Stock Price Prediction with Twitter Sentiment Analysis

Andi Xu[1], Zhan Shi[2], Leyao Yang[3], and Mingxuan Qu[4]

{*andixu, zhanshi, leyyang, miqu*}@*umich.edu*

Abstract

Stock market is always a fascinating place for investors and researchers not only because of its potential to gain but also its unpredictable risk and changes in the market. As more intelligence grow in the recent years, unpredictable changes become more and more predictable. Researchers have successfully developed instincts in predicting stock prices with quantitative methods and machine learning techniques. With these research foundations, we take it to the next step by investigating public opinion on social media with respect to the stock market.

1 Introduction

Predicting the trend and fluctuations of the chaotic stock market is a popular field in research. Traditionally, traders would look at data such as candle graphs, figure out the support and resistance, and evaluate stock fluctuations using indicators such as Average True Range (ATR). There are limitations to these methods. It is hard for traders to deal with massive amounts of data at once and take qualitative data like sentiments and emotions. Consequently, traditional methods of predicting stock prices by human eyes are primitive.

People have developed fundamental analysis, technical analysis, and quantitative methods to analyze statistics to understand the stock price changes better. Fundamental analysis involves examining economic factors that influence the price of a stock. Such factors include a balance sheet and income statement. The balance sheet is a financial statement that provides information about a company's assets, liabilities, and equity of its shareholders at a specific point in time. Technical analysts use many different indicators calculated from stock price and volume history to predict future prices. Overall, the key to technical analysis is the trend. Technical analysis heavily relies on visualizations of various graphs and seeks interpretations to explain the pattern. Quantitative analysis is also popular with the emergence of big data and fast-developing computing power. Researchers performed times series analyses on the stock data. Additionally, leveraging natural language processing skills to interpret public and private opinions on social media is another way to acknowledge stock price changes.

Considering the above findings, we initiate the idea that takes advantage of both stock data and public opinions to model stock price changes. We experimented with clustering methods, sentiment computing, and time series analysis on the Twitter posts to perform sentiment analysis. We then feed the result into models such as random forest and neural networks methods to predict the trend of the stock. From this experiment, we observe that though sentiment knowledge overlaps with stock market information, it still can improve the performance by providing different public opinions.

In this project, we propose incorporating both stock market information and the market sentiment information on Twitter to understand the stock price fluctuations better.

We utilized Twitter posts and stock market data as our original dataset, using various machine learning algorithms to extract features, train the model and find the relationship between emotions and stock price fluctuations. When users input the company name, our model will retrieve recent Twitter posts and analyze and predict the stock trend.

We summarized our contributions as follows:

(1) we used Time Lagged Cross-Correlation (TLCC) and time series analysis to find the time dependencies between Twitter sentiments and stock prices. (2) We demonstrated aligning stock data and sentiment data with predicting stock market trends.

The remainder of the paper is organized as follows: In Section 2, we summarized previous researchers' work to predict the stock market. Section 3 introduced our experiment (data, method, algorithms and results). Section 4 is a reflection on our approaches and ethical implications.

2 Related Work

2.1 Stock market data analysis

Several works have used historical stock data to predict the future stock market. Ethan Johnson used the overall stock market state and volatility to classify the state of the economy with GMM. The GMM model is more flexible and suitable for other prediction tasks besides supervised learning. Ashwini Pathak el, used a series of classifiers – Random Forest, SVM, K-nearest neighbor, and Logistic Regression – to analyze data from the National Stock Exchange of India from 2016 to 2017. Their results showed that Random Forest had the best accuracy and recall, and Logistic Regression achieved the best precision and F-score.

2.2 Sentiment analysis

Researchers have also applied sentiment analysis to stock prediction. Tien Thanh Vu el, proposed new feature engineering techniques to pre-process raw Twitter data, such as defining positive, negative tagging, and bullish or bearish features. Rafeeque Pandarachalil el, used 4 kinds of ngrams (1, 2, 3, and 4 words at a time) as input, searching them through existing sentiment lexicons like SenticNet, SentiWordNet, and SentislangNet to compute the score for each term. Using the SN-SWN method, they could achieve an F Score of 67.46. Boon Peng Yap, el, used the Base BERT and neural networks to achieve an F1-score between 73.0 to 81.0.

