pp. 477-482.

[2] J. Coelho, D. D'almeida, S. Coyne, N. Gilkerson, K. Mills, and P. Madiraju, "Social media and forecasting stock price change," in *2019 IEEE 43rd Annual Computer Software and Applications Conference (COMPSAC)*, 2019, vol. 2: IEEE, pp. 195-200.

[3] B. Kim, H. Kim, and G. Kim, "Abstractive summarization of reddit posts with multi-level memory networks," *arXiv preprint arXiv:1811.00783,* 2018.

[4] V. Morini, L. Pollacci, and G. Rossetti, "Capturing Political Polarization of Reddit Submissions in the Trump Era," in *SEBD*, 2020, pp. 80-87.

[5] O. Olabanjo *et al.*, "From Twitter to Aso-Rock: A sentiment analysis framework for understanding Nigeria 2023 presidential election," *Heliyon,* vol. 9, no. 5, 2023.

[6] A. S. Wisnubroto, A. Saifunas, A. B. Santoso, P. K. Putra, and I. Budi, "Opinion-based sentiment analysis related to 2024 Indonesian Presidential Election on YouTube," in *2022 5th International Seminar on Research of Information Technology and Intelligent Systems (ISRITI)*, 2022: IEEE, pp. 318-323.

[7] A. Sarkar, S. Chakraborty, S. Ghosh, and S. K. Naskar, "Evaluating Impact of Social Media Posts by Executives on Stock Prices," presented at the Proceedings of the 14th Annual Meeting of the Forum for Information Retrieval Evaluation, Kolkata, India, 2023. [Online]. Available: https://doi.org/10.1145/3574318.3574339.

[8] A. Trawinski, H. Wimmer, and D. Oliver, "Sentiment Based LSTM for Stock Price Prediction: Congress vs General Public," in *2022 International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT)*, 2022: IEEE, pp. 885-893.

[9] Y. Guo, "Stock price prediction based on LSTM neural network: the effectiveness of news sentiment analysis," in *2020 2nd International Conference on Economic Management and Model Engineering (ICEMME)*, 2020: IEEE, pp. 1018-1024.

[10] A. Sarkar, A. K. Sahoo, S. Sah, and C. Pradhan, "LSTMSA: a novel approach for stock market prediction using lstm and sentiment analysis," in *2020 International Conference on Computer Science, Engineering and Applications (ICCSEA)*, 2020: IEEE, pp. 1-6.

[11] M. Lim and C. K. Yeo, "Harvesting social media sentiments for stock index prediction," in *2020 IEEE 17th Annual Consumer Communications & Networking Conference (CCNC)*, 2020: IEEE, pp. 1-4.

[12] X. Weng, X. Lin, and S. Zhao, "Stock price prediction based on lstm and bert," in *2022 International Conference on Machine Learning and Cybernetics (ICMLC)*, 2022: IEEE, pp. 12-17.

[13] R. Karlemstrand and E. Leckström, "Using Twitter attribute information to predict stock prices," *arXiv preprint arXiv:2105.01402,* 2021.

[14] J. Sen, "STOCK PRICE PREDICTION USING DEEP LEARNING AND NATURAL LANGUAGE PROCESSING JAYDIP SEN and SIDRA MEHTAB."

[15] J. Bollen, H. Mao, and X. Zeng, "Twitter mood predicts the stock market," *Journal of computational science,* vol. 2, no. 1, pp. 1-8, 2011.

[16] L. Sidi, "Improving S&P stock prediction with time series stock similarity," *arXiv preprint arXiv:2002.05784,* 2020.

[17] S. Mohanty, A. Vijay, and N. Gopakumar, "Stockbot: Using lstms to predict stock prices," *arXiv preprint arXiv:2207.06605,* 2022.

[18] J. Kalyani, P. Bharathi, and P. Jyothi, "Stock trend prediction using news sentiment analysis," *arXiv preprint arXiv:1607.01958,* 2016.

[19] J. Kim, H. Wimmer, H. Liu, and S. Kim, "A Streaming Data Collection and Analysis for Cryptocurrency Price Prediction using LSTM," in *2021 IEEE/ACIS 6th International Conference on Big Data, Cloud Computing, and Data Science (BCD)*, 2021: IEEE, pp. 45-52.

[20] D. T. N. Huy, B. Loan, and P. T. Anh, "Impact of selected factors on stock price: a case study of Vietcombank in Vietnam," *Entrepreneurship and Sustainability Issues,* vol. 7, no. 4, pp. 2715-2730, 2020.