However, the work above did not combine sentiment data and stock market data, and they didn't consider the time effectiveness of Twitter Sentiment. Our work proposed a new sentiment score processing technique, using time series analysis and TLCC, to align the two data types. We also used GMM and pos-tagging to categorize tweets, considering different topics of tweets would have different influences on the financial market. Finally, our work did experiments with a deep-learning method and a non-deep-learning method.

3 Experiment

3.1 Data

There are two major components that we utilized in this project - Twitter posts data and stock market data.

We employed the data set from Karolina's A Tweet-based Dataset for Company-level Stock Return Prediction for the Twitter posts data that we used to perform market sentiment analysis,. The data set contains 928673 data entries. The Twitter posts were dated from 01/03/2017 to 12/07/2018. Each data entry is described as the post id, post-created time, and post text content.

For the market stock data that we leveraged to make stock price predictions, we retrieved 20 companies' stock information, which lies in the same period as Twitter posts data. The 20 companies include Tesco, H&M, Adidas, Microsoft, Ryanair, Walmart, AT&T, TMobile, CBS, Facebook, Disney, McDonald's, Starbucks, Google, Apple, Amazon, Netflix, Reuters, Nike, and eBay, which are companies with the most number of tweets in our dataset. There are six features to describe each day in the stock market: open price, close price, high price, low price, adjusted close price, and transaction volume. There are around 480 data entries for each company.

3.2 Method

With the two main datasets, we performed market sentiment analysis. We output the sentiment score indicating positiveness and negativeness and the offset describing the post's effective period on the market. Then, we generated a combined dataset, Combined_Dataset, for prediction usage, which contains both stock and sentiment information.

On the sentiment side, we have Tweets with dates associated with each company as input. We first used GMM to cluster and label tweets into different categories. We used pos-tagging and set features as ['anger': 0, 'anticipation': 1, 'disgust':

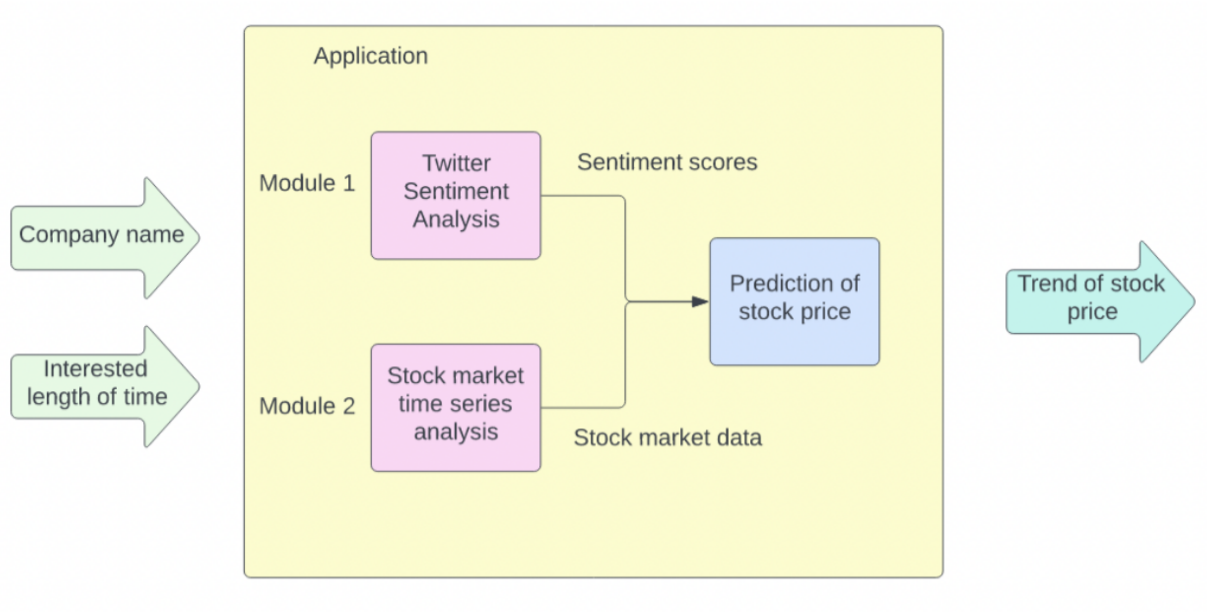
Application
Company name
Interested length of time
Module 1
Twitter Sentiment Analysis
Sentiment scores
Prediction of stock price
Module 2
Stock market time series analysis
Stock market data
Trend of stock price

Figure 1: Overview of our system

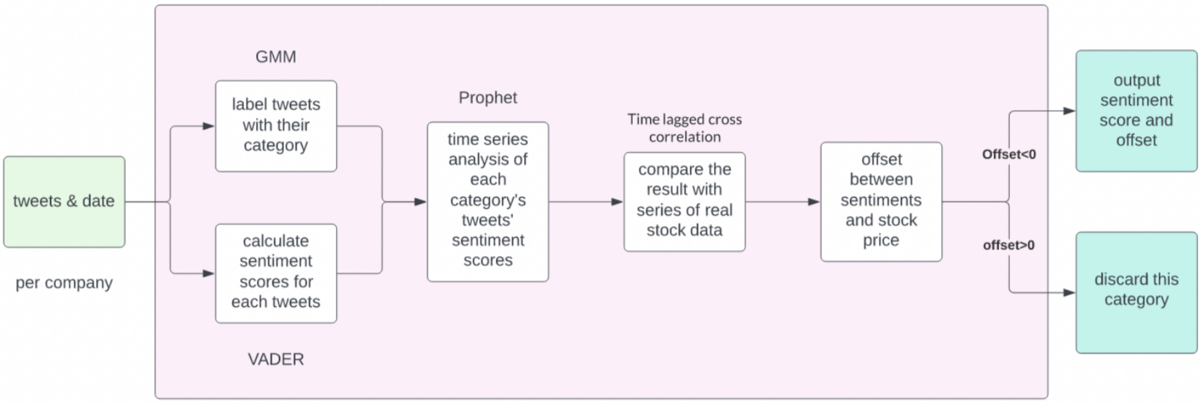
GMM
label tweets with their category
Prophet
time series analysis of each category's tweets' sentiment scores
Time lagged cross correlation
compare the result with series of real stock data
offset between sentiments and stock price
Offset<0
offset>0
output sentiment score and offset
discard this category
tweets & date
per company
calculate sentiment scores for each tweets
VADER

Figure 2: Procedure of data processing

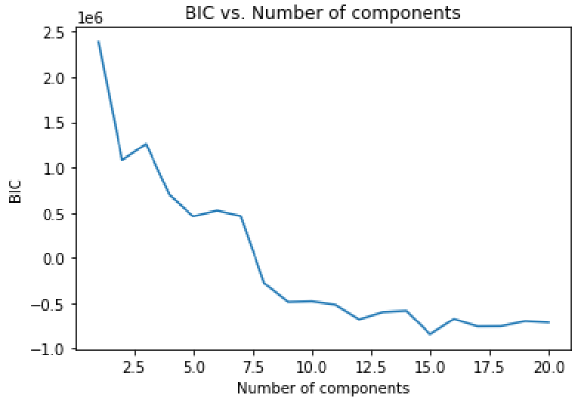

Figure 3: Results of using BIC to find n_components in GMM

2, ’fear’: 3, ’joy’: 4, ’sadness’: 5, ’surprise’: 6, ’trust’: 7, ’negative’: 8, ’positive’: 9, ’CC’: 10, ’IN’: 11, ’JJR’: 12, ’JJS’: 13, ’PRP’: 14]. We experimented with only the first eight features, but the 14-feature version turned out to have a better prediction in later experiments. We used BIC and found that 15 components worked the best for GMM clustering performance.

After labeling every tweet with its category, we used VADER to get the compound sentiment score for each Tweet. For one company, it has 15 categories of tweets. We used the average and variance of all compound scores as representation. Therefore, one company has $15 \times 2 = 30$ entries of sentiment score on one day.

For the next step, we are interested in computing the effective period that will be realized in the market. Intuitively, different types of news have different effective periods. We computed the effect offset to demonstrate the information lag in the market flow. To achieve this step, we first used Prophet to perform a time-series analysis on each category’s Tweet sentiment scores, then we did a time series analysis on stock close prices again. We then used Time Lagged Cross-Correlation (TLCC) to compare the result from the two time-series data, which returned us the offset between the sentiment and the stock price. TLCC can identify directionality between two signals such as a leader-follower relationship in which the leader initiates a response that is repeated by the follower.[2] If the offset is greater than 0, indicating that the Twitter sentiment is leading and affecting the stock market movement, we will keep the sentiment result when merging stock data and sentiment data, since we were investigating Twitter’s effect on the stock market. On the other hand, if the offset is less than 0, indicating that the market is leading the Twitter sentiment, in which case we will discard it. The offset was calculated separately for each company.

On the stock market side, we fetched the stock market data from Yahoo Finance described by open price, close price, high price, low price, adjusted close price, and transaction volume on each day for each stock. We normalized the data of each feature to alleviate the absolute stock price effect on the computation.

At last, we combined our work from these two modules and prepared the data for stock price prediction. We concatenated the two data sets so that each data entry represents a company’s relevant statistics on a day, which includes both the stock market information as well as the associated company’s Twitter sentiment information.

To match sentiment scores with stock data, we used offset scores. With one offset O, we can match a company’s sentiment score on day S with its stock data on day S+O. As a result, the numerical representation of a company on one day is expressed as sentiment scores given by offset, normalized open price, normalized close price, normalized high price, normalized low price, normalized adjusted close price, and transaction volume. Each row of Combined_Dataset has 36 numbers. The first 6 are stock data: normalized open price, normalized close price, normalized high price, normalized low price, normalized adjusted close price, and transaction volume. The next 30 are sentiment scores: there are 15 clusters, and each cluster has 1 mean score and 1 variance score.

3.3 Algorithms

With Combined_Dataset, we performed supervised learning to study the relationships between them and the stock close prices. Considering the chaotic nature of the stock market and to lower the influence of the wrong prediction, instead of predicting a numerical value of stock close prices, we changed the problem to a binary classification problem. Given a desired time length N (N=0 means

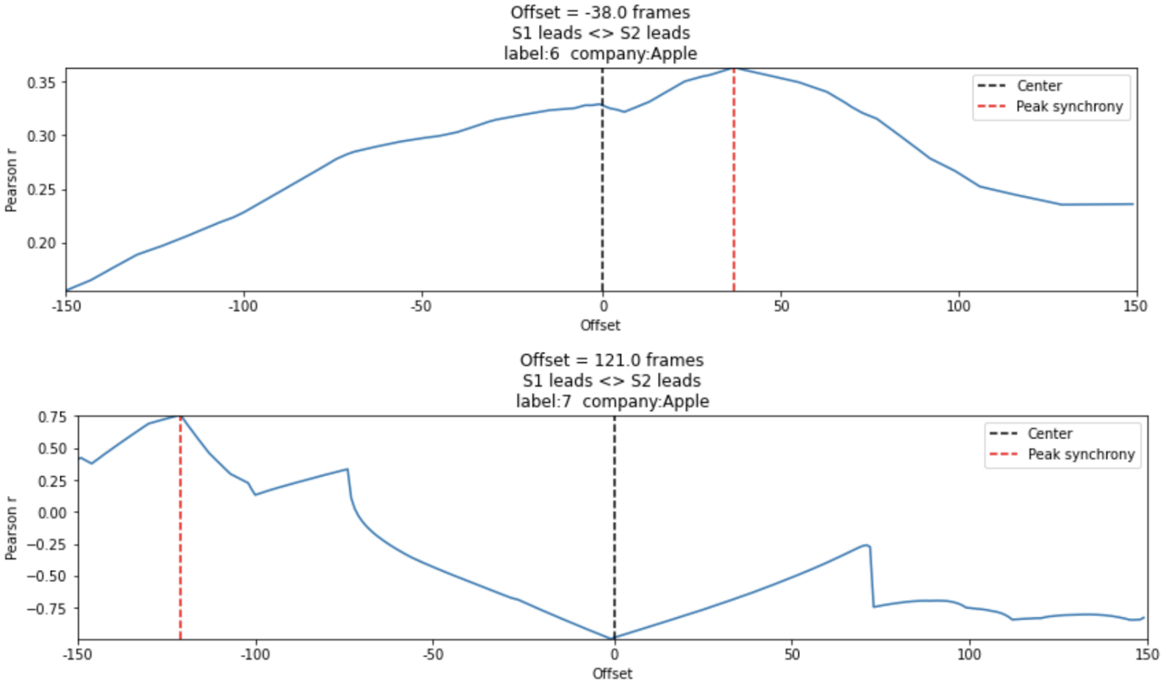

Figure 4: Upper: The offset (red line) shows that stock price is leading the interaction of sentiment of tweets in category 6 (correlation is maximized when sentiment score is pulled forward by 38 frames). Lower: Sentiment of tweets in category 7 is leading the interaction (correlation is maximized when stock price is pulled forward by 121 frames).

same day, =1 means after 1 day, =2 after 2 days, etc), if the close price of a company's stock is price S at the starting day and price E at day N, we label the stock trend as 1 if $S > E$ and 0 if $S \leq E$. Therefore for each company, on each day, there are 6 labels (either 0 or 1) corresponding to N=0, 1, 2, 3, 5, 10. To predict the stock trend, with Combined_Dataset and labels, we chose two classification algorithms, a simple random forest classification, and CNN-LSTM, a deep learning method.

3.3.1 Random Forest

Stock market data is very chaotic and can vary much during different periods. Simple models are easy to train but prone to overfit, thus not suitable for our problem. Random Forest is an ensemble method, which uses multiple randomly partitioned decision trees and chooses the majority vote of the result. We chose Random Forest as our non-deep-learning method because it is less likely to be impacted by noise and is robust to outliers.

We used the implementation from the Random Forest package in sklearn and set random_state as 42. At fine-tuning, we set the multiple hyperparameter candidates and used sklearn.model_selection.RandomizedSearchCV to randomly search through all combinations. We chose Random Search strategy over Grid Search because the former can achieve similar results with less time. After experiment we found that the best hyperparameters are 'n_estimators': 80, 'min_samples_split': 2, 'min_samples_leaf': 1, 'max_features': 'auto', 'max_depth': 28, 'criterion': 'entropy', 'bootstrap': True.

	Sentiment	**Stock**	**Combined**
Avg accuracy	0.56	0.66	0.75

Table 1: Results of Random Forest

We experimented with three types of input: only use the stock data (columns 1-6) of Combined_Dataset as input, only use sentiment score (columns 7-36) as input, and use all columns as input. We combined the 6 labels into a 6-dimensional vector and each entry represents one label. We then fed the data into our Random Forest model to make predictions.

Results The first model only used sentiment data to predict stock price trends and it produced an accuracy of 56%. The second model only used stock market data, and we achieved 66%

accuracy. The third model used stock market data and sentiment data as input to predict trends, which produced 75% accuracy. Though 75% seems not to be very high compared with other binary classification tasks, considering the fluctuating nature of the stock market, we think the accuracy is reasonable.

The experiment results show that sentiment data alone cannot capture the fluctuations in stock close prices. Close prices are related to other data in the stock market, but they are also influenced by many other factors. Combining stock data and sentiment data can better predict stock close prices.

3.3.2 CNN-LSTM

In order to solve the problem about time, we used LSTM and wanted to set the sequence length as 30, aiming to capture inter-month dependencies. We implemented our model in Pytorch and reshaped our data. Firstly, since each company has 480-490 days of data, we only took the first $480 = 16 \times 30$ days and discarded the rest. We then reshaped the dataset into a tensor of shape ($16 \times 20 = 320$, 30, 36). Label data underwent a similar procedure and has shape (320, 30, 6). We used 0-200 as training data, 200-280 as validation data, and 280-320 as testing data. For hyperparameters, we set batch size = 40, learning rate = 10^{-3}, epoch = 100. We set dataloader's shuffle parameter as False to prevent losing the time dependencies. Our network is shown in Table 2. B represents batch size.

Input Size: (B, 30, 36)
Structure:
Conv 1: (B, 30, 36) → $(B, 64, 20)$
Conv 2: (B, 64, 20) → $(B, 64, 10)$
Max Pooling: (B, 64, 10) → $(B, 64, 5)$
Flatten Vector: (B, 320)
Stack 30 times: (B, 30, 320)
LSTM: (B, 30, 320) → $(B, 30, 16)$
Fully connected: (B, 30, 16) → $(B, 30, 2)$
Binary Classification

Table 2: CNN-LSTM Architecture

CNN We applied two convolutional layers at the beginning to upscale the sequence length from 36 to 64 and downscale feature dimensions from 36 to 10 using two convolutional layers. We then added one max-pooling layer to further reduce feature dimensions to 5. Now the data dimension is (B, 64, 5). After that, we flattened the data into (B, 320) and repeated it 30 times along dim 1 to get (B, 30, 320), which fit with a sequence length of 30.

LSTM We used one LSTM layer and set the hidden state and cell state to default 0. Output has dimensions (B, 30, 16). Binary classification: We used a fully connected layer to get a tensor of shape (B, 30, 2). We used softmax and then argmax to get the final result.

Result However, our neural network did not achieve results better than Random Forest. The Result is shown in Table 3. Since our dataset is imbalanced, we used precision score besides accuracy as an evaluating metric. Training accuracy and training precision for all 6 types are around 0.6. The highest testing accuracy happened with 10_day_trend, with 0.53. The highest precision score happened with 5_day_trend, with 0.56. 10_day_trend also achieved the highest recall score of 0.8. We also noticed that 10_day_trend achieved overall best testing performance, and these results align with [1]'s results that stock data and sentiment data can best support the prediction of 10_day_return of a stock.

4 Discussion and Implications

4.1 Discussion on results

Our Random Forest Model has the problem that it cannot capture "time" in data. To address the issue and improve accuracy by involving in more complex models, we began experimenting with neural networks and hoped to achieve a better result. However, our CNN-LSTM model did not perform well. But we conclude that it cannot prove that neural networks work worse than random forest in stock prediction because it might imply there are other problems with our experiments. We will discuss them in this section.

Dataset The stock data retrieved from Yahoo Finance has no missing data. However, for Twitter data, after we finished the processing and turned tweets into sentiment vectors, we noticed there are a lot of missing values in our dataset. We concluded the reasons that caused sparse data as follows: first, we used 20 and 15 as dimension of sentiment vectors, and that number might be too large for modeling. Second, though we chose the 20 companies in our dataset with the most number of tweets, some companies still do not have enough tweets to generate a good vector of sentiment score and have

	Training		Testing			
Label type	accuracy	precision	accuracy	f1 score	precision score	recall score
same day	0.5800	0.5832	0.5083	0.5845	0.5123	0.6803
1 day	0.5700	0.5700	0.4975	0.4903	0.5088	0.4731
2 day	0.5700	0.5875	0.5275	0.6557	0.5294	0.8612
3 day	0.5800	0.5928	0.5125	0.5585	0.5362	0.5827
5 day	0.5900	0.5979	0.5317	0.6248	0.5571	0.7112
10 day	0.6000	0.6152	0.5300	0.6510	0.5489	0.8000

Table 3: Result of CNN-LSTM Model.

missed values on certain dimensions. We replaced missing data with 0s, considering 0 could be interpreted as no emotional preference in VADER compound score. Having many 0s might make the features less distinguishable.

Sentiment score processing We did several steps to get the final sentiment scores for each company in one day. We removed stopwords, punctuation, and metadata like links and mention of account names. Then we calculated the VADER score. The final 2 scores we used to represent a cluster are mean and variance. There are several potential improvements: first, because there are no existing packages to remove out-of-vocabulary words, we ignored this in our preprocessing. Second, VADER compound score might not be the best sentiment score to choose, and experiments on BERT could be done. Finally, mean and variance might not be able to represent clusters accurately, and we could explore other ways like distributions.

Neural network design When we were building our network, we referred to an existing model that used CNN-LSTM in its architecture and made changes based on that model. However, the original model was designed for geological data. Therefore it might fail to capture the “time” we cared about in the LSTM layer. There might also be problems in our number of layers, the way we structured the layers, and the ways we down or upsampled the features.

4.2 Ethical Implications

The financial nature of our project poses many challenges to its ethical implications. In this section, we will list 3 major implications and discuss how we have addressed or will address them.

1. Our results show that Twitter sentiment does influence the stock market. However, some Twitter users might take use of the result and spread fake news on Twitter to manipulate the market. People who learned about our study might also consider random investing suggestions on Twitter, but it is risky because those tweets users might have little financial knowledge. In future work, we can investigate what Twitter accounts influence the stock market the most. Future work could also be done on the influence of fake news on the stock market.

2. Our current prediction was done on historical data, collected by other researchers. If we want to extend our project to retrieve the newest data from Twitter API to predict the current stock market, we need to acquire users’ consent. We also need to pay attention to privacy issues. The implications could be resolved in the following ways. First we could anonymize all tweets that we collect and discard the original tweets when this process completes. Second, since Twitter asks users upon their willingness to make their tweets for research purposes, it is possible to retrieve data from those people exclusively.

3. It is difficult to capture every effective factor in the stock market. We recognize the limit of our model and have considered possible consequences of inaccurate predictions. We do not want our model to trick users into the act of “gambling” in the stock market. Also, people have different investing preferences and ability to take risks. Therefore, we specifically designed our model to only give out binary results and not give any trading suggestions. Though this cannot completely remove the implications on financial well-being, it will relieve the threats to some degree.

5 Conclusion

We conclude that both twitter sentiment data and historical stock market data could be used solely to predict future market fluctuations. However, combining them together could achieve a better performance. Our random forest implementations achieved satisfactory results of around 70 percent, while our neural network did not. Further work could be further exploring time as a factor in the stock market and effective ways to quantify tweets and build sentiment data, and explore better ways to handle missing values.

6 Reference

1. A. Pathak and S. Pathak, "Study of machine learning algorithms for stock market prediction," International Journal of Engineering Research amp; Technology, 15-Jun-2020. [Online]. Available: https://www.ijert.org/study-of-machine-learning-algorithms-for-stock-market-prediction. [Accessed: 19-Apr-2022].

2. J. Liu, C. Wang, J. Gao, and J. Han, "Multi-view clustering via joint nonnegative matrix Factorization," Multi-View Clustering via Joint Nonnegative Matrix Factorization. [Online]. Available: http://hanj.cs.illinois.edu/pdf/sdm13_jliu.pdf. [Accessed: 20-Apr-2022].

3. K. Sowinska and P. Madhyastha, "A tweet-based dataset for company-level stock ... - arxiv." [Online]. Available: https://arxiv.org/pdf/2006.09723. [Accessed: 20-Apr-2022].

4. M. S. Ethan Johnson-Skinner, "Using an unsupervised machine learning algorithm to detect different stock market regimes," Medium, 01-Dec-2021. [Online]. Available: https://medium.datadriveninvestor.com/using-an-unsupervised-machine-learning-algorithm-to-detect-different-stock-market-regimes-5c6354a1826a. [Accessed: 19-Apr-2022].

5. R. Pandarachalil, S. Sendhilkumar, and G. S. Mahalakshmi, "Twitter sentiment analysis for large-scale data: An unsupervised approach - cognitive computation," SpringerLink, 07-Nov-2014. [Online]. Available: https://link.springer.com/article/10.1007/s12559-014-9310-zciteas. [Accessed: 19-Apr-2022].

6. T. T. Vu, S. Chang, Q. T. Ha, and N. Collier, "An experiment in integrating sentiment features for Tech stock prediction in Twitter," ACL Anthology. [Online]. Available: https://aclanthology.org/W12-5503/. [Accessed: 19-Apr-2022